

**Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Κρήτης**



**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής**

Πτυχιακή Εργασία

**Τίτλος: «Ανάπτυξη λογισμικού για την παρακολού-
θηση ασθενών οφθαλμολογικής κλινικής.»**

Παναγιώτης Ζιώγας (Α.Μ. 679)

Επιβλέπων Καθηγητής: Χατζάκης Ηλίας

Επιτροπή Αξιολόγησης:

Ημερομηνία Παρουσίασης:

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω, πρώτιστα, τον επόπτη καθηγητή μου Κύριο Χατζάκη Ηλία για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτήν την εργασία καθώς, επίσης, και για την πολύτιμη βοήθεια και καθοδήγησή του σε όλη τη διάρκειά της. Κυρίως τον ευχαριστώ για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα πολύ ενδιαφέρον αντικείμενο, όπως αυτό της μηχανοργάνωσης των νοσοκομείων και τη χρησιμοποίηση των σύγχρονων τεχνολογιών της Επιστήμης της Πληροφορικής για την καλύτερη εξυπηρέτηση τόσο των ασθενών όσο και των ιατρών.

Επιπρόσθετα, θα ήθελα να εξάρω την υπερπολύτιμη βοήθεια του στενού φίλου μου Καραδήμα Χρήστου, διότι χωρίς τις χρήσιμες συμβουλές του, θα ήταν αδύνατη η εκπόνηση της πτυχιακής τούτης.

Τέλος, θα ήταν μεγάλη παράλειψή μου να μην εκφράσω τις ευχαριστίες μου σε όλους τους κοντινούς μου ανθρώπους και ξεχωριστά στον πατέρα μου Ζιώγα Ιωάννη, στη μητέρα μου Χόντου Χρυσούλα, στην αδερφή μου Ζιώγα Αικατερίνη, στη σύντροφό μου Κουτσογιάννη Νόνικα καθώς και στους αδελφικούς μου φίλους Μακρή Αθανάσιο, Δερβίση Σπυρίδων και Χρήστο Μάνο, για την αμέριστη υλική και ηθική συμπαράστασή τους σε όλο το χρονικό διάστημα τόσο της ενασχόλησής μου με τούτη την αναφορά όσο του συνόλου των σπουδών μου.

Abstract

The purpose of this diplomatic thesis (work) is the development of a software system for the surveillance of patients of an eye hospital. For this purpose the development of a database is required, which will register system administrators, doctors, eye hospitals, patients and ophthalmologic examinations of patients, which will be conducted in any case. Access to the base will be permitted only to classified online users. In addition, safe handling and storage of data must ensure medical confidentiality.

More specifically three kinds of users will be permitted to access the information system, which is going to be developed in this thesis, namely system administrators, doctors, eye hospitals. Thus, current system is based on the interaction of administrator-doctor-eye hospital-patient in order to conduct ophthalmologic examinations.

HIS (Hospital Information Systems) make use of computers with regard to collect and organize medical, technical and administrative information, which concern the health and the care of patients, and with the purpose of easy and flexible access to this kind of information. The implementation of such a system will be accomplished in this thesis using the Web applications programming technique JSP and the database management system MySQL.

In conclusion the detailed implementation of a HIS and the benefits that can be gained by the NHS (National Health System) of a country using such systems, which are, for example, the minimization of latency as far as examination of insured people is concerned, the elimination of bureaucracy and mainly the reduction of operational costs, will be stated in this thesis.

Σύνοψη

Η παρούσα διπλωματική εργασία έχει σκοπό την ανάπτυξη ενός συστήματος λογισμικού για την παρακολούθηση ασθενών οφθαλμολογικής κλινικής. Για το λόγο αυτόν, απαιτείται η ανάπτυξη μίας βάσης δεδομένων, η οποία θα καταχωρεί, τους διαχειριστές του εν λόγω συστήματος, τους ιατρούς, τα νοσοκομεία, τους ασθενείς καθώς και τις οφθαλμιατρικές εξετάσεις των τελευταίων, που θα διενεργούνται σε κάθε περίπτωση. Η πρόσβαση στην προαναφερόμενη βάση θα γίνεται διαδικτυακά και μόνο από διαβαθμισμένους χρήστες. Επιπλέον, θα απαιτείται η ασφαλή διακίνηση και αποθήκευση κάθε μορφής δεδομένων, προκειμένου να εξασφαλίζεται το ιατρικό απόρρητο.

Πιο αναλυτικά, στο πληροφοριακό σύστημα που πρόκειται να αναπτυχθεί στα πλαίσια τούτης της αναφοράς θα μπορούν να εισέρχονται τρία είδη χρηστών: οι διαχειριστές του εν λόγω συστήματος, οι ιατροί και το κατάλληλο σε κάθε περίπτωση νοσοκομείο-οφθαλμολογική κλινική. Έτσι, λοιπόν, το τρέχον σύστημα θα έγκειται στην αλληλεπίδραση μεταξύ διαχειριστή-ιατρού-νοσοκομείου-ασθενούς για την πραγματοποίηση οφθαλμιατρικών εξετάσεων.

Τα ΠΣΝ (Πληροφοριακά Συστήματα Νοσοκομείων) αποσκοπούν στη μέσω υπολογιστών, συλλογή και οργάνωση ιατρικών, τεχνικών και διοικητικών πληροφοριών, που αφορούν την υγεία και την περίθαλψη των ασθενών και στην ταχεία και ευέλικτη προσπέλαση και ανάκτηση αυτών των πληροφοριών. Η υλοποίηση ενός τέτοιου πληροφοριακού συστήματος θα πραγματοποιηθεί στην παρούσα πτυχιακή διατριβή κατά βάση με χρήση της τεχνικής προγραμματισμού εφαρμογών του Διαδικτύου JSP και του συστήματος διαχείρισης βάσεων δεδομένων MySQL.

Κλείνοντας, κατά την συγγραφή της πτυχιακής εργασίας τούτης, θα παρουσιαστεί διεξοδικά η ανάπτυξη ενός τέτοιου συστήματος καθώς και τα οφέλη που μπορούν να αποκομίσει το εκάστοτε ΕΣΥ (Εθνικό Συστήματα Υγείας), χρησιμοποιώντας τέτοια συστήματα, όπως είναι, για παράδειγμα, η ελαχιστοποίηση του χρόνου αναμονής εξέτασης των ασφαλισμένων, η εξάλειψη της γραφειοκρατίας και κυρίως ο περιορισμός των λειτουργικών δαπανών.

Περιεχόμενα

1 Εισαγωγή στην Πτυχιακή Εργασία	16
1.1 Η Περίληψη της Πτυχιακής Εργασίας	16
1.2 Το Κίνητρο για τη Διεξαγωγή της Πτυχιακής Εργασίας	16
1.3 Ο Σκοπός και οι Στόχοι της Πτυχιακής Εργασίας	17
1.4 Η Δομή της Πτυχιακής Εργασίας	17
2 Το Θεωρητικό Υπόβαθρο της Πτυχιακής Εργασίας	19
2.1 Οι Βασικοί Όροι, Ιστοσελίδα και Ιστοτόπος	19
<i>2.1.1 Εισαγωγή στις Ιστοσελίδες και στους Ιστοτόπους</i>	<i>19</i>
<i>2.1.2 Οι Κατηγορίες των Ιστοτόπων</i>	<i>20</i>
<i>2.1.3 Σύγκριση των Στατικών και των Δυναμικών Ιστοτόπων</i>	<i>20</i>
2.2 Ο Βασικός Όρος, Εξυπηρετητής Ιστού	21
<i>2.2.1 Εισαγωγή στους Εξυπηρετητές Ιστού</i>	<i>21</i>
<i>2.2.2 Ο Εξυπηρετητής Ιστού Apache Tomcat</i>	<i>22</i>
<i>2.2.3 Σύγκριση του Εξυπηρετητή Ιστού Apache Tomcat</i>	<i>23</i>
2.3 Το Πρωτόκολλο Επικοινωνίας του Διαδικτύου, HTTP	24
<i>2.3.1 Εισαγωγή στο HTTP</i>	<i>24</i>
<i>2.3.2 Οι Κατηγορίες των Συνδέσεων του HTTP</i>	<i>25</i>
<i>2.3.3 Οι Μέθοδοι του HTTP</i>	<i>26</i>
<i>2.3.4 Οι Κεφαλίδες Μηνυμάτων του HTTP</i>	<i>27</i>
<i>2.3.5 Η Δομή ενός Μηνύματος του HTTP</i>	<i>29</i>
<i>2.3.6 Η Αλληλεπίδραση Χρήστη-Εξυπηρετητή στο HTTP</i>	<i>32</i>
<i>2.3.7 Το πρωτόκολλο HTTPS</i>	<i>33</i>
<i>2.3.8 Σύγκριση του HTTP</i>	<i>33</i>

2.4 Η Αρχιτεκτονική Ανάπτυξης Λογισμικού, Πελάτη-Εξυπηρετητή	34
2.4.1 Εισαγωγή στην Αρχιτεκτονική Πελάτη-Εξυπηρετητή	34
2.4.2 Τα Συστατικά της Αρχιτεκτονικής Πελάτη-Εξυπηρετητή	35
2.4.3 Οι Κατηγορίες της Αρχιτεκτονικής Πελάτη-Εξυπηρετητή	37
2.4.4 Σύγκριση της Αρχιτεκτονικής Πελάτη-Εξυπηρετητή	39
2.5 Η Αρχιτεκτονική Ανάπτυξης Λογισμικού, Πράκτορες Λογισμικού	39
2.5.1 Εισαγωγή στους Πράκτορες Λογισμικού	39
2.5.2 Οι Κατηγορίες των Πρακτόρων Λογισμικού	39
2.5.3 Τα Χαρακτηριστικά των Κινητών Πρακτόρων	41
2.5.4 Εφαρμογές των Κινητών Πρακτόρων Λογισμικού	41
2.5.5 Σύγκριση των Κινητών Πρακτόρων Λογισμικού	42
2.6 Η Αρχιτεκτονική Ανάπτυξης Λογισμικού, MVC	43
2.6.1 Εισαγωγή στην MVC	43
2.6.2 Εφαρμογές της MVC	44
2.6.3 Σύγκριση της MVC	44
2.7 Το Πλαίσιο Λογισμικού	45
2.7.1 Εισαγωγή στο Πλαίσιο Λογισμικού	45
2.7.2 Το Πλαίσιο Λογισμικού Διαδικτυακών Εφαρμογών, Apache Struts 1	45
2.8 Η Γλώσσα Μοντελοποίησης Λογισμικού, UML	46
2.8.1 Εισαγωγή στη UML	46
2.8.2 Τα Βασικά Συστατικά της UML	46
2.8.3 Σύγκριση της UML	51
2.9 Το Μοντέλο Αντικειμένου Εγγράφου	51
2.9.1 Εισαγωγή στο Μοντέλο Αντικειμένου Εγγράφου	51
2.9.2 Η Αναπαράσταση του Μοντέλου Αντικειμένου Εγγράφου	51
2.9.3 Οι Διεπαφές του Μοντέλου Αντικειμένου Εγγράφου	52

2.10 Η Γλώσσα Σήμανσης Ιστοσελίδων, HTML	53
2.10.1 Εισαγωγή στην HTML	53
2.10.2 Τα Βασικά Συστατικά της HTML	53
2.10.3 Οι Δεσμοί Υπερκειμένου της HTML	56
2.10.4 Άλλες Δυνατότητες της HTML	57
2.11 Η Γλώσσα Μορφοποίησης Περιεχομένου Ιστοσελίδων, CSS	61
2.11.1 Εισαγωγή στη CSS	61
2.11.2 Οι Κανόνες Σύνταξης της CSS	61
2.11.3 Οι Ιδιότητες της CSS	63
2.11.4 Οι Μέθοδοι Εισαγωγής της CSS	67
2.11.5 Η Προτεραιότητα της CSS	68
2.11.6 Η Κληρονομικότητα της CSS	68
2.11.7 Σύγκριση της CSS	69
2.12 Η Γλώσσα Σεναρίων, JavaScript	69
2.12.1 Εισαγωγή στην JavaScript	69
2.12.2 Οι Μέθοδοι Εισαγωγής της JavaScript	70
2.12.3 Οι Βασικοί Συντακτικοί Κανόνες της JavaScript	71
2.12.4 Οι Μεταβλητές, τα Δεδομένα και οι Τελεστές της JavaScript	72
2.12.4.1 Οι Μεταβλητές	72
2.12.4.2 Τα Δεδομένα	72
2.12.4.3 Οι Τελεστές	73
2.12.5 Οι Δομές Προγραμματισμού της JavaScript	77
2.12.6 Οι Συναρτήσεις της JavaScript	84
2.12.7 Τα Συμβάντα και οι χειριστές Συμβάντων της JavaScript	88
2.12.8 Τα Αντικείμενα της JavaScript	89
2.12.8.1 Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό	89
2.12.8.2 Τα Αντικείμενα	91
2.12.8.3 Το Αντικείμενο Array	91
2.12.8.4 Το Αντικείμενο String	96
2.12.9 Τα Παράθυρα Διαλόγου της JavaScript	100
2.12.10 Η JavaScript στα Προγράμματα Περιήγησης	102
2.12.10.1 Εισαγωγή	102

2.12.10.2 Το Αντικείμενο Window	103
2.12.10.3 Τα Πλαίσια	110
2.12.10.4 Το Αντικείμενο Navigator	111
2.12.10.5 Το Αντικείμενο Screen	112
2.12.11 Τα Έγγραφα, οι Εικόνες και η JavaScript	112
2.12.11.1 Το Αντικείμενο Document	112
2.12.11.2 Το Αντικείμενο Image	114
2.12.12 Οι Φόρμες και η JavaScript	115
2.12.12.1 Εισαγωγή	115
2.12.12.2 Το Αντικείμενο Form	115
2.12.12.3 Τα Βασικά Συστατικά μίας HTML Φόρμας μίας Ιστοσελίδας	117
2.12.12.4 Η Υποβολή μίας HTML Φόρμας μίας Ιστοσελίδας σε ένα Διακομιστή Ιστού	118
2.12.12.5 Η Επικύρωση των Δεδομένων μίας HTML Φόρμας μίας Ιστοσελίδας	119
2.12.13 Η Βιβλιοθήκη jQuery της JavaScript	119
2.12.13.1 Εισαγωγή	119
2.12.13.2 Οι Μέθοδοι Εισαγωγής της jQuery	119
2.12.13.3 Η Σύνταξη της jQuery.	120
2.12.13.4 Οι Επιλογείς της jQuery	120
2.12.13.5 Οι Μέθοδοι της jQuery	122
2.12.14 Σύγκριση της JavaScript	123
2.13 Η Επεκτάσιμη Γλώσσα Σήμανσης Περιεχομένου Ιστοσελίδων, XML	123
2.13.1 Εισαγωγή στην XML	123
2.13.2 Η Μοντελοποίηση της XML	124
2.13.3 Οι Κατηγορίες της XML	125
2.13.4 Ο Επεξεργαστής και ο Αναλυτής της XML	126
2.13.5 Οι Ενότητες της XML	126
2.13.6 Η Δομή της XML	127
2.13.7 Τα Βασικά Συστατικά της XML	128
2.13.8 Οι Βασικοί Συντακτικοί Κανόνες της XML	133
2.13.9 Οι Χώροι Ονομασίας της XML	133
2.13.10 Σύγκριση της XML	134
2.14 Η Επεκτάσιμη Μορφή της HTML, XHTML	134
2.14.1 Εισαγωγή στην XHTML	134
2.14.2 Μετατροπή της HTML σε XHTML	134
2.14.3 Σύγκριση της XHTML με την HTML	135

2.15 Η Τεχνική Προγραμματισμού Διαδικτύου, AJAX	136
2.15.1 Εισαγωγή στην AJAX	136
2.15.2 Η Δημιουργία του Αντικειμένου XMLHttpRequest της AJAX	137
2.15.3 Η Διαδικασία Αποστολής Αιτήματος σε έναν Εξυπηρετητή Ιστού στην AJAX	138
2.15.4 Οι Ιδιότητες responseText και responseXML της AJAX	139
2.15.5 Το Γεγονός onreadystatechange της AJAX	139
2.15.6 Τα Πολλαπλά Αντικείμενα XMLHttpRequest της AJAX	140
2.16 Η Γλώσσα Προγραμματισμού, Java	142
2.16.1 Εισαγωγή στην Java	142
2.16.2 Οι Βασικοί Συντακτικοί Κανόνες της Java	143
2.16.3 Οι Μεταβλητές, τα Δεδομένα και οι Τελεστές της Java	144
2.16.3.1 Οι Μεταβλητές	144
2.16.3.2 Τα Δεδομένα	145
2.16.3.3 Οι Τελεστές	147
2.16.4 Οι Δομές Προγραμματισμού της Java	148
2.16.5 Οι Μέθοδοι της Java	148
2.16.6 Οι Πίνακες και τα Αλφαριθμητικά της Java	149
2.16.6.1 Οι Πίνακες	149
2.16.6.2 Τα αλφαριθμητικά	151
2.16.7 Οι Βασικές Έννοιες του Αντικειμενοστραφούς Προγραμματισμού	153
2.16.7.1 Οι Κλάσεις και τα Αντικείμενα	153
2.16.7.2 Τα Πακέτα	155
2.16.7.3 Οι Αρχές του Αντικειμενοστραφούς Προγραμματισμού	156
2.16.7.4 Οι Τελικές Κλάσεις και Μέθοδοι	158
2.16.7.5 Οι Αφηρημένου Τύπου Κλάσεις και Μέθοδοι	158
2.16.7.6 Οι Διασυνδέσεις	158
2.16.8 Τα Applets της Java	159
2.16.9 Οι Μικροεφαρμογές Διακομιστή Ιστού, Java Servlets	160
2.16.9.1 Εισαγωγή στις Java Servlets	160
2.16.9.2 Ο Τρόπος Λειτουργίας των Java Servlets	161
2.16.10 Η Τεχνολογία JavaBeans της Java	162
2.16.11 Οι Εξαιρέσεις στην Java	163
2.16.11.1 Εισαγωγή	163
2.16.11.2 Οι Τύποι των Εξαιρέσεων	164
2.16.11.3 Ο Χειρισμός των Εξαιρέσεων	165
2.16.11.4 Η Δημιουργία των Εξαιρέσεων ενός Χρήστη	166
2.16.12 Σύγκριση της Java	166

2.17 Η Java και οι Βάσεις Δεδομένων	168
2.17.1 Εισαγωγή	168
2.17.2 Οι Βάσεις Δεδομένων και οι Σχεσιακές Βάσεις Δεδομένων	168
2.17.3 Η Γλώσσα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων, SQL	169
2.17.4 Η Διασύνδεση JDBC της Γλώσσας Java	172
2.17.4.1 Εισαγωγή	172
2.17.4.2 Οι Κλάσεις και οι Διασυνδέσεις της JDBC	173
2.17.4.3 Η Σύνδεση της Java με τις Σχεσιακές Βάσεις Δεδομένων	174
2.17.5 Η Γλώσσα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων, MySQL	176
2.18 Η Τεχνική Προγραμματισμού Διαδικτύου, JSP	177
2.18.1 Εισαγωγή στις JSP	177
2.18.2 Η Αρχιτεκτονική των JSP	177
2.18.3 Ο Κύκλος Ζωής των JSP	178
2.18.4 Η Σύνταξη των JSP	180
2.18.5 Οι Κατευθυντήριες Οδηγίες των JSP	181
2.18.6 Οι Ενέργειες των JSP	183
2.18.7 Τα Προκαθορισμένα Αντικείμενα των JSP	188
2.18.8 Οι Φόρμες των JSP	190
2.18.9 Σύγκριση των JSP	190
3 Η Παρουσίαση του Πρακτικού Κομματιού της Πτυχιακής Εργασίας	192
3.1 Το Μοντέλο του Καταρράκτη	192
3.2 Η Ανάλυση και ο Καθορισμός των Απαιτήσεων του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	193
3.3 Ο Σχεδιασμός του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	197
3.3.1 Τα UML Διαγράμματα Περιπτώσεων Χρήσης του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	197
3.3.2 Τα UML Διαγράμματα Αντικειμένων του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	199
3.3.3 Το UML Διάγραμμα Καταστάσεων του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	201

3.3.4 Το UML Διάγραμμα Συνεργασίας του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	201
3.4 Ο Τρόπος Λειτουργίας του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	202
3.4.1 Η Είσοδος ενός Διαχειριστή στο Σύστημα Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	202
3.4.2 Η Είσοδος ενός Νοσοκομείου στο Σύστημα Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	213
3.4.3 Η Είσοδος της Γραμματείας ενός Νοσοκομείου στο Σύστημα Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	227
3.4.3 Η Επικύρωση των Πεδίων των Φορμών του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής	244
4 Επίλογος της Πτυχιακής Εργασίας	249
4.1 Τα Συμπεράσματα της Πτυχιακής Εργασίας	249
4.2 Το Οι Επεκτάσεις της Πτυχιακής Εργασίας	249
ΠΑΡΑΡΤΗΜΑΤΑ	250
ΠΑΡΑΡΤΗΜΑ Ι-ΒΙΒΛΙΟΓΡΑΦΙΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	251
ΠΑΡΑΡΤΗΜΑ ΙΙ-ΟΔΗΓΟΣ ΕΚΤΕΛΕΣΗΣ ΚΩΔΙΚΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	255
ΠΑΡΑΡΤΗΜΑ ΙΙΙ –ΠΑΡΟΥΣΙΑΣΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	257

Λίστα Εικόνων

Εικόνα 1: Η κεντρική ιδέα του HTTP	24
Εικόνα 2: Η δομή ενός μηνύματος αίτησης του HTTP	29
Εικόνα 3: Η δομή ενός μηνύματος απόκρισης του HTTP	30
Εικόνα 4: Η αρχιτεκτονική πελάτη-εξυπηρετητή δύο στρωμάτων	37
Εικόνα 5: Η αρχιτεκτονική πελάτη-εξυπηρετητή τριών στρωμάτων	38
Εικόνα 6.: Η αρχιτεκτονική μη-κινητού πράκτορα λογισμικού	40
Εικόνα 7: Η αρχιτεκτονική κινητού πράκτορα λογισμικού	40
Εικόνα 8: Η διασύνδεση των τμημάτων της αρχιτεκτονικής MVC	44
Εικόνα 9: Παράδειγμα μίας οντότητας του Μοντέλου ΟΣ της UML	47
Εικόνα 10: Παράδειγμα των ιδιοτήτων μίας οντότητας του Μοντέλου ΟΣ της UML	47
Εικόνα 11: Το σύμβολο της εξάρτησης της UML	47
Εικόνα 12: Παράδειγμα εξάρτησης της UML	48
Εικόνα 13: Το σύμβολο της γενίκευσης της UML	48
Εικόνα 14: Το σύμβολο της σύνδεσης της UML	48
Εικόνα 15: Το σύμβολο της υλοποίησης της UML	49
Εικόνα 16: Παράδειγμα αναπαράστασης ενός HTML εγγράφου μέσω του DOM	52
Εικόνα 17: Η ιεραρχία των μοντέλων στο Μοντέλο Αντικειμένου Εγγράφου	103
Εικόνα 18: Η δενδρική μορφή ενός εγγράφου της XML	128
Εικόνα 19: Η ταξινόμηση των οντοτήτων ενός εγγράφου της XML	130
Εικόνα 20: Η λειτουργία της AJAX	137
Εικόνα 21: Η λειτουργία του μεταγλωττιστή μίας παραδοσιακής γλώσσας προγραμματισμού	142
Εικόνα 22: Η λειτουργία του μεταγλωττιστή της γλώσσας προγραμματισμού Java	143
Εικόνα 23: Η κληρονομικότητα των κλάσεων της Java	157
Εικόνα 24: Η θέση των applet της Java	159
Εικόνα 25: Η θέση των Java Servlets σε μία διαδικτυακή εφαρμογή	161

Εικόνα 26: Η κλάση των εξαιρέσεων Throwable της Java	164
Εικόνα 27: Ο ρόλος της JDBC σε μία εφαρμογή της Java	172
Εικόνα 28: Η διαδικασία της σύνδεσης μίας εφαρμογής της Java με μία σχεσιακή βάση δεδομένων	174
Εικόνα 29: Η θέση του JSP container και των JSP αρχείων σε μία εφαρμογή στο Διαδίκτυο	177
Εικόνα 30: Ο τρόπος δημιουργίας Ιστοσελίδων με τη χρήση της JSP	178
Εικόνα 31: Οι φάσεις του κύκλου ζωής μίας JSP Ιστοσελίδας	180
Εικόνα 32: Το διάγραμμα του Μοντέλου του Καταρράκτη	192
Εικόνα 33: Το διάγραμμα της ανάλυσης των πιο βασικών απαιτήσεων του λογισμικού, όσον αφορά ένα χρήστη-διαχειριστή	195
Εικόνα 34: Το διάγραμμα της ανάλυσης των πιο βασικών απαιτήσεων του λογισμικού, όσον αφορά ένα χρήστη-ιατρό	196
Εικόνα 35: Το διάγραμμα της ανάλυσης των πιο βασικών απαιτήσεων του λογισμικού, όσον αφορά ένα χρήστη-νοσοκομείο	196
Εικόνα 36: Το διάγραμμα της κεντρικής ιδέας του λογισμικού	197
Εικόνα 37: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-διαχειριστή	198
Εικόνα 38: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-ιατρό	198
Εικόνα 39: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-νοσοκομείο	199
Εικόνα 40: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-τεχνικό	199
Εικόνα 41: Το UML διάγραμμα αντικειμένων του λογισμικού, όσον αφορά ένα χρήστη-διαχειριστή	200
Εικόνα 42: Το UML διάγραμμα αντικειμένων του λογισμικού, όσον αφορά ένα χρήστη-ιατρό	200
Εικόνα 43: Το UML διάγραμμα αντικειμένων του λογισμικού, όσον αφορά ένα χρήστη-νοσοκομείο	201
Εικόνα 44: Το UML διάγραμμα καταστάσεων του λογισμικού	201
Εικόνα 45: Το UML διάγραμμα συνεργασίας του λογισμικού	202
Εικόνα 46: Το αρχικό παράθυρο του λογισμικού	203
Εικόνα 47: Το παράθυρο προειδοποίησης της ανεπιτυχούς εσόδου στο λογισμικό ενός διαχειριστή	203
Εικόνα 48: Το παράθυρο της διαχείρισης πληροφοριών ενός διαχειριστή	204

Εικόνα 49: Το παράθυρο της φόρμας εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή	205
Εικόνα 50: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου ονόματος χρήστη του ιατρού	206
Εικόνα 51: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου κινητού τηλεφώνου του ιατρού	206
Εικόνα 52: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενων ονόματος χρήστη και κινητού τηλεφώνου του ιατρού	207
Εικόνα 53: Το παράθυρο επιβεβαίωσης της επιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή	207
Εικόνα 54: Το παράθυρο της φόρμας εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή	208
Εικόνα 55: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου ονόματος χρήστη του νοσοκομείου	209
Εικόνα 56: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενης ονομασίας του νοσοκομείου	209
Εικόνα 57: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενων ονόματος χρήστη και ονομασίας του νοσοκομείου.	210
Εικόνα 58: Το παράθυρο επιβεβαίωσης της επιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή	210
Εικόνα 59: Το παράθυρο της φόρμας εγγραφής ενός ασθενούς στο λογισμικό από ένα διαχειριστή	211
Εικόνα 60: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ασθενούς στο Λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου ΑΜΚΑ του ασθενούς	212
Εικόνα 61: Το παράθυρο επιβεβαίωσης της επιτυχούς εγγραφής ενός ασθενούς στο λογισμικό από ένα διαχειριστή	212
Εικόνα 62: Το αρχικό παράθυρο ενός ιατρού	214
Εικόνα 63: Το παράθυρο της φόρμας σύνταξης ενός παραπεμπτικού ασθενούς από έναν ιατρό	215
Εικόνα 64: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός παραπεμπτικού ασθενούς στο λογισμικό από έναν ιατρό, λόγω ήδη χρησιμοποιούμενου ID του παραπεμπτικού ασθενούς	216
Εικόνα 65: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός παραπεμπτικού ασθενούς στο λογισμικό από έναν ιατρό, λόγω ήδη χρησιμοποιούμενου ΑΜΚΑ του ασθενούς	217

Εικόνα 66: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός παραπεμπτικού ασθενούς στο λογισμικό από έναν ιατρό, λόγω λανθασμένης σύνδεσης του ΑΜΚΑ του ασθενούς με την Ασφάλεια του ασθενούς	217
Εικόνα 67: Το παράθυρο επιβεβαίωσης της επιτυχούς καταχώρησης ενός παραπεμπτικού Ασθενούς στο λογισμικό από έναν ιατρό	218
Εικόνα 68: Το παράθυρο της αναζήτησης των στοιχείων ενός παραπεμπτικού ασθενούς από έναν ιατρό	219
Εικόνα 69: Το παράθυρο προειδοποίησης της ανεπιτυχούς αναζήτησης ενός παραπεμπτικού ασθενούς από έναν ιατρό	219
Εικόνα 70: Το παράθυρο εμφάνισης των στοιχείων ενός παραπεμπτικού ασθενούς, το οποίο έχει συντάξει ο ιατρός ο οποίος είναι κάθε φορά ο τρέχων χρήστης του λογισμικού	220
Εικόνα 71: Το παράθυρο εμφάνισης των στοιχείων ενός παραπεμπτικού ασθενούς, το οποίο έχει συντάξει ένας διαφορετικός ιατρός από τον ιατρό ο οποίος είναι κάθε φορά ο τρέχων χρήστης του λογισμικού	221
Εικόνα 72: Το παράθυρο της αναζήτησης του ιστορικού ενός ασθενούς από έναν ιατρό	222
Εικόνα 73: Το παράθυρο προειδοποίησης της ανεπιτυχούς αναζήτησης του ιστορικού ενός ασθενούς από έναν ιατρό	222
Εικόνα 74: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος έχει παραπεμφθεί εστω μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περεταίρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από έναν ιατρό	223
Εικόνα 75: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος δεν έχει παραπεμφθεί ούτε μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περεταίρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από έναν ιατρό	224
Εικόνα 76: Το παράθυρο της αναζήτησης του ιστορικού ενός ιατρού από τον ίδιο τον ιατρό	225
Εικόνα 77: Το παράθυρο της αναζήτησης του ιστορικού ενός ιατρού από τον ίδιο τον ιατρό, ο οποίος έχει συντάξει τουλάχιστον ένα παραπεμπτικό ασθενούς	226
Εικόνα 78: Το παράθυρο της αναζήτησης του ιστορικού ενός ιατρού από τον ίδιο τον ιατρό, ο οποίος δεν έχει συντάξει ούτε ένα παραπεμπτικό ασθενούς	227
Εικόνα 79: Το αρχικό παράθυρο ενός νοσοκομείου	228
Εικόνα 80: Το παράθυρο της αναζήτησης των εκκρεμών παραπεμπτικών ασθενών ως προς τον ορισμό ραντεβού από ένα νοσοκομείο	229
Εικόνα 81: Το παράθυρο της επιλογής των εκκρεμών παραπεμπτικών ασθενών ως προς τον ορισμό ραντεβού από ένα νοσοκομείο	230
Εικόνα 82: Το παράθυρο προειδοποίησης της ανεπιτυχούς επιλογής εκκρεμούς παραπεμπτικού ασθενούς ως προς τον ορισμό ραντεβού από ένα νοσοκομείο	230
Εικόνα 83: Το παράθυρο ορισμού ραντεβού για έναν ασθενή από ένα νοσοκομείο	231

Εικόνα 84: Το παράθυρο προειδοποίησης του ανεπιτυχούς ορισμό ραντεβού για έναν ασθενή από ένα νοσοκομείο	232
Εικόνα 85: Το παράθυρο επιβεβαίωσης του επιτυχούς ορισμού ραντεβού για έναν ασθενή από ένα νοσοκομείο	232
Εικόνα 86: Το παράθυρο πληροφόρησης για τη μη ύπαρξη εκκρεμών παραπεμπτικών ως προς τον ορισμό ραντεβού σε ένα νοσοκομείο	233
Εικόνα 87: Το παράθυρο της αναζήτησης των εκκρεμών παραπεμπτικών ασθενών ως προς την καταχώρηση αποτελέσματος εξέτασης από ένα νοσοκομείο	234
Εικόνα 88: Το παράθυρο της επιλογής των εκκρεμών παραπεμπτικών ασθενών ως προς την καταχώρηση αποτελέσματος εξέτασης από ένα νοσοκομείο	235
Εικόνα 89: Το παράθυρο προειδοποίησης της ανεπιτυχούς επιλογής εκκρεμού παραπεμπτικού ασθενούς ως προς την καταχώρηση αποτελέσματος εξέτασης από ένα νοσοκομείο	236
Εικόνα 90: Το παράθυρο της καταχώρησης αποτελέσματος εξέτασης για έναν ασθενή από ένα νοσοκομείο	236
Εικόνα 91: Το παράθυρο επιβεβαίωσης της επιτυχούς καταχώρησης αποτελέσματος εξέτασης για έναν ασθενή από ένα νοσοκομείο	237
Εικόνα 92: Το παράθυρο πληροφόρησης για τη μη ύπαρξη εκκρεμών παραπεμπτικών ως προς την καταχώρηση αποτελέσματος εξέτασης σε ένα νοσοκομείο	238
Εικόνα 93: Το παράθυρο της αναζήτησης του ιστορικού ενός ασθενούς από ένα νοσοκομείο	239
Εικόνα 94: Το παράθυρο προειδοποίησης της ανεπιτυχούς αναζήτησης του ιστορικού ενός ασθενούς από ένα νοσοκομείο	239
Εικόνα 95: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος έχει παραπεμφθεί έστω μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περεταιίρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από ένα νοσοκομείο	240
Εικόνα 96: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος δεν έχει παραπεμφθεί ούτε μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περεταιίρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από ένα νοσοκομείο	241
Εικόνα 97: Το παράθυρο της αναζήτησης του ιστορικού ενός νοσοκομείου από το ίδιο το νοσοκομείο	242
Εικόνα 98: Το παράθυρο της αναζήτησης του ιστορικού ενός νοσοκομείου από το ίδιο το νοσοκομείο, στο οποίο έχει αποσταλεί τουλάχιστον ένα παραπεμπτικό ασθενούς	243
Εικόνα 99: Το παράθυρο της αναζήτησης του ιστορικού ενός νοσοκομείου από το ίδιο το νοσοκομείο, στο οποίο δεν έχει αποσταλεί ούτε ένα παραπεμπτικό ασθενούς	244
Εικόνα 100: Το παράθυρο προειδοποίησης που αναδύεται, όταν ένας χρήστης-διαχειριστής του λογισμικού δεν καταχωρήσει τον κωδικό του στη φόρμα εισόδου του λογισμικού	245
Εικόνα 101: Το παράθυρο προειδοποίησης που αναδύεται, όταν ένας χρήστης-διαχειριστής του λογισμικού καταχωρήσει στη φόρμα εγγραφής ιατρού του λογισμικού ένα όνομα χρήστη ιατρού το οποίο δε συμφωνεί με τους κανόνες συμπλήρωσης του αντίστοιχου πεδίου της φόρμας, οι οποίοι υποδεικνύονται άμεσα στην ετικέτα στα	

δεξιά του πεδίου	246
Εικόνα 102: Το παράθυρο προειδοποίησης που αναδύεται, όταν ένας χρήστης-ιατρός του λογισμικού καταχωρήσει στη φόρμα αναζήτησης ιστορικού ασθενούς του λογισμικού ένα ΑΜΚΑ ασθενούς το οποίο δε συμφωνεί με τις εξ ορισμού ιδιότητες της έννοιας ΑΜΚΑ	247
Εικόνα 103: Το παράθυρο προειδοποίησης που αναδύεται, όταν ένας χρήστης- νοσοκομείο του λογισμικού καταχωρήσει στη φόρμα ορισμού ραντεβού του λογισμικού μία ημερομηνία εξέτασης ασθενούς η οποία είναι προγενέστερη της τρέχουσας σε κάθε φάση ημερομηνίας	248

Λίστα Πινάκων

Πίνακας 1: Οι μέθοδοι του HTTP	26
Πίνακας 2: Οι πιο βασικές κεφαλίδες των μηνυμάτων του HTTP	28
Πίνακας 3: Οι κατηγορίες των κωδικών κατάστασης των μηνυμάτων απόκρισης του HTTP	31
Πίνακας 4: Οι πιο βασικοί ειδικοί χαρακτήρες της HTML	61
Πίνακας 5: Οι αριθμητικοί τελεστές της JavaScript	74
Πίνακας 6: Οι τελεστές εκχώρησης της JavaScript	75
Πίνακας 7: Οι τελεστές σύγκρισης της JavaScript	75
Πίνακας 8: Οι λογικοί τελεστές της JavaScript	76
Πίνακας 9: Οι τελεστές δυαδικών ψηφίων της JavaScript	77
Πίνακας 10: Οι πιο βασικές καθολικές συναρτήσεις της JavaScript	87
Πίνακας 11: Τα πιο βασικά συμβάντα και οι αντίστοιχοι χειριστές συμβάντων της JavaScript	89
Πίνακας 12: Οι πιο βασικές μέθοδοι του αντικειμένου Array της JavaScript	95
Πίνακας 13: Οι πιο βασικές μέθοδοι του αντικειμένου String της JavaScript	100
Πίνακας 14: Οι πιο βασικές ιδιότητες του αντικειμένου Window της JavaScript	104
Πίνακας 15: Οι πιο βασικές μέθοδοι του αντικειμένου Window της JavaScript	110
Πίνακας 16: Οι πιο βασικές ιδιότητες του αντικειμένου Navigator της JavaScript	111
Πίνακας 17: Οι πιο βασικές ιδιότητες του αντικειμένου Screen της JavaScript	112
Πίνακας 18: Οι πιο βασικές ιδιότητες του αντικειμένου Document της JavaScript	113
Πίνακας 19: Οι πιο βασικές μέθοδοι του αντικειμένου Document της JavaScript	114
Πίνακας 20: Οι πιο βασικές ιδιότητες του αντικειμένου Form της JavaScript	116
Πίνακας 21: Οι πιο βασικές μέθοδοι του αντικειμένου Form της JavaScript	116
Πίνακας 22: Οι πιο βασικοί επιλογείς της jQuery	122

Πίνακας 23: Οι πιο βασικοί προκαθορισμένοι χαρακτήρες οντοτήτων της XML	131
Πίνακας 24: Οι ακέραιοι τύποι δεδομένων της Java	146
Πίνακας 25: Οι αριθμοί κινητής υποδιαστολής της Java	146
Πίνακας 26: Οι πιο βασικές ακολουθίες διαφυγής της Java	146
Πίνακας 27: Οι αρχικές τιμές των μεταβλητών-μελών της Java	147
Πίνακας 28: Οι Τα πιο βασικά τυπικά πακέτα της Java	156
Πίνακας 29: Η δυνατότητα προσπέλασης των προσδιοριστών πρόσβασης της Java	157
Πίνακας 30: Οι μέθοδοι της διασύνδεσης Servlet	161
Πίνακας 31: Οι πιο βασικές εξαιρέσεις της Java	165
Πίνακας 32: Οι πιο βασικοί τύποι δεδομένων της SQL	170
Πίνακας 33: Οι πιο βασικές κλάσεις και διασυνδέσεις της JDBC	173
Πίνακας 34: Οι πιο βασικές κατευθυντήριες οδηγίες της JSP	182
Πίνακας 35: Οι πιο βασικές ιδιότητες της κατευθυντήριας οδηγίας page της JSP	183
Πίνακας 36: Τα στοιχεία δράσης της JSP	184
Πίνακας 37: Η ιδιότητα του JSP στοιχείου δράσης jsp:forward	185
Πίνακας 38: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης jsp:getProperty	185
Πίνακας 39: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης jsp:include	186
Πίνακας 40: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης jsp:setProperty	187
Πίνακας 41: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης jsp:useBean	188
Πίνακας 42: Τα προκαθορισμένα αντικείμενα της JSP	188
Πίνακας 43: Οι πιο βασικές μέθοδοι του αντικειμένου JSPWriter της JSP	189

1 Εισαγωγή στην Πτυχιακή Εργασία

Είναι, πλέον, ξεκάθαρο ότι η πρόοδος που έχει επιτευχθεί τη σύγχρονη εποχή στον τομέα της Τεχνολογίας και ειδικά στην Πληροφορική, επιλύει με εύκολο, γρήγορο και συνάμα δίκαιο και ασφαλή τρόπο καθημερινές ανάγκες των ανθρώπων, από τις πιο απλές, όπως είναι, για παράδειγμα, η διαδικτυακή ενημέρωση για τα καθημερινά δρώμενα μέχρι και τις πιο σύνθετες και κρίσιμες, όπως είναι για παράδειγμα, η εγχείριση ασθενούς μέσω Τηλεϊατρικής. Στην τρέχουσα πτυχιακή διατριβή θα γίνει αναφορά σε μία σημαντική περιοχή εφαρμογής του κλάδου της Πληροφορικής, που δεν είναι άλλος από τον ιδιαίτερα ευαίσθητο για μία χώρα τομέα της Υγείας. Η παρούσα αναφορά θα έχει ως αντικείμενο μελέτης της τον όρο ΠΣΝ (Πληροφοριακά Συστήματα Νοσοκομείων). Πιο συγκεκριμένα, θα παρουσιαστεί η ανάπτυξη ενός λογισμικού για την παρακολούθηση ασθενών οφθαλμολογικής κλινικής, τονίζοντας, παράλληλα, τα οφέλη που αποκομίζει ένα ΕΣΥ (Εθνικό Σύστημα Υγείας) με χρήση τέτοιων τεχνολογιών.

1.1 Η Περίληψη της Πτυχιακής Εργασίας

Σκοπός αυτής της πτυχιακής εργασίας είναι η όσο το δυνατό καλύτερη προσέγγιση του όρου ΠΣΝ, δηλαδή συστημάτων τα οποία αποσκοπούν στη συλλογή και οργάνωση ιατρικών, τεχνικών και διοικητικών πληροφοριών, που αφορούν την υγεία και περίθαλψη των ασθενών και στην ταχεία και εύελικτη προσπέλαση και ανάκτηση των προαναφερόμενων πληροφοριών με την αρωγή των ηλεκτρονικών υπολογιστών. Πιο ειδικά, στην τρέχουσα αναφορά θα γίνει μελέτη, σχεδίαση και ανάπτυξη ενός λογισμικού για την παρακολούθηση ασθενών οφθαλμιατρικής κλινικής και θα επισημανθούν, συνάμα, τα οφέλη που αποκομίζει ένα ΕΣΥ από την Πληροφορική, αξιοποιώντας τις δυνατότητες που τα ΠΣΝ παρέχουν, έτσι ώστε τόσο οι ιατροί όσο και οι ασθενείς να έχουν αυτοματοποιημένη και ασφαλή συλλογή πληροφοριών, οποιαδήποτε στιγμή οι ίδιοι το θελήσουν.

Για τη διεκπεραίωση των ανωτέρω θα γίνει συνδυαστική χρήση και εκμετάλλευση της γλώσσας σήμανσης κειμένου στο Διαδίκτυο, HTML, της γλώσσα μορφοποίησης κειμένου στο Διαδίκτυο, CSS, της γλώσσας σεναρίων, JavaScript, της επεκτάσιμης γλώσσας σήμανσης κειμένου στο Διαδίκτυο, XML, της αναδιατυπωμένης και αναβαθμισμένης HTML, XHTML, της γλώσσας προγραμματισμού, Java, των μικροεφαρμογών ενός διακομιστή Ιστού Java Servlets, της Java τεχνολογίας JavaBeans, της τεχνικής προγραμματισμού Διαδικτύου, JSP, και, τέλος, του συστήματος διαχείρισης βάσεων δεδομένων MySQL.

Εν κατακλείδι θα αναπτυχθεί ένα πληροφοριακό σύστημα που θα επιτρέπει την αλληλεπίδραση μεταξύ των διαχειριστών της διαδικτυακής εφαρμογής την οποία πραγματεύεται η παρούσα διπλωματική, των συμβεβλημένων με τα Ασφαλιστικά Ταμεία του τομέα της Υγείας μίας χώρας ιατρών, των, επίσης, συνεργαζόμενων με τους Ασφαλιστικούς Φορείς του κλάδου της Υγείας ενός κράτους νοσοκομείων-οφθαλμολογικών κλινικών, είτε δημοσίων είτε ιδιωτικών, και, επιπλέον, των ασθενών-ασφαλισμένων μίας χώρας, με απώτερο σκοπό και στόχο την όσο είναι εφικτό καλύτερη επίβλεψη της υγείας των τελευταίων.

[1]

1.2 Το Κίνητρο για τη Διεξαγωγή της Πτυχιακής Εργασίας

Η παρούσα πτυχιακή εργασία ανοίγει νέους ορίζοντες στον τομέα της Ιατρικής, αφού δείχνει τον τρόπο με τον οποίο με τη βοήθεια της Πληροφορικής μπορούν να αντικατασταθούν οι παραδοσιακές προσεγγίσεις διοίκησης-οργάνωσης-διαχείρισης ενός νοσοκομείου, που περιλαμβάνουν τόνους πληροφοριών αποτυπωμένες πάνω σε χαρτιά και θέσεις εργασίας με εξειδικευμένο προσωπικό, που

εκτελεί αυτές τις διεργασίες, από κατάλληλα πληροφοριακά συστήματα, μερικώς ή πλήρως αυτοματοποιημένα, μέσω των οποίων μπορεί να επιτευχθεί τόσο η άριστη δυνατή υποστήριξη των ασθενών όσο και η βέλτιστη διοίκηση του νοσοκομείου.

Αναλυτικότερα η χρήση Πληροφοριακών Συστημάτων Νοσοκομείων προσφέρει πολλά οφέλη στο σύστημα Υγείας μίας χώρας, τα βασικότερα από τα οποία είναι τα ακόλουθα:

- Εύκολη πρόσβαση σε στοιχεία των ασθενών, επειδή υπάρχει ταξινόμηση με βάση το φύλο, την ηλικία και με άλλα κριτήρια.
- Καλύτερη εξωνοσοκομειακή εξυπηρέτηση, διότι με την πρόσβαση στο Διαδίκτυο (Internet) υπάρχει η δυνατότητα απομακρυσμένης πρόσβασης δεδομένων.
- Αποτελεσματική και ακριβής διαχείριση των οικονομικών του εκάστοτε νοσοκομείου, της διατροφής των ασθενών του και της διανομής ιατρικής βοήθειας μέσω αυτού.
- Βελτιωμένη παρακολούθηση της χρήσης φαρμάκων και της μελέτης της αποτελεσματικότητας αυτών με άμεση συνέπεια τον περιορισμό των αρνητικών αλληλεπιδράσεων αυτών και παράλληλα την προώθηση της καταλληλότερης φαρμακευτικής αγωγής.
- Ενίσχυση της ακεραιότητας των πληροφοριών, μείωση των σφαλμάτων μεταγραφής και ταυτόχρονο περιορισμό της επανάληψης των καταχωρήσεων πληροφορίας.
- Υποστηρικτική δράση, όσον αφορά τις αποφάσεις του εκάστοτε νοσοκομείου για την ανάπτυξη πολιτικών σχετιζόμενων με την παρεχόμενη από ένα ΕΣΥ υγειονομική περίθαλψη.

Πέρα από τα παραπάνω πλεονεκτήματα, που θεωρούνται υψίστης σημασίας, υπάρχουν και άλλα δευτερεύουσας σημασίας, όπως είναι, για παράδειγμα, ο περιορισμός της γραφειοκρατίας, η βελτίωση της πληροφόρησης των συναλλασσομένων και της ταχύτητας εξυπηρέτησής τους, ο περιορισμός της χειρωνακτικής διαδικασίας και η βελτίωση του εργασιακού περιβάλλοντος, η αποκέντρωση, που πείθουν ότι η υιοθέτηση πληροφοριακών συστημάτων στον τομέα της Υγείας κρίνεται επιτακτική.

[2]

1.3 Ο Σκοπός και οι Στόχοι της Πτυχιακής Εργασίας

Στην παρούσα πτυχιακή εργασία θα υλοποιηθεί ένα πληροφοριακό σύστημα αλληλεπίδρασης μεταξύ ιατρού–οφθαλμολογικής κλινικής–ασθενούς για την πραγματοποίηση οφθαλμιατρικών εξετάσεων. Ιδιαίτερη σημασία για την βέλτιστη απόδοση του συγκεκριμένου πληροφοριακού συστήματος θα δοθεί στην δημιουργία ιατρικού φακέλου ασθενούς και στην ασφάλεια των στοιχείων αυτού. Με άλλα λόγια, θα επιχειρηθεί μία προσπάθεια να κατασκευαστεί ένα ολοκληρωμένο πληροφοριακό σύστημα, όπου θα αλληλεπιδρούν όλοι οι φορείς Υγείας μίας χώρας, δηλαδή οι ασθενείς, οι ιατροί, τα νοσοκομεία της εν λόγω χώρας, ώστε να πραγματοποιηθεί με όσο το δυνατό μεγαλύτερη ορθότητα ο ιατρικός φάκελος ασθενούς, κάτι που συνιστά κρίσιμο σημείο για την άριστη λειτουργία αυτού του συστήματος.

Οι διαφορές μεταξύ ενός κλασσικού χειρόγραφου ιατρικού φακέλου και ενός ηλεκτρονικού ιατρικού φακέλου είναι αξιοσημείωτες. Αυτός είναι και ο λόγος που μέρα με την μέρα ο δεύτερος κερδίζει έδαφος στο χώρο της Ιατρικής. Ένας χειρόγραφος ιατρικός φάκελος δίνει ελευθερία στην έκφραση, δεν απαιτεί ειδική εκπαίδευση του ιατρικού προσωπικού, μπορεί να μεταφερθεί εύκολα στους χώρους ενός νοσοκομείου χωρίς τη χρήση ηλεκτρονικού υπολογιστή και έχει ελάχιστο κόστος. Ωστόσο, ένας ηλεκτρονικός ιατρικός φάκελος μπορεί να χρησιμοποιηθεί ταυτόχρονα από πολλούς χρήστες, εφόσον η πρόσβαση σε αυτόν είναι γρήγορη και εύκολη, δεν τίθεται θέμα αναγνωσιμότητας αυτού, μίας και υποστηρίζεται μία δομημένη είσοδος των δεδομένων, και ενισχύεται η ορθότητα μίας ιατρικής απόφασης, αφού συγκεντρώνει όλα τα απαραίτητα στοιχεία ενός ασθενούς, εξάγοντας με τον τρόπο αυτό μία σαφή εικόνα για την συνολική κατάσταση της υγείας του. Επιπρόσθετα, ελαχιστοποιείται ο απαιτούμενος χώρος αποθήκευσης των κλασσικών φακέλων καθώς και ο φόβος να χαθούν σημαντικά δεδομένα που περιγράφουν την κατάσταση της υγείας του κάθε ασθενούς. Παρόλα αυτά, ένας ηλεκτρονικός ιατρικός φάκελος ελλοχεύει ένα μεγάλο κίνδυνο ο οποίος σχετίζεται με τη φύλαξη των προσωπικών δεδομένων καθενός ασθενούς και κατά επέκταση τη διασφάλιση του ιατρικού απορρήτου.

[3]

1.4 Η Δομή της Πτυχιακής Εργασίας

Στο πρώτο κεφάλαιο της τρέχουσας πτυχιακής εργασίας, που είναι εισαγωγικό, παρουσιάζεται, αρχικά, μία περίληψη και έπειτα μία γενική αναφορά στα κίνητρα διεξαγωγής, στο σκοπό και στους

στόχους αυτής. Στο δεύτερο κεφάλαιο προσεγγίζεται μία λεπτομερής θεωρητική ανάλυση των μοντέλων, των τεχνολογιών λογισμικού και των γλωσσών προγραμματισμού που θα συμβάλλουν στην υλοποίηση του επιθυμητού λογισμικού, το οποίο θα ανταποκρίνεται στις ανάγκες της συγκεκριμένης αναφοράς. Στο τρίτο κεφάλαιο, το οποίο συνιστά και το κύριο μέρος της πτυχιακής, γίνεται στην ουσία πρακτική εφαρμογή των θεωριών που μελετήθηκαν στο δεύτερο κεφάλαιο και επιτυγχάνεται με αυτόν κατά αυτόν τον τρόπο η ανάλυση των απαιτήσεων, ο σχεδιασμός και η υλοποίηση του συστήματος που διαπραγματεύεται η τρέχουσα πτυχιακή εργασία. Στο τέταρτο και τελευταίο κεφάλαιο αναφέρονται τα τελικά συμπεράσματα και οι επεκτάσεις του λογισμικού που αποτελεί το σκοπό και τον στόχο τούτης της εργασίας.

2 Η Ανάλυση του Θεωρητικού Υποβάθρου της Πτυ- χιακής Εργασίας

Στην ενότητα αυτή θα γίνει λεπτομερής ανάλυση των θεωριών, των μοντέλων, των αλγορίθμων και των εργαλείων ανάπτυξης και υλοποίησης του θέματος που πραγματεύεται η τρέχουσα πτυχιακή εργασία.

2.1 Οι Βασικοί Όροι, Ιστοσελίδα και Ιστοτόπος

2.1.1 Εισαγωγή στις Ιστοσελίδες και στους Ιστοτόπους

Με τον όρο Ιστοσελίδα (Web Page) ή απλά Σελίδα (Page) νοείται ένα είδος εγγράφου του Παγκόσμιου Ιστού (World Wide Web-WWW) ή απλά Ιστού (Web) που περιλαμβάνει πληροφορίες με την μορφή κειμένου, υπερκειμένου, εικόνας, ήχου, κινούμενων γραφικών και βίντεο.

Πολλές Ιστοσελίδες μαζί συνθέτουν έναν Ιστότοπο\Ιστοχώρο\Δικτυακό Τόπο (Web Site\Internet Site) ή απλά Τόπο\Χώρο (Site). Οι Σελίδες ενός Ιστοτόπου εμφανίζονται κάτω από το ίδιο Σύστημα Ονομάτων Τομέων\Χώρων\Περιοχών (Domain Name System-DNS) (πχ microsoft.com). Μία Ιστοσελίδα, τώρα, αποτελείται από αντικείμενα. Με τον όρο αντικείμενο μίας Ιστοσελίδας εννοείται ένα απλό αρχείο, όπως είναι, για παράδειγμα, ένα αρχείο εικόνας ή ένα αρχείο βίντεο, το οποίο μπορεί να προσπελαστεί μέσω ενός Ενιαίου Εντοπιστή Πόρων (Uniform Resource Locator-URL), που στην ουσία αποτελεί τη διεύθυνση ενός Ιστοτόπου στον Παγκόσμιο Ιστό. Οι περισσότερες Ιστοσελίδες αποτελούνται από ένα βασικό HTML αρχείο και διάφορα σχετικά αντικείμενα. Έστω μία Ιστοσελίδα που περιέχει ένα HTML αρχείο και τέσσερα αρχεία εικόνας, τότε θεωρείται ότι η Σελίδα αυτή έχει συνολικά τέσσερα αντικείμενα. Το βασικό HTML αρχείο αναφέρεται στα άλλα αντικείμενα της συγκεκριμένης Σελίδας μέσω των URL των αντικειμένων. Κάθε URL αποτελείται από δύο τμήματα: το όνομα του εκάστοτε υπολογιστή υπηρεσίας (host) στον οποίο είναι αποθηκευμένο το εκάστοτε αρχείο και το όνομα του εκάστοτε μονοπατιού (path) του εκάστοτε αντικειμένου. Έτσι, για παράδειγμα, η URL “www.sourpi.gr/index.htm” έχει ως όνομα host το “www.sourpi.gr” και ως όνομα μονοπατιού το “index.htm”.

Το εκάστοτε πρόγραμμα περιήγησης στο Διαδίκτυο (browser) (πχ Microsoft Internet Explorer (IE), Mozilla Firefox, Google Chrome, Opera, Apple Safari) είναι ο αντιπρόσωπος του Ιστού που απεικονίζει στον εκάστοτε χρήστη το περιεχόμενο μίας Ιστοσελίδας, που αυτός έχει αιτηθεί, και παρέχει πληθώρα χαρακτηριστικών πλοήγησης και παραμετροποίησης. Επίσης, στους φυλλομετρητές υλοποιείται η πλευρά του πελάτη του βασικού πρωτοκόλλου επικοινωνίας στο Διαδίκτυο, HTTP. Ένας εξυπηρετητής\διακομιστής Ιστού (Web server) αποθηκεύει τα αντικείμενα της Ιστοσελίδας, το καθένα από τα οποία έχει ως διεύθυνση μία URL. Στους εξυπηρετητές Ιστού υλοποιείται η πλευρά του εξυπηρετητή του πρωτοκόλλου HTTP.

Οι Ιστοσελίδες αλληλοσυνδέονται και μπορεί ο εκάστοτε επισκέπτης\χρήστης\πελάτης ενός Ιστοτόπου να μεταβεί από τη μία στην άλλη, κάνοντας ένα απλό κλικ, επιλέγοντας, δηλαδή, συνδέσμους που τυχόν υπάρχουν σε κάποιο από τα αντικείμενα μίας Σελίδας. Οι σύνδεσμοι προς άλλες Σελίδες εμφανίζονται συνήθως υπογραμμισμένοι και με μπλε χρώμα, για να είναι γρήγορα ξεκάθαρο στον εκάστοτε επισκέπτη ότι πρόκειται για σύνδεσμο προς άλλη Ιστοσελίδα, χωρίς, όμως, πάντα να είναι αυτό απαραίτητο.

Η κατασκευή, τώρα, των Ιστοσελίδων και άρα και των Ιστοτόπων είναι κάτι, που ειδικά σήμερα, μπορεί να επιτευχθεί πολύ εύκολα με προγράμματα που κυκλοφορούν ακόμα και δωρεάν στο

Διαδίκτυο και επιτρέπουν σε απλούς χρήστες να δημιουργήσουν εύκολα, γρήγορα και αυτοματοποιημένα προσωπικές ή ακόμα και εμπορικές Ιστοσελίδες. Από την άλλη μεριά, υπάρχουν και πολλές εταιρίες που εξειδικεύονται στη δημιουργία πιο σύνθετων, ελκυστικών και λειτουργικών Ιστοσελίδων.

Κλείνοντας τούτη την εισαγωγή, να αναφερθεί πως ο σκοπός δημιουργίας ενός Ιστοχώρου μπορεί να ποικίλει και να στοχεύει από κάτι ιδιαίτερα απλό, όπως είναι, για παράδειγμα, η προβολή στο Διαδίκτυο του προφίλ μίας εταιρείας, μέχρι κάτι ιδιαίτερα πολυσήμαντο και πολύπλοκο, όπως είναι, για παράδειγμα η πραγματοποίηση μίας εγχείρησης ενός ασθενούς μέσω ενός ρομποτό, το οποίο κατευθύνεται απομακρυσμένα από μία ομάδα ιατρών. Άλλοι λόγοι που δύναται να οδηγήσουν στην υλοποίηση ενός Ιστοτόπου είναι η επικοινωνία χρηστών από διάφορα μέρη του κόσμου μεταξύ τους μέσω του Διαδικτύου (chat rooms), η παρακολούθηση ταινιών στο Διαδίκτυο (online films), το παίξιμο διαδικτυακών παιχνιδιών (e-gaming), η τηλεεκπαίδευση (e-learning), η αγορά προϊόντων μέσω του Διαδικτύου (e-shopping), οι τραπεζικές συναλλαγές μέσω του Διαδικτύου (e-banking) και ένα σωρό ακόμα ενέργειες που αποσκοπούν στο να χαρίσουν απλότητα, ευκολία και ταχύτητα στην απαιτητική καθημερινότητα των σύγχρονου ανθρώπων.

[4] [5]

2.1.2 Οι Κατηγορίες των Ιστοτόπων

Οι δύο μεγάλες κατηγορίες Ιστοτόπων είναι οι στατικοί Ιστότοποι (static Web Sites) και οι δυναμικοί Ιστότοποι (dynamic Web Sites). Όσον αφορά την όψη/εμφάνισή τους, οι δύο προαναφερόμενες κατηγορίες δεν έχουν μεγάλες διαφορές μεταξύ τους, αλλά, σχετικά με τη λειτουργία και τις δυνατότητές τους, οι διαφορές είναι πολλές και ουσιαστικές.

Οι στατικοί Ιστότοποι αποτελούνται από μία ή περισσότερες στατικές Ιστοσελίδες, που δεν είναι τίποτε άλλο παρά μόνο απλά ηλεκτρονικά έγγραφα, τα οποία μπορούν να περιέχουν κείμενο (text), φωτογραφίες (photos), ήχο (sound), βίντεο (video), κινούμενα γραφικά (animated graphics) συνδέσμους (links), αρχεία (files) για κατέβασμα (download). Είναι κατάλληλοι κυρίως για την δημιουργία μόνιμων/στατικών παρουσιάσεων, όπου δεν υπάρχει συχνά η ανάγκη να τροποποιείται το περιεχόμενό τους. Δηλαδή, θα μπορούσε να χρησιμοποιηθεί ένας στατικός Ιστότοπος, για παράδειγμα, για την γενική παρουσίαση μίας επιχείρησης και των προϊόντων ή των υπηρεσιών της. Ο χαρακτηρισμός «στατικός», δεν σημαίνει πως τίποτα στον Ιστοχώρο δεν «κουνιέται», αντίθετα μπορεί να έχει κινούμενα γραφικά και άλλα «μπιχλιμπιδία». Σημαίνει ότι το περιεχόμενο των Σελίδων που τον αποτελούν είναι σταθερό και συγκεκριμένο. Πρόκειται, με άλλα λόγια για σταθερά ηλεκτρονικά έγγραφα.

Το κύριο μειονέκτημά των στατικών Ιστοτόπων είναι ότι δεν είναι πρακτική η χρήση τους, από τη στιγμή που χρειάζεται το περιεχόμενό τους να αλλάζει συχνά (πχ τουλάχιστον μία φορά την εβδομάδα να προστίθεται κάποιο άρθρο ή να αλλάζουν τα προϊόντα σε έναν Τόπο που προβάλλει μία επιχείρηση). Αυτό συμβαίνει, γιατί, για να τροποποιηθεί ένας στατικός Ιστότοπος, πρέπει να γίνουν επεμβάσεις απευθείας πάνω του, με κάποιο σχετικό πρόγραμμα (πχ HTML επεξεργαστή είτε απλό είτε τύπου WYSIWYG (“What You See Is What You Get”), μία διαδικασία, δηλαδή, όπως το να σβήνει κάποιος και να γράφει σε ένα τετράδιο, που, ωστόσο, χρειάζεται μία σχετική εμπειρία.

Οι δυναμικοί Ιστότοποι, μπορεί στην εμφάνιση, σε πολλές περιπτώσεις, να μην έχουν μεγάλη διαφορά με τους στατικούς, όμως οι δυνατότητές τους είναι από πολλές πλευρές πολύ περισσότερες, καθώς στην περίπτωση αυτή ουσιαστικά πρόκειται για ολοκληρωμένες εφαρμογές, και όχι για απλά ηλεκτρονικά έγγραφα. Συνήθως οι δυναμικοί Ιστοχώροι χρησιμοποιούν κάποια βάση δεδομένων (database), όπου αποθηκεύουν πληροφορίες και από όπου αντλούν το περιεχόμενό τους ανάλογα με το τι ζητάει ο εκάστοτε πελάτης σε κάθε του κλικ. Η εκμετάλλευση των βάσεων δεδομένων είναι αυτή που επιτρέπει την εύκολη προσθαφαίρεση περιεχομένου στους δυναμικούς Ιστοτόπους, καθώς δεν απαιτείται να επεξεργάζεται κανείς κάθε φορά την ίδια την Ιστοσελίδα, αλλά απλά να διαχειρίζεται έμμεσα το περιεχόμενο της κατάλληλης σε κάθε φάση βάσης δεδομένων και οι υπόλοιπες διαδικασίες γίνονται αυτοματοποιημένα από τον αρμόδιο μηχανισμό του εκάστοτε Ιστοχώρου.

Για την εύκολη διαχείριση του περιεχομένου σε ένα δυναμικό Ιστότοπο, διατίθεται ένα Σύστημα Διαχείρισης Περιεχομένου (Content Management System-CMS) μέσω του οποίου η προσθαφαίρεση περιεχομένου στο συγκεκριμένο Τόπο μπορεί να γίνει ακόμα και από κάποιον αρχάριο. Φυσικά, η περιοχή διαχείρισης αυτού του Χώρου προστατεύεται με κωδικό πρόσβασης (password) και δεν μπορούν να εισέλθουν σε αυτή οποιοδήποτε επισκέπτες της.

[6] [7]

2.1.3 Σύγκριση των Στατικών και των Δυναμικών Ιστοτόπων

Γενικά, αυτό που προσφέρουν οι δυναμικοί Ιστοτόποι, είναι μεγαλύτερη αλληλεπίδραση του εκάστοτε χρήστη με αυτούς (πχ ένας χρήστης μπορεί να προσθέτει τα σχόλιά του σε έναν Τόπο) και ευκολίες στον εκάστοτε διαχειριστή του περιεχομένου (πχ του ιδιοκτήτη) ενός Ιστοχώρου, εφόσον, λόγω της χρήσης ενός CMS, οι προσθήκες επιμέρους Ιστοσελίδων σε έναν Ιστότοπο, ή οι διαγραφές επιμέρους Ιστοσελίδων από έναν Ιστότοπο, ή οι προσθήκες αντικειμένων στις επιμέρους Ιστοσελίδες ενός Ιστοτόπου, ή οι διαγραφές αντικειμένων από τις επιμέρους Ιστοσελίδες ενός Ιστοτόπου γίνονται απλά και γρήγορα.

Επιπρόσθετα, οι δυναμικοί Ιστοτόποι, εξαιτίας της συχνής ενημέρωσης του περιεχομένου τους, παρουσιάζουν φιλικότητα προς τον όρο Βελτιστοποίηση Ιστοσελίδων για τις Μηχανές Αναζήτησης (Search Engine Optimization-SEO). Έτσι, κατατάσσονται σε υψηλότερη θέση στις Σελίδες Αποτελεσμάτων Μηχανών Αναζήτησης (Search Engines Results Pages-SERPs) σε σχέση με τους στατικούς Ιστοτόπους, άρα εμφανίζουν και αυξημένη επισκεψιμότητα, αποκτώντας έτσι σαφές προβάδισμα έναντι των τελευταίων όσον αφορά την προώθησή τους και κατά επέκταση την προσέλκυση δυνητικών πελατών.

Πάντως το γεγονός ότι ένας δυναμικός Ιστοχώρος προσφέρει περισσότερες δυνατότητες, δεν σημαίνει ότι αυτές είναι απαραίτητες σε όλους. Δηλαδή, σε αρκετές περιπτώσεις, ένας στατικός Ιστότοπος μπορεί να καλύπτει πλήρως τις ανάγκες μίας συνοπτικής παρουσίασης. Από πλευράς κόστους, ένας στατικός Χώρος είθισται να είναι η φτηνή επιλογή, διότι είναι πιο απλή η κατασκευή του, ενώ οι δυναμικοί Τόποι, λόγω της πολυπλοκότητάς τους, κοστίζουν ακριβότερα και αυτό είναι λογικό, τουλάχιστον, όταν γίνονται κατά παραγγελία. Υπάρχει βέβαια και η περίπτωση υλοποίησης ενός δυναμικού Ιστοχώρου με χρήση κάποιας εφαρμογής ανοιχτού κώδικα (open source), η οποία διατίθεται δωρεάν μέσω του Διαδικτύου, και σε αυτήν την περίπτωση η κατασκευή ενός Ιστοτόπου μπορεί να έχει ακόμα και μηδενικό κόστος.

Επιπλέον, οι στατικοί Ιστοτόποι φορτώνονται από τον εκάστοτε πλοηγό στο Διαδίκτυο με μικρότερη καθυστέρηση σε σχέση με τους δυναμικούς, οπότε και προτιμούνται σε περιοχές με χαμηλές διαδικτυακές ταχύτητες. Αυτό συμβαίνει, επειδή οι δυναμικοί Τόποι συνδέονται με μία βάση δεδομένων αποθηκευμένη σε κάποιο εξυπηρετητή Ιστού.

Από τα παραπάνω, μπορεί κάποιος να αξιολογήσει τα υπέρ και τα κατά των Στατικών και των Δυναμικών Ιστοτόπων και τις διαφορές που αυτοί εμφανίζουν. Ωστόσο, η τάση της αγοράς δείχνει μία αυξανόμενη ζήτηση για συστήματα διαχείρισης προσαρμοσμένου περιεχόμενου Δυναμικών Ιστοχώρων και λιγότερο προς την δημιουργία στατικών. Για να πάρει την τελική απόφαση κάποιος που επιθυμεί την κατασκευή ενός Ιστοτόπου, τι είδους Ιστοχώρο χρειάζεται (στατικό ή δυναμικό), οφείλει καταρχήν να αποσαφηνίσει τον κύριο λόγο δημιουργίας του Ιστοτόπου και στη συνέχεια να αξιολογήσει ποια κατηγορία Ιστοτόπου (στατικός ή δυναμικός) θα του δώσει μακροπρόθεσμα την καλύτερη αξία για την επένδυσή του. Σε κάθε περίπτωση η γνώμη ενός ειδικού σε θέματα του Παγκόσμιου Ιστού μόνο χρησιμη μπορεί να φανεί.

[6] [7]

2.2 Ο Βασικός Όρος, Εξυπηρετητής Ιστού

2.2.1 Εισαγωγή στους Εξυπηρετητές Ιστού

Ο όρος εξυπηρετητής\διακομιστής Ιστού (Web Server) αναφέρεται είτε σε ένα ολοκληρωμένο υπολογιστικό σύστημα, είτε σε συγκεκριμένο λογισμικό, είτε ακόμα και σε μία εξειδικευμένη εφαρμογή που δέχεται και εποπτεύει αιτήσεις του πρωτοκόλλου HTTP. Τις αιτήσεις αυτές τις δημιουργεί ο εκάστοτε πελάτης (client) ενός Ιστοτόπου, ο οποίος αιτείται περιεχόμενο από τον Ιστοχώρο μέσω του εκάστοτε προγράμματος περιήγησης, που χρησιμοποιεί για να πλοηγηθεί στο Διαδίκτυο.

Οι υπολογιστές που χρησιμοποιούνται ως εξυπηρετητές Ιστού πρέπει να μπορούν να δουλεύουν όλη την ημέρα, συνεχώς, χωρίς διακοπές και να εξυπηρετούν με όσο το δυνατό μεγαλύτερη ταχύτητα γίνεται τους πελάτες τους. Για το λόγο αυτόν οι εταιρείες κατασκευής υπολογιστών και τμημάτων υπολογιστών προσφέρουν ξεχωριστές κατασκευές για εξυπηρετητές, εξαιρετικά προηγμένες τεχνολογικά και σε υψηλό κόστος. Φυσικά, αυτές οι κατασκευές μπορούν να χρησιμοποιηθούν και για προσωπικούς

υπολογιστές, όμως, τις περισσότερες φορές το κόστος αγοράς τους είναι απαγορευτικό για τους μέσους χρήστες.

Έτσι, συνηθισμένα χαρακτηριστικά για το υλικό (hardware) ενός εξυπηρετητή Ιστού είναι τα παρακάτω:

- Μητρικές πλακέτες με:
 - ο μεγάλη αντοχή στις θερμοκρασίες,
 - ο ενσωματωμένα συστήματα Πλεονάζουσας Παράταξης Ανεξάρτητων Δίσκων (Redundant Array of Inexpensive Disks-RAID),
 - ο ενσωματωμένη κάρτα οθόνης μικρής μνήμης και ταχύτητας,
 - ο άνετα κατανεμημένα στοιχεία μνήμης και επεξεργαστή, ώστε να είναι δυνατή η όσο το δυνατό καλύτερη ψύξη τους,
 - ο λιγότερες θύρες επέκτασης,
 - ο περισσότερα ενδεικτικά στοιχεία λειτουργίας,
 - ο δυνατότητα χρήσης μνήμης με Κωδικοποίηση Διόρθωσης Λάθους (Error Correcting\Correction Code-ECC).
- Κουτιά:
 - ο βαριά,
 - ο με ειδικό κλείδωμα,
 - ο θέσεις για περισσότερα ανεμιστηράκια,
 - ο ειδικά διαμορφωμένους εσωτερικούς χώρους, ώστε να γίνεται πιο καλή κυκλοφορία του αέρα,
 - ο με δυνατότητα ενσωμάτωσης σε επιτοίχια μεταλλικά ντουλάπια που φιλοξενούν τον ενεργητικό και παθητικό τηλεπικοινωνιακό εξοπλισμός ενός δικτύου (racks).
- Δίσκοι:
 - ο με προδιαγραφές για μεγαλύτερο συνεχόμενο χρόνο χρήσης,
 - ο με δυνατότητα αποσύνδεσης την ώρα λειτουργίας τους.

Επιπρόσθετα, τα περισσότερα λειτουργικά συστήματα έχουν ειδική έκδοση για χρήση ως εξυπηρετητές Ιστού ή μπορούν να διαμορφωθούν για το σκοπό αυτόν εκ των υστέρων, εκτελώντας λογισμικό (software) που κάνει την εξυπηρέτηση.

Οι περισσότεροι διακομιστές Ιστού αναλαμβάνουν πολύ δουλειά και είναι κατάλληλα οργανωμένοι με χαρακτηριστικά που τους διαφοροποιούν από τους απλούς εξυπηρετητές ενός τοπικού δικτύου, τα οποία είναι τα επόμενα:

- Δυνατότητα λειτουργίας χωρίς την ύπαρξη:
 - ο πληκτρολογίου,
 - ο οθόνης,
 - ο γραφικού περιβάλλοντος,
 - ο κάρτας ήχου.
- Δυνατότητες επιλογής χρόνου επεξεργασίας των διάφορων προγραμμάτων.
- Πρόγραμμα επικοινωνίας με συσκευές Αδιάλειπτης Παροχής Ενέργειας (Uninterruptible Power Supply-UPS).
- Δυνατότητα χρήσης περισσότερων του ενός επεξεργαστών.
- Δυνατότητα συνεπεξεργασίας με άλλους υπολογιστές-εξυπηρετητές Ιστού.
- Περισσότερες δυνατότητες αλλαγής του υλικού και αναβάθμισης του λογισμικού συστημάτων που τους φιλοξενούν, χωρίς την ανάγκη επανεκκίνησης των συστημάτων.
- Περισσότερα συστήματα ασφαλείας.

[8]

2.2.2 Ο Εξυπηρετητής Ιστού Apache Tomcat

Ένας από τους δημοφιλέστερους εξυπηρετητές Ιστού, αν όχι ο πιο δημοφιλής, είναι ο Apache Tomcat ή απλά Tomcat της εταιρείας ASF (Apache Software Foundation). Ο κύριος λόγος που ο συγκεκριμένος διακομιστής είναι ιδιαίτερα γνωστός είναι ότι διατίθεται δωρεάν, είναι πολύ γρήγορος και σταθερός. Είναι προϊόν ενός έργου ανοικτού κώδικα της εταιρείας ASF, η οποία διανέμει μία ποικιλία λογισμικών για το Διαδίκτυο σε οποιονδήποτε έχει τη δυνατότητα να τα μεταφορτώσει.

Ο εξυπηρετητής Ιστού Apache Tomcat μπορεί να εγκατασταθεί στα λειτουργικά συστήματα UNIX, LINUX και Windows. Η εγκατάσταση του Tomcat είναι σχετικά εύκολη. Ένα πρόγραμμα εγκατάστασης οδηγεί τον εκάστοτε χρήστη κατά τη βασική εγκατάσταση. Στη συνέχεια, ο χρήστης μπορεί να αλλάξει διάφορα στοιχεία που βρίσκονται στο αρχείο διαμόρφωσης. Αυτά περιλαμβάνουν: το όνομα του διακομιστή, την IP (Internet Protocol) (απόδοση στα Ελληνικά: Πρωτόκολλο Διαδικτύου) διεύθυνση που βρίσκεται ο εξυπηρετητής, τη θύρα ή τις θύρες που παρακολουθεί ο διακομιστής (συνήθως η 80), το αν ο εξυπηρετητής είναι αυτοδύναμος (η συνηθέστερη διαμόρφωση) ή αν άλλο λογισμικό διαχειρίζεται τις συνδέσεις του διακομιστή, τη διεύθυνση ηλεκτρονικού ταχυδρομείου του εξυπηρετητή, τον κατάλογο αποθήκευσης των σημαντικών αρχείων του εξυπηρετητή, τον κατάλογο αποθήκευσης των αρχείων καταγραφών του διακομιστή, ακόμα και τον κατάλογο των Ιστοσελίδων που θα διανέμει ο εξυπηρετητής. Μόλις διαμορφωθούν τα βασικά χαρακτηριστικά του διακομιστή, αυτός είναι έτοιμος να λειτουργήσει. Ο έλεγχος λειτουργίας του μπορεί να γίνει με το αν ο εκάστοτε περιηγητής δύναται να φορτώσει μία από τις Σελίδες που υπάρχουν αποθηκευμένες στον κατάλογο των Σελίδων του εξυπηρετητή.

Μετά τη βασική διαμόρφωση του εξυπηρετητή Ιστού Apache Tomcat, υπάρχουν πολλές τροποποιήσεις που μπορούν να γίνουν στο διακομιστή, οι πιο σπουδαίες από τις οποίες παρουσιάζονται παρακάτω:

- Πρόσθεση αρθρωμάτων στο διακομιστή τρίτων εταίρων, όπως είναι, για παράδειγμα, αρθρώματα που ασχολούνται με νέες τεχνολογίες.
- Διαμόρφωση του εξυπηρετητή, ώστε να μπορεί να φιλοξενεί πολλούς Ιστοτόπους.
- Περιορισμός της πρόσβασης στον διακομιστή, έτσι ώστε μερικές HTTP εντολές να μην επιτρέπεται να προσπελαίνουν ορισμένους πόρους του διακομιστή.
- Περιορισμός της πρόσβασης στον εξυπηρετητή από συγκεκριμένους υπολογιστές με βάση τις IP διευθύνεις τους.
- Διαμόρφωση του αρχείου “robots.txt” του διακομιστή που περιγράφει τις επιθυμίες του σχετικά με την εξωτερική ευρετηρίαση ενός Ιστοτόπου. Οι Ιστοτόποι δέχονται συνεχώς επισκέψεις από λογισμικούς πράκτορες, γνωστούς ως robots, οι οποίοι ευρετηριάζουν καθέναν από αυτούς, για λογαριασμό κάποιας μηχανής αναζήτησης. Το αρχείο “robots.txt” καθορίζει τι πρέπει να ευρετηριαστεί. Έτσι, για παράδειγμα, Ιστοχώροι υπό κατασκευή δεν πρέπει να ευρετηριαστούν.
- Διαμόρφωση του εξυπηρετητή, ώστε να μπορεί να διαθέτει δυναμικό περιεχόμενο. Ο διακομιστής ιστού Apache Tomcat δύναται να διαμορφωθεί, προκειμένου να διαχειρίζεται διάφορες τεχνολογίες δυναμικών Ιστοτόπων, όπως είναι, για παράδειγμα, η JSP και η PHP.
- Φόρτωση και αποθήκευση προγραμμάτων στον εξυπηρετητή που πρέπει να εκτελεστούν κατά την επεξεργασία μίας Ιστοσελίδας. Συνήθως αυτά τα προγράμματα συνδέονται με φόρμες του Παγκόσμιου Ιστού και βρίσκονται στον κατάλογο “cgi-bin” του διακομιστή. Όταν μία φόρμα υποβάλλεται από έναν περιηγητή, τότε εκτελείται κάποιος κώδικας και γίνεται κάποια επεξεργασία, όπως είναι, για παράδειγμα, η ανάκτηση ή η αποθήκευση δεδομένων μίας σχεσιακής βάσης δεδομένων.
- Βελτίωση της απόδοσης του διακομιστή, για παράδειγμα, ορίζοντας το μέγεθος της κρυφής του μνήμης και ποιες Ιστοσελίδες θα πρέπει να φορτώνονται σε αυτήν. Η κρυφή μνήμη (cache memory ή απλά cache) ενός εξυπηρετητή Ιστού είναι μία γρήγορη μνήμη, όπου αποθηκεύονται οι δημοφιλείς HTML Ιστοσελίδες. Όταν ένας χρήστης ζητά μία Σελίδα που βρίσκεται στη κρυφή μνήμη, τότε η Σελίδα επιστρέφεται άμεσα, χωρίς να απαιτείται η καθυστέρηση ανάκτησης από το σκληρό δίσκο.
- Διαμόρφωση των διαφόρων αρχείων καταγραφής του εξυπηρετητή, όπου αυτός αποθηκεύει δεδομένα για την λειτουργία των Ιστοσελίδων, όπως είναι, για παράδειγμα, το σύνολο των ανακτημένων Σελίδων.
- Διαχείριση της ασφάλειας του διακομιστή, η οποία περιλαμβάνει διάφορες τροποποιήσεις αυτού. Όσο πιο ασφαλής είναι αναγκαίο να είναι ο εξυπηρετητής τόσο πιο πολλές τροποποιήσεις απαιτούνται. Αυτές περιλαμβάνουν την ταυτοποίηση χρηστών ή άλλων Ιστοτόπων και την ασφαλή μεταφορά δεδομένων πάνω από το Διαδίκτυο.

[9]

2.2.3 Σύγκριση του Εξυπηρετητή Ιστού Apache Tomcat

Στη σύγχρονη εποχή υπάρχει μία πληθώρα εξυπηρετητών Ιστού. Η ίδια η ASF, εκτός από τον Apache Tomcat, διαθέτει και άλλους αξιόλογους διακομιστές Ιστού, όπως είναι, για παράδειγμα ο Apache HTTP Server Project. Πέρα από τους διακομιστές της ASF, ευρείας χρήσης εξυπηρετητές Ιστού είναι ο IIS (Internet Information Services) της Microsoft, ο GWS (Google Web Server) της Google, ο SJSWS (Sun Java System Web Server) της Sun και αρκετοί άλλοι.

Από όλους τους εξυπηρετητές Ιστού, που απαντώνται σήμερα, αυτός που ξεχωρίζει, με διαφορά μάλιστα από τους υπολοίπους, είναι ο Apache Tomcat. Ο Tomcat, εκτός του ότι τρέχει στα περισσότερα λειτουργικά συστήματα (πχ Unix, Linux, Windows) υποστηρίζει τις τελευταίες εκδόσεις των ευρέως χρησιμοποιούμενων πρωτοκόλλων στο Διαδίκτυο (πχ HTTP/2) και υλοποιεί τις πιο σύγχρονες τεχνικές προγραμματισμού Ιστοτόπων (πχ JSP, PHP), έχει διαπιστωθεί ότι είναι πραγματικά ταχύτερος, πιο σταθερός και γενικά πληρέστερος σε σχέση με άλλους εξυπηρετητές του Παγκοσμίου Ιστού που διατίθενται στις παρούσες μέρες.

[9]

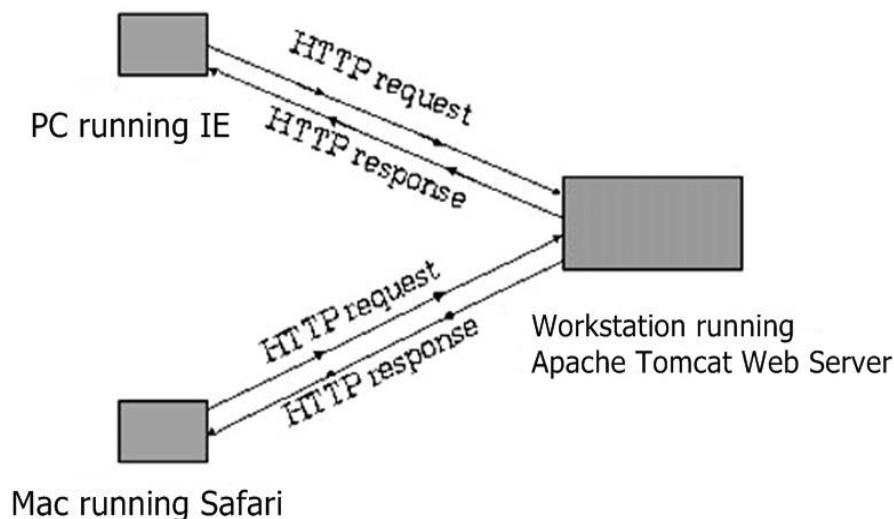
2.3 Το Πρωτόκολλο Επικοινωνίας του Διαδικτύου, HTTP

2.3.1 Εισαγωγή στο HTTP

Στις αρχές τις δεκαετίας του 1980 το Διαδίκτυο ήταν διαθέσιμο κυρίως στην ακαδημαϊκή κοινότητα. Οι υπηρεσίες που παρείχε ήταν η πρόσβαση σε έναν απομακρυσμένο υπολογιστή, το ηλεκτρονικό ταχυδρομείο (e-mail), η μεταφορά αρχείων και η αποστολή και λήψη νέων. Το 1989 στα εργαστήρια του Ευρωπαϊκού Κέντρου Πυρηνικών Ερευνών (Conseil Européenne pour la Recherche Nucléaire-CERN) προτάθηκε η ιδέα του Παγκόσμιου Ιστού, από τον φυσικό Tim Berners-Lee, ως μία αρχιτεκτονική για την επικοινωνία μεταξύ των διαφόρων ερευνητικών ομάδων του CERN. Η βασική ιδέα της αρχιτεκτονικής αυτής ήταν η διασύνδεση των εγγράφων που ήταν διασκορπισμένα στο Διαδίκτυο. Γρήγορα έγινε ιδιαίτερα δημοφιλής και γύρω του αναπτύχθηκε μία ολόκληρη βιομηχανία. Η δημοτικότητά του οφείλεται τόσο στο γραφικό περιβάλλον χρήσης του όσο και στην μεγάλη πληθώρα πληροφοριών που διαθέτει, με αποτέλεσμα ο κάθε χρήστης να μπορεί να βρει αυτό που τον ενδιαφέρει με ελάχιστο κόστος.

Το πρωτόκολλο HTTP (HyperText Transfer Protocol) (απόδοση στα Ελληνικά: Πρωτόκολλο Μεταφοράς Υπερκειμένου) είναι η καρδιά του Ιστού. Το HTTP ανήκει στο στρώμα εφαρμογών του Διαδικτύου και υλοποιείται ως δύο προγράμματα: ένα πρόγραμμα πελάτη (client program) και ένα πρόγραμμα εξυπηρετητή (server program). Τα δύο αυτά προγράμματα εκτελούνται σε διαφορετικά μηχανήματα, επικοινωνώντας μεταξύ τους με την ανταλλαγή HTTP μηνυμάτων. Συγκεκριμένα, το HTTP καθορίζει τη δομή των μηνυμάτων αυτών καθώς και τον τρόπο ανταλλαγής τους ανάμεσα σε έναν πελάτη και σε έναν εξυπηρετητή Ιστού.

Το πρωτόκολλο HTTP ορίζει τον τρόπο με τον οποίο οι πελάτες του Ιστού, δηλαδή οι πλοηγοί, ζητούν (request) Ιστοσελίδες από τους εξυπηρετητές Ιστού και πως αυτοί οι διακομιστές απαντούν (response) στις παραπάνω αιτήσεις, μεταφέροντας τις Σελίδες στους πελάτες. Η βασική ιδέα του πρωτοκόλλου αυτού φαίνεται στην παρακάτω Εικόνα:



Εικόνα 1: Η κεντρική ιδέα του HTTP

Σύμφωνα με την παραπάνω Εικόνα, από τη στιγμή που ένας χρήστης ζητά περιεχόμενο από μία Ιστοσελίδα, τότε ο εκάστοτε περιηγητής στέλνει ένα μήνυμα HTTP αίτησης (HTTP request message), για τα διάφορα αντικείμενα της Σελίδας, σε έναν εξυπηρετητή Ιστού. Ο εξυπηρετητής αυτός, τώρα, όταν λάβει το μήνυμα αυτό, ανταποκρίνεται με μηνύματα HTTP απόκρισης (HTTP response messages), στα οποία περιέχονται τα αιτούμενα αντικείμενα

Το πρωτόκολλο HTTP χρησιμοποιεί το TCP (Transmission Control Protocol) (απόδοση στα Ελληνικά: Πρωτόκολλο Ελέγχου Μεταφοράς) ως πρωτόκολλο μεταφοράς. Εφόσον ένας πελάτης εγκαταστήσει μία σύνδεση TCP με έναν εξυπηρετητή Ιστού, αρχίζει την αποστολή μηνυμάτων-αιτήσεων προς αυτόν και τη λήψη μηνυμάτων-αποκρίσεων από αυτόν. Λόγω της χρήσης του TCP, το HTTP δε χρειάζεται να ασχοληθεί καθόλου με τη μεταφορά των δεδομένων. Το μόνο που πρέπει να κάνει είναι να στείλει τις αιτήσεις μέσω της TCP σύνδεσης και να περιμένει τις αποκρίσεις. Το TCP εγγυάται την αξιόπιστη μεταφορά των δεδομένων καθώς και τον έλεγχο της συμφόρησης του εκάστοτε δικτύου.

Οι εξυπηρετητές Ιστού του πρωτοκόλλου HTTP δεν κρατάνε καθόλου στοιχεία για την κατάσταση των πελατών τους. Επομένως, αν ένας πελάτης στείλει μία αίτηση για ένα αρχείο δύο φορές, τότε ο εξυπηρετητής θα του στείλει το αρχείο αυτό δύο φορές. Τα πρωτόκολλα που δεν κρατάνε καθόλου πληροφορία για την κατάσταση του πελάτη ονομάζονται δίχως κατάσταση πρωτόκολλα (stateless protocols).

[10]

2.3.2 Οι Κατηγορίες των Συνδέσεων του HTTP

Το πρωτόκολλο HTTP μπορεί να χρησιμοποιεί τόσο μόνιμες (persistent) όσο και μη μόνιμες (non-persistent) συνδέσεις.

Στις μόνιμες συνδέσεις του πρωτοκόλλου HTTP ο εκάστοτε εξυπηρετητής Ιστού δεν τερματίζει την υφιστάμενη σε κάθε περίπτωση TCP σύνδεση, αμέσως μετά την αποστολή ενός αντικειμένου μίας Ιστοσελίδας, σε ένα χρήστη, ο οποίος αιτήθηκε το συγκεκριμένο αντικείμενο. Αντίθετα, την τερματίζει μετά από κάποιο συγκεκριμένο χρονικό διάστημα κατά το οποίο η πιο πάνω σύνδεση είναι ανενεργή, δηλαδή δεν γίνεται μεταφορά δεδομένων. Μέσα στο χρονικό διάστημα αυτό ο προαναφερόμενος χρήστης έχει τη δυνατότητα να ζητήσει την αποστολή και άλλων αντικειμένων από την εκάστοτε τρέχουσα Ιστοσελίδα, μέσω του κατάλληλου κάθε φορά φυλλομετρητή αλλά και την αποστολή και άλλων Ιστοσελίδων από τον εκάστοτε τρέχων εξυπηρετητή.

Το πρωτόκολλο HTTP υποστηρίζει δύο τύπους μόνιμων συνδέσεων: με pipelining και χωρίς pipelining. Στην πρώτη περίπτωση ένας πελάτης στέλνει μία καινούργια αίτηση μόνο μετά τη λήψη της εκάστοτε προηγούμενης απάντησης, οπότε το κάθε αντικείμενο της εκάστοτε παρούσας Ιστοσελίδας υπόκειται σε έναν επιπλέον χρόνο RTT. Ο τρέχων κάθε στιγμή χρόνος απόκρισης μπορεί να βελτιωθεί

ακόμα περισσότερο, αν χρησιμοποιηθούν μόνιμες συνδέσεις με pipelining. Επιπλέον, η χρήση pipelining καλύπτει και ένα άλλο μειονέκτημα που έχει η απουσία pipelining. Το μειονέκτημα αυτό είναι ο χρόνος αναμονής της λήψης απόκρισης από τους πελάτες και τη λήψη αιτήσεων από τους εξυπηρετητές Ιστού, κατά τον οποίο γίνεται σπατάλη πόρων, οι οποίοι θα μπορούσαν να χρησιμοποιηθούν σε άλλους σκοπούς

Στις μη-μόνιμες συνδέσεις του πρωτοκόλλου HTTP ο εκάστοτε διακομιστής Ιστού τερματίζει την υπάρχουσα σε κάθε περίπτωση σύνδεση, αμέσως μετά την αποστολή ενός αντικειμένου μίας Ιστοσελίδας, σε ένα χρήστη, ο οποίος αιτήθηκε αυτό το αντικείμενο. Προκειμένου να λάβει ο συγκεκριμένο χρήστης κάποιο άλλο αντικείμενο της προαναφερόμενης Ιστοσελίδας, θα πρέπει, κατόπιν νέας αίτησής του προς τον παραπάνω διακομιστή, να εγκαθιδρυθεί μία νέα TCP σύνδεση, ώστε να του αποστείλει ο τελευταίος το καινούργιο περιεχόμενο.

Παρά το γεγονός ότι στις HTTP συνδέσεις μη μόνιμου χαρακτήρα πρέπει να εγκατασταθούν τόσες διαφορετικές TCP συνδέσεις όσα είναι τα αντικείμενα μίας Ιστοσελίδας, που είναι επιθυμητό να μεταφερθούν σε κάθε περίπτωση στον εκάστοτε χρήστη, οι σύγχρονοι περιηγητές υποστηρίζουν την παράλληλη μεταφορά (parallel transfer) των διαφόρων αντικειμένων μίας Σελίδας. Σε αυτή την περίπτωση όλα τα αντικείμενα μίας Σελίδας μεταφέρονται παράλληλα, με αποτέλεσμα τη μείωση του συνολικού χρόνου που χρειάζεται για τη λήψη όλης της συγκεκριμένης Σελίδας από έναν πλοηγό και την παρουσίαση του περιεχομένου της σε ένα χρήστη. Από τη στιγμή που εγκαθίστανται παράλληλες TCP συνδέσεις το εκάστοτε συνολικά διαθέσιμο εύρος ζώνης (bandwidth) μοιράζεται ανάμεσα σε αυτές τις παράλληλες συνδέσεις.

[10] [11]

2.3.3 Οι Μέθοδοι του HTTP

Αν και το πρωτόκολλο HTTP σχεδιάστηκε για χρήση στον Παγκόσμιο Ιστό, είναι εσκεμμένα πιο γενικό από ότι απαιτείται, αποσκοπώντας σε μελλοντικές αντικειμενοστραφείς εφαρμογές. Για το συγκεκριμένο λόγο υποστηρίζονται λειτουργίες, που ονομάζονται μέθοδοι (methods), διαφορετικές από την απλή αίτηση για την εκάστοτε Ιστοσελίδα. Κάθε αίτηση αποτελείται από μία ή περισσότερες γραμμές κειμένου ASCII, με την πρώτη λέξη της πρώτης γραμμής να είναι το όνομα της εκάστοτε ζητούμενης μεθόδου. Οι ενσωματωμένες μέθοδοι του HTTP φαίνονται στον παρακάτω Πίνακα:

ΜΕΘΟΔΟΣ	ΛΕΙΤΟΥΡΓΙΑ
CONNECT	Είναι δεσμευμένη για μελλοντική χρήση.
DELETE	Χρησιμοποιείται στην κατάργηση μίας Ιστοσελίδας.
GET	Χρησιμοποιείται στη λήψη περιεχομένου από μία Ιστοσελίδα.
HEAD	Χρησιμοποιείται στην ανάγνωση της κεφαλίδας του εκάστοτε HTTP μηνύματος αίτησης ή απόκρισης.
OPTIONS	Χρησιμοποιείται στον καθορισμό διαφόρων ιδιοτήτων σχετικά με τον εκάστοτε διακομιστή Ιστού, από τον οποίο ο εκάστοτε χρήστης αιτείται περιεχόμενο, ή αναφορικά με το εν λόγω περιεχόμενο.
POST	Χρησιμοποιείται στην προσθήκη νέων αντικειμένων στα ήδη υπάρχοντα μίας Ιστοσελίδας.
PUT	Χρησιμοποιείται στην τοποθέτηση περιεχομένου σε μία Ιστοσελίδα.
TRACE	Χρησιμοποιείται στην παρακολούθηση του εκάστοτε HTTP μηνύματος αίτησης.

Πίνακας 1: Οι μέθοδοι του HTTP

Ακολουθεί μία λεπτομερέστερη ανάλυση των μεθόδων του πρωτοκόλλου HTTP που αναφέρονται στον παραπάνω Πίνακα:

- Η μέθοδος CONNECT (απόδοση στα Ελληνικά: ΣΥΝΔΕΣΗ) είναι δεσμευμένη για μελλοντική χρήση.
- Η μέθοδος DELETE (απόδοση στα Ελληνικά: ΔΙΑΓΡΑΦΗ) κάνει το αναμενόμενο: καταργεί μία Ιστοσελίδα. Στην παρούσα μέθοδο παίζουν μεγάλο ρόλο η πιστοποίηση της ταυτότητας του εκάστοτε χρήστη που αιτείται να καταργήσει μία Σελίδα και της άδειας εκτέλεσης της συγκεκριμένης λειτουργίας. Δεν υπάρχει εγγύηση πως η μέθοδος DELETE θα επιτύχει, αφού,

ακόμα και αν ο εκάστοτε απομακρυσμένος HTTP διακομιστής είναι πρόθυμος να διαγράψει μία Σελίδα, η εν λόγω Σελίδα μπορεί να έχει ιδιότητες οι οποίες να απαγορεύουν στο διακομιστή αυτόν να την τροποποιήσει ή να την καταργήσει.

- Η μέθοδος GET (απόδοση στα Ελληνικά: ΛΗΨΗ) ζητά από έναν εξυπηρετητή Ιστού να στείλει το περιεχόμενο μίας Ιστοσελίδας. Η Σελίδα αυτή κωδικοποιείται κατάλληλα σε μορφή MIME (Multipurpose Internet Mail Extensions). Η πλειοψηφία των αιτήσεων προς τους διακομιστές Ιστού είναι τύπου GET. Η συνηθισμένη μορφή της μεθόδου GET, είναι η εξής: GET filename HTTP/version, όπου filename είναι το όνομα του εκάστοτε πόρου-αρχείου που πρέπει να προσκομιστεί και version είναι η έκδοση του πρωτοκόλλου HTTP που χρησιμοποιείται κάθε φορά.
- Η μέθοδος HEAD (απόδοση στα Ελληνικά: ΚΕΦΑΛΙΔΑ) ζητά μόνο την κεφαλίδα ενός HTTP μηνύματος, δίχως την πραγματική Σελίδα. Η μέθοδος αυτή χρησιμοποιείται, ώστε να λαμβάνεται ο χρόνος τελευταίας τροποποίησης μίας Σελίδας για τη συλλογή πληροφοριών για λόγους δεικτοδότησης ή απλά και μόνο για τον έλεγχο εγκυρότητας μίας URL.
- Η μέθοδος OPTIONS (απόδοση στα Ελληνικά: ΕΠΙΛΟΓΕΣ) παρέχει έναν τρόπο, ώστε ο εκάστοτε πελάτης να στέλνει ερωτήματα σε ένα διακομιστή Ιστού σχετικά με τις ιδιότητες αυτού ή ενός συγκεκριμένου αρχείου που είναι αποθηκευμένο σε αυτόν.
- Η μέθοδος POST (απόδοση στα Ελληνικά: ΤΟΠΟΘΕΤΗΣΗ) προσαρτά νέα δεδομένα στα ήδη υπάρχοντα μίας Ιστοσελίδας. Η δημοσίευση ενός μηνύματος σε μία ομάδα συζητήσεων ή η προσθήκη ενός αρχείου σε ένα σύστημα ηλεκτρονικού πίνακα ανακοινώσεων είναι παραδείγματα μίας τέτοιας προσάρτησης.
- Η μέθοδος PUT (απόδοση στα Ελληνικά: ΤΟΠΟΘΕΤΗΣΗ) είναι η αντίστροφη της GET. Αντί να διαβάζει μία Ιστοσελίδα, γράφει μία Σελίδα. Η μέθοδος αυτή κάνει εφικτή την κατασκευή μίας συλλογής Ιστοσελίδων. Το σώμα της αίτησης περιέχει μία Σελίδα. Μπορεί να κωδικοποιείται μέσω του MIME, οπότε οι γραμμές που ακολουθούν την PUT, περιέχουν Content-Type κεφαλίδες και πιστοποίησης ταυτότητας, ώστε να αποδεικνύουν πως ο σε κάθε περίπτωση αιτών περιεχομένου από μία Σελίδα να έχει πραγματικά την άδεια να εκτελέσει τη ζητούμενη λειτουργία.
- Η μέθοδος TRACE (απόδοση στα Ελληνικά: ΠΑΡΑΚΟΛΟΥΘΗΣΗ) χρησιμοποιείται για τον εντοπισμό σφαλμάτων. Ζητά από έναν εξυπηρετητή Ιστού να επιστρέψει μία HTTP αίτηση. Η μέθοδος αυτή είναι χρήσιμη σε περιπτώσεις που η επεξεργασία των αιτήσεων δεν γίνεται σωστά και ο εκάστοτε πελάτης θέλει να δει ποια αίτηση έλαβε πραγματικά ο εκάστοτε διακομιστής.

Στο σημείο αυτό, να τονιστεί ότι για την προσπέλαση γενικών αντικειμένων μίας Ιστοσελίδας μπορεί να είναι διαθέσιμες και πρόσθετες εξειδικευμένες μέθοδοι που αφορούν συγκεκριμένα αντικείμενα καθώς, επίσης, και ότι στα ονόματα των HTTP μεθόδων γίνεται διάκριση πεζών και κεφαλαίων γραμμάτων. Έτσι, για παράδειγμα, η GET είναι μία έγκριτη μέθοδος, ενώ η get δεν είναι).

[12] [13]

2.3.4 Οι Κεφαλίδες Μηνυμάτων του HTTP

Η γραμμή ενός HTTP μηνύματος αίτησης μπορεί να ακολουθείται από πρόσθετες γραμμές με περισσότερες πληροφορίες. Οι γραμμές αυτές ονομάζονται κεφαλίδες αίτησης (request headers). Οι πληροφορίες αυτές είναι ανάλογες με τις παραμέτρους σε μία κλήση διαδικασίας. Επίσης, ένα HTTP μήνυμα απάντησης μπορεί να έχει και αυτό κεφαλίδες απάντησης (response headers). Μερικές κεφαλίδες μπορούν να χρησιμοποιηθούν και προς τις δύο κατευθύνσεις. Στον παρακάτω Πίνακα φαίνονται μερικές από τις βασικότερες κεφαλίδες μηνυμάτων του πρωτοκόλλου HTTP:

ΚΕΦΑΛΙΔΑ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
Accept	Αίτηση	Καθορίζει τον τύπο των Ιστοσελίδων, τον οποίο μπορεί να χειριστεί ένας πελάτης.
Accept-Charset	Αίτηση	Καθορίζει το σύνολο των χαρακτήρων των Ιστοσελίδων, τους οποίους μπορεί να χειριστεί ένας πελάτης.
Accept-Encoding	Αίτηση	Καθορίζει τις κωδικοποιήσεις των Ιστοσελίδων, τις οποίες μπορεί να χειριστεί ένας πελάτης.

Accept-Language	Αίτηση	Καθορίζει τις φυσικές γλώσσες των Ιστοσελίδων, τις οποίες μπορεί να χειριστεί ένας πελάτης.
Authorization	Αίτηση	Καθορίζει μία λίστα των πιστοποιητικών των Ιστοσελίδων, τα οποία μπορεί να χειριστεί ένας πελάτης.
Cookie	Αίτηση	Καθορίζει τα cookies που αποστέλλονται σε ένα διακομιστή Ιστού, κατά την αποστολή ενός HTTP μηνύματος αίτησης.
Host	Αίτηση	Καθορίζει το DNS όνομα ενός διακομιστή Ιστού.
User-Agent	Αίτηση	Καθορίζει τις πληροφορίες ενός φυλλομετρητή.
Accept-Ranges	Απάντηση	Καθορίζει το εύρος σε bytes ενός HTTP μηνύματος απόκρισης.
Content-Encoding	Απάντηση	Καθορίζει τον τρόπο κωδικοποίησης του περιεχομένου μίας Ιστοσελίδας.
Content-Language	Απάντηση	Καθορίζει τη φυσική γλώσσα που χρησιμοποιείται σε μία Ιστοσελίδα.
Content-Length	Απάντηση	Καθορίζει το μήκος σε bytes του περιεχομένου μίας Ιστοσελίδας, το οποίο περιέχεται σε ένα HTTP μήνυμα απόκρισης.
Content-Type	Απάντηση	Καθορίζει τον τύπο MIME μίας Ιστοσελίδας.
Last-Modified	Απάντηση	Καθορίζει την ώρα και την ημερομηνία της τελευταίας αλλαγής μίας Ιστοσελίδας.
Location	Απάντηση	Καθορίζει την τοποθεσία του διακομιστή Ιστού, στην οποία θα κατευθύνει ένας πελάτης την HTTP αίτησή του.
Server	Απάντηση	Καθορίζει τις πληροφορίες ενός διακομιστή Ιστού.
Set-Cookie	Απάντηση	Καθορίζει νέα cookies κατά την αποστολή ενός HTTP μηνύματος απόκρισης.
Connection	Αίτηση-Απάντηση	Καθορίζει αν μία HTTP σύνδεση θα είναι μόνιμη ή όχι.
Date	Αίτηση-Απάντηση	Καθορίζει την ημερομηνία και την ώρα αποστολής ενός HTTP μηνύματος αίτησης ή απόκρισης.
Upgrade	Αίτηση-Απάντηση	Καθορίζει ένα πρωτόκολλο στο οποίο θα γίνει μετάβαση, κατά την αποστολής ενός HTTP μηνύματος αίτησης ή απόκρισης.

Πίνακας 2: Οι πιο βασικές κεφαλίδες των μηνυμάτων του HTTP.

Έπεται μία πιο διεξοδική παρουσίαση των κεφαλίδων των μηνυμάτων του πρωτοκόλλου HTTP που καταγράφονται στον παραπάνω Πίνακα:

Πρώτον, σχετικά με τις κεφαλίδες που χρησιμοποιούνται εξολοκλήρου από τους πελάτες σε μία HTTP αίτηση:

- Οι τέσσερις κεφαλίδες Accept (απόδοση στα Ελληνικά: Αποδοχή) λένε σε ένα διακομιστή Ιστού τι είναι πρόθυμος να αποδεχθεί ο υπάρχων σε κάθε περίπτωση πελάτης. Η πρώτη κεφαλίδα ορίζει τους εκάστοτε τύπους MIME που είναι ευπρόσδεκτοι (πχ text, html). Η δεύτερη καθορίζει το κάθε φορά αποδεκτό σύνολο χαρακτήρων (πχ ISO-8859-5, Unicode-1-1). Η τρίτη ασχολείται με τις δεκτές σε κάθε περίπτωση μεθόδους συμπίεσης (πχ gzip). Η τέταρτη υποδεικνύει την αποδεκτή σε κάθε φάση φυσική γλώσσα (πχ Ισπανικά). Αν ένας εξυπηρετητής δύναται να απαντήσει με ένα πλήθος Ιστοσελίδων στο αίτημα ενός πελάτη, τότε έχει τη δυνατότητα να χρησιμοποιεί τις πληροφορίες που απορρέουν από τις κεφαλίδες Accept, ώστε να επιλέξει τη Σελίδα που θέλει ο συγκεκριμένος πελάτης. Ωστόσο, αν ένας διακομιστής δεν είναι σε θέση να ικανοποιήσει το αίτημα ενός πελάτη, τότε επιστρέφει ένα κωδικός σφάλματος και το τρέχον αίτημα αποτυγχάνει.

- Η κεφαλίδα Authorization (απόδοση στα Ελληνικά: Εξουσιοδότηση) απαιτείται για τις Ιστοσελίδες που είναι προστατευμένες. Στην περίπτωση αυτή, ο εκάστοτε πελάτης χρειάζεται να αποδείξει ότι έχει το δικαίωμα να δει την εκάστοτε ζητούμενη Σελίδα και για αυτόν ακριβώς το λόγο είναι διαθέσιμη τούτη η επικεφαλίδα.
- Η κεφαλίδα Cookie χρησιμοποιείται από τους πελάτες, για να επιστρέφουν στον κατάλληλο σε κάθε περίπτωση διακομιστή Ιστού ένα cookie, που τους στάλθηκε στο παρελθόν από κάποια μηχανή της περιοχής αυτού του εξυπηρετητή. Τα cookies είναι μικρά αρχεία κειμένου τα οποία αποθηκεύονται στον εκάστοτε φυλλομετρητή κατά την πλοήγησή ενός χρήστη στο Διαδίκτυο. Σκοπός τους είναι να ειδοποιούν τον εκάστοτε Ιστότοπο που επισκέπτεται ο εκάστοτε χρήστης για την προηγούμενη δραστηριότητα του συγκεκριμένου χρήστη.
- Η κεφαλίδα Host (απόδοση στα Ελληνικά: Υπολογιστής Υπηρεσίας) κατονομάζει τον διακομιστή Ιστού που κάθε φορά χρησιμοποιείται. Η πληροφορία αυτή λαμβάνεται από την εκάστοτε τρέχουσα διεύθυνση URL. Η συγκεκριμένη κεφαλίδα είναι υποχρεωτική. Χρησιμοποιείται, επειδή ορισμένες IP διευθύνσεις μπορούν να εξυπηρετούν περισσότερα από ένα DNS ονόματα, οπότε ο υφιστάμενος σε κάθε περίπτωση διακομιστής χρειάζεται κάποιον τρόπο να διακρίνει τον host (υπολογιστή υπηρεσίας) εκείνον στον οποίο θα διαβιβάσει την εκάστοτε αίτηση.
- Η κεφαλίδα User-Agent (απόδοση στα Ελληνικά: Πράκτορας-Χρήστης) επιτρέπει σε έναν πελάτη να πληροφορεί τον εκάστοτε εξυπηρετητή Ιστού σχετικά με το εκμεταλλεμένο σε κάθε περίπτωση φυλλομετρητή, το λειτουργικό σύστημα αυτού και τις άλλες ιδιότητές του.

Δεύτερον, αναφορικά με τις κεφαλίδες που εκμεταλλεύονται αποκλειστικά οι εξυπηρετητές Ιστού σε μία HTTP απάντηση:

- Αν μία Ιστοσελίδα είναι πολύ μεγάλη, τότε ένας μικρός πελάτης μπορεί να μη θέλει να τη λάβει ολόκληρη. Ορισμένοι εξυπηρετητές Ιστού δέχονται αιτήσεις για εύρος σε bytes, έτσι ώστε μία Σελίδα να μπορεί να προσκομιστεί σε πολλά μικρά τμήματα. Η κεφαλίδα Accept-Ranges (απόδοση στα Ελληνικά: Αποδοχή Εύρους) ανακοινώνει την προθυμία κάποιου διακομιστή να χειριστεί αυτόν τον τύπο αποκρίσεων για τμήματα Σελίδων.
- Οι τέσσερις επόμενες κεφαλίδες, οι οποίες ξεκινούν όλες με τη λέξη Content (απόδοση στα Ελληνικά: Περιεχόμενο), επιτρέπουν στους εξυπηρετητές Ιστού να περιγράφουν τις ιδιότητες της Ιστοσελίδας που στέλνουν.
- Η κεφαλίδα Last-Modified (απόδοση στα Ελληνικά: Τελευταία Τροποποιημένη) λέει πότε τροποποιήθηκε για τελευταία φορά μία Ιστοσελίδα. Η κεφαλίδα αυτή παίζει σημαντικό ρόλο στην αποθήκευση των Σελίδων σε κρυφή μνήμη.
- Η κεφαλίδα Location (απόδοση στα Ελληνικά: Τοποθεσία) χρησιμοποιείται από ένα διακομιστή Ιστού, προκειμένου να ενημερώσει τον εκάστοτε πελάτη ότι θα πρέπει να δοκιμάσει μία διαφορετική διεύθυνση URL. Χρησιμοποιείται, αν μία Ιστοσελίδα έχει μετακινηθεί, ή, όταν είναι αναγκαία η αναφορά σε μία ίδια Σελίδα (πιθανόν σε διαφορετικούς εξυπηρετητές) περισσότερων από μία διευθύνσεων URL. Χρησιμοποιείται, επιπλέον, από εταιρείες που έχουν μία κύρια Ιστοσελίδα στην περιοχή com, αλλά ανακατευθύνουν τους πελάτες σε μία εθνική ή περιφερειακή Σελίδα, ανάλογα με την IP διεύθυνση τους ή την προτιμώμενη γλώσσα τους.
- Η κεφαλίδα Server (απόδοση στα Ελληνικά: Εξυπηρετητής) επιτρέπει σε ένα διακομιστή Ιστού να πει ποιος είναι και να προσδιορίσει, αν επιθυμεί, ορισμένες από τις ιδιότητές του.
- Η κεφαλίδα Set-Cookie (απόδοση στα Ελληνικά: Ορισμός Cookie) είναι ο τρόπος με τον οποίο οι διακομιστές Ιστού στέλνουν cookies στους εκάστοτε πελάτες, οι οποίοι με τη σειρά τους αναμένεται να τα αποθηκεύσουν και να τους τα επιστρέψουν στις επόμενες αιτήσεις.

Τρίτον, σε σχέση με τις κεφαλίδες που χρησιμοποιούν τόσο οι πελάτες σε μία HTTP αίτηση όσο και οι εξυπηρετητές Ιστού σε μία HTTP απάντηση:

- Η κεφαλίδα Connection (απόδοση στα Ελληνικά: Σύνδεση) καθορίζει αν η υφιστάμενη κάθε φορά HTTP σύνδεση είναι μόνιμη ή όχι.
- Η κεφαλίδα Date (απόδοση στα Ελληνικά: Ημερομηνία) περιέχει την ημερομηνία και την ώρα που στάλθηκε το εκάστοτε HTTP μήνυμα αίτησης ή απάντησης.
- Η κεφαλίδα Upgrade (απόδοση στα Ελληνικά: Αναβάθμιση) είναι διαθέσιμη, για να διευκολυνθεί η μετάβαση σε μία μελλοντική (πιθανόν μη συμβατή) έκδοση του πρωτοκόλλου HTTP. Επιτρέπει στον υπαρκτό κάθε φορά πελάτη να ανακοινώνει τι πρωτόκολλο μπορεί να στηρίξει

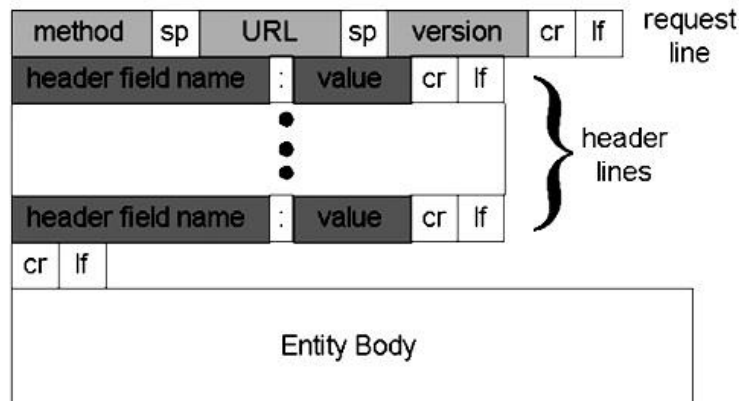
και στον υφιστάμενο κάθε στιγμή διακομιστή Ιστού να βεβαιώνει τι πρωτόκολλο μπορεί να χρησιμοποιεί.

[12] [14]

2.3.5 Η Δομή ενός Μηνύματος του HTTP

Το πρωτόκολλο HTTP ορίζει μόνο δύο τύπους μηνυμάτων: τις HTTP αιτήσεις (requests) και τις HTTP αποκρίσεις\απαντήσεις (responses).

Η δομή ενός HTTP μηνύματος αίτησης φαίνεται στην παρακάτω Εικόνα:



Εικόνα 2: Η δομή ενός μηνύματος αίτησης του HTTP

Σύμφωνα με την παραπάνω Εικόνα, ένα HTTP μήνυμα αίτησης αποτελείται από τρία μέρη: τη γραμμή αίτησης (request line), τις γραμμές επικεφαλίδας (header lines) και το τμήμα περιεχομένου (entity body).

Η γραμμή αίτησης περιέχει τρία πεδία: το πεδίο μεθόδου (method), το πεδίο URL και το πεδίο HTTP έκδοσης (version). Το πεδίο μεθόδου μπορεί να έχει μία από τις μεθόδους που χρησιμοποιεί το HTTP. Η πιο συνηθισμένη μέθοδος στις HTTP αιτήσεις είναι η GET, με την οποία ζητείται από έναν εξυπηρετητή Ιστού η αποστολή του εκάστοτε αρχείου που εμφανίζεται στο πεδίο URL. Στο πεδίο URL δεν είναι απαραίτητο να υπάρχει και το όνομα του εκάστοτε host, αφού ήδη έχει εγκατασταθεί μία σύνδεση με το αυτόν host. Τέλος, στο πεδίο HTTP έκδοσης περιγράφεται η έκδοση του πρωτοκόλλου HTTP που χρησιμοποιεί ο προαναφερόμενος host.

Οι γραμμές επικεφαλίδας ενός HTTP μηνύματος αίτησης περιέχουν μία ή περισσότερες κεφαλίδες αίτησης (request headers) του πρωτοκόλλου HTTP. Οι κεφαλίδες αυτές παρέχουν, όπως προαναφέρθηκε, πρόσθετες πληροφορίες για την κάθε αίτηση.

Το τμήμα περιεχομένου είναι προαιρετικό και η ύπαρξη του εξαρτάται από την εκάστοτε μέθοδο που χρησιμοποιείται στο πεδίο μεθόδου. Διατίθεται, προκειμένου να αποστέλλει σε ένα διακομιστή Ιστού επιπλέον δεδομένα, που αποτελούν ουσιαστικά παραμέτρους της μεθόδου που χρησιμοποιεί αυτό το μέρος της αίτησης. Έτσι, το συγκεκριμένο πεδίο δε χρησιμοποιείται για τη μέθοδο GET, χρησιμοποιείται, όμως, αφότου γίνεται εφαρμογή της μεθόδου POST. Ένα παράδειγμα χρήσης της μεθόδου POST είναι η αποστολή των δεδομένων που συμπληρώνει ένας χρήστης σε διάφορες φόρμες μίας Ιστοσελίδας. Στην περίπτωση αυτή τα δεδομένα αποστέλλονται στο τμήμα περιεχομένου και ανάλογα με τα δεδομένα αυτά ο παραπάνω χρήστης λαμβάνει μία κατάλληλη έκδοση της προαναφερόμενης Ιστοσελίδας. Παρόμοια χρήση αυτού του πεδίου υπάρχει και στην περίπτωση χρήσης της μεθόδου HEAD.

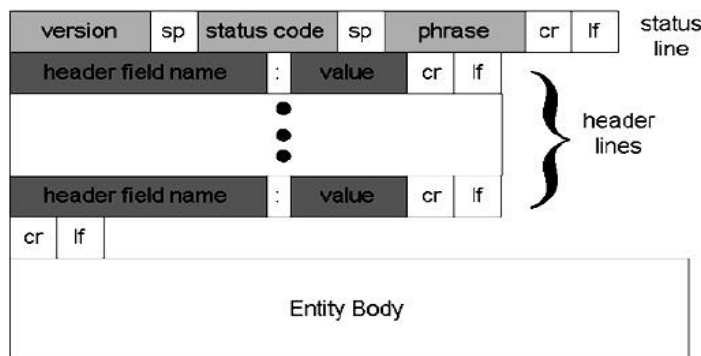
Ένα παράδειγμα HTTP μηνύματος αίτησης, που συμφωνεί με τα παραπάνω, είναι το ακόλουθο:

```
GET/lessons/index.htm HTTP/1.1
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: gr
```

(extra carriage return, line feed)

Όπως φαίνεται στο παραπάνω παράδειγμα, οι HTTP αιτήσεις (αλλά και οι HTTP αποκρίσεις, όπως θα δειχθεί παρακάτω) γράφονται με χαρακτήρες ASCII και μπορούν να διαβαστούν από τους ανθρώπους. Οι περισσότερες HTTP αιτήσεις αποτελούνται από 5 γραμμές κειμένου ακολουθούμενες από μία κενή γραμμή. Οι HTTP αιτήσεις περιέχουν τουλάχιστον μία γραμμή κειμένου, ενώ υπάρχουν και περιπτώσεις που αποτελούνται και από περισσότερες από 5 γραμμές κειμένου.

Η δομή ενός HTTP μηνύματος απόκρισης φαίνεται στην ακόλουθη Εικόνα:



Εικόνα 3: Η δομή ενός μηνύματος απόκρισης του HTTP

Σύμφωνα με την παραπάνω Εικόνα, ένα HTTP μήνυμα απάντησης αποτελείται από τρία μέρη: τη γραμμή κατάστασης (status line), τις γραμμές επικεφαλίδας (header lines) και το τμήμα περιεχομένου (entity body).

Η γραμμή κατάστασης περιέχει τρία πεδία: το πεδίο HTTP έκδοσης (version), το πεδίο κωδικού κατάστασης (status code) και το πεδίο του αντίστοιχου μηνύματος κατάστασης (phrase). Το πεδίο της HTTP έκδοσης ορίζει την έκδοση του HTTP πρωτοκόλλου που διατίθεται από τον εκάστοτε host. Το πεδίο του κωδικού κατάστασης περιέχει έναν τριψήφιο αριθμό ο οποίος δηλώνει κατά πόσο ικανοποιήθηκε η εκάστοτε αίτηση. Το πεδίο του μηνύματος κατάστασης επεξηγεί τη σημασία του αντίστοιχου κωδικού κατάστασης, που είναι ένας τριψήφιος αριθμός, το πρώτο ψηφίο του οποίου διευκολύνει τη διάκριση των HTTP μηνυμάτων απόκρισης σε πέντε κατηγορίες, όπως φαίνεται και στον επόμενο Πίνακα:

ΚΩΔΙΚΟΣ	ΣΗΜΑΣΙΑ	ΠΑΡΑΔΕΙΓΜΑΤΑ
1xx	Πληροφορίες	100 = ο διακομιστής Ιστού συμφωνεί να χειριστεί την αίτηση του πελάτη
2xx	Επιτυχία	200 = η HTTP αίτηση πέτυχε
3xx	Ανακατεύθυνση	301 = η Ιστοσελίδα μετακινήθηκε, 304 = η αποθηκευμένη στην κρυφή μνήμη Ιστοσελίδα είναι ακόμα έγκυρη
4xx	Σφάλμα πελάτη	403 = απαγορευμένη Ιστοσελίδα, 404 = η Ιστοσελίδα δε βρέθηκε
5xx	Σφάλμα διακομιστή Ιστού	500 = εσωτερικό σφάλμα διακομιστή Ιστού, 503 = ο διακομιστής Ιστού είναι προσωρινά μη διαθέσιμος, 505 = η αιτούμενη HTTP έκδοση δεν υποστηρίζεται από τον διακομιστή Ιστού

Πίνακας 3: Οι κατηγορίες των κωδικών κατάστασης των μηνυμάτων απόκρισης του HTTP

Όπως προκύπτει, λοιπόν, από τον παραπάνω Πίνακα, τα HTTP μηνύματα απόκρισης είναι πέντε ειδών, τα εξής:

- I. Όσα έχουν κωδικό της μορφής 1xx: Χρησιμοποιούνται σπάνια και σχετίζονται με πληροφορίες που έχουν να κάνουν με το χειρισμό μίας αίτησης ενός πελάτη από ένα διακομιστή Ιστού.

- II. Όσα έχουν κωδικό της μορφής 2xx: Χρησιμοποιούνται, για να δηλώσουν πως ο χειρισμός μίας αίτησης έγινε με επιτυχία και αναμένεται να επιστραφεί περιεχόμενο, αν υπάρχει.
- III. Όσα έχουν κωδικό της μορφής 3xx: Χρησιμοποιούνται, για να ειδοποιήσουν έναν πελάτη να ψάξει το περιεχόμενο που αναζητά είτε σε νέα URL είτε εξετάζοντας την κρυφή του μνήμη.
- IV. Όσα έχουν κωδικό της μορφής 4xx: Χρησιμοποιούνται, για να σημάνουν πως μία αίτηση απέτυχε, λόγω κάποιου σφάλματος ενός πελάτη, όπως είναι, για παράδειγμα, μία άκυρη αίτηση ή μία ανύπαρκτη Ιστοσελίδα.
- V. Όσα έχουν κωδικό της μορφής 5xx: Χρησιμοποιούνται, για να υποδηλώσουν ότι ένας διακομιστής Ιστού έχει πρόβλημα, είτε εξαιτίας ενός σφάλματος στον κώδικά του, είτε λόγω προσωρινής υπερφόρτωσης, είτε με αφορμή λανθασμένης επιλογής πρωτοκόλλου από την πλευρά του εκάστοτε πελάτη.

Οι γραμμές επικεφαλίδας ενός HTTP μηνύματος απόκρισης περιέχουν μία ή περισσότερες κεφαλίδες απάντησης (response headers) του HTTP. Οι κεφαλίδες αυτές προσφέρουν, όπως αναπτύχθηκε πρωτίτερα, επιπλέον πληροφορίες για την εκάστοτε απάντηση.

Το τμήμα περιεχομένου είναι προαιρετικό και περικλείει δεδομένα του υφιστάμενου σε κάθε περίπτωση αιτούμενου αρχείου, τα οποία δεν είναι πάντοτε απαραίτητα.

Ένα παράδειγμα ενός HTTP μηνύματος απόκρισης, που συνάδει με τα πιο πάνω, είναι το επόμενο:

```
HTTP/1.1 200 OK Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821 Content-Type: text/html data data ...
```

[12] [13] [14]

2.3.6 Η Αλληλεπίδραση Χρήστη-Εξυπηρετητή στο HTTP

Το πρωτόκολλο HTTP ανήκει στα stateless πρωτόκολλα, αφού δεν κρατά την κατάσταση των συνδέσεων με τους διάφορους χρήστες-πελάτες. Σε μερικές περιπτώσεις, όμως, απαιτείται η αναγνώριση των χρηστών από τον υφιστάμενο σε κάθε περίπτωση εξυπηρετητή Ιστού (πχ αν ένα σύνολο υπηρεσιών προορίζεται μόνο για κάποια συγκεκριμένη ομάδα χρηστών). Για την αναγνώριση των χρηστών το HTTP προσφέρει δύο μηχανισμούς: την πιστοποίηση (authentication) και τα cookies.

Πολλές τοποθεσίες του Παγκόσμιου Ιστού απαιτούν την παροχή ενός ονόματος και ενός κωδικού για την πρόσβαση των χρηστών σε ένα σύνολο υπηρεσιών. Η διαδικασία αυτή ονομάζεται πιστοποίηση. Το πρωτόκολλο HTTP παρέχει ειδικούς κωδικούς κατάστασης και επικεφαλίδες για το λόγο αυτό. Εφόσον, τώρα, ένας χρήστης θέλει να πάρει ένα αντικείμενο από έναν εξυπηρετητή Ιστού, για το οποίο, όμως, απαιτείται πιστοποίηση, τότε θα ακολουθηθεί η παρακάτω σειρά ενεργειών.

- I. Ένας χρήστης θα στείλει μία συνηθισμένη αίτηση μέσω του επιθυμητού σε κάθε περίπτωση προγράμματος περιήγησης σε έναν εξυπηρετητή Ιστού για την παροχή ενός αντικειμένου.
- II. Αυτός ο εξυπηρετητής θα απκριθεί με ένα μήνυμα χωρίς περιεχόμενο και με κωδικό κατάστασης “401 Authentication Required”. Σε αυτό το μήνυμα ο συγκεκριμένος εξυπηρετητής θα εσωκλείσει μία επικεφαλίδα WWW-Authentication, η οποία θα περιγράφει τις λεπτομέρειες της πιστοποίησης, που συνήθως είναι ένα όνομα χρήστη και ένας κωδικός χρήστη.
- III. Ο κατάλληλος σε κάθε περίπτωση περιηγητής θα λάβει το παραπάνω μήνυμα και θα καλέσει τον τρέχον χρήστη να εισάγει ένα όνομα και τον αντίστοιχο κωδικό. Στη συνέχεια θα ξαναστείλει την πιο προαναφερόμενη αίτηση, αλλά αυτή τη φορά θα συμπεριλάβει και μία Authentication επικεφαλίδα, η οποία θα περιέχει το όνομα και τον κωδικό του συγκεκριμένου χρήστη.

Μετά τη λήψη του πρώτου αντικειμένου ο πιο πάνω πλοηγός θα συνεχίσει να στέλνει το όνομα και τον κωδικό του παραπάνω χρήστη σε κάθε αίτηση προς τον παρών εξυπηρετητή Ιστού, μία διαδικασία η οποία τις περισσότερες φορές επαναλαμβάνεται μέχρις ότου αυτός ο χρήστης κλείσει το συγκεκριμένο φυλλομετρητή. Αν και ο παραπάνω μηχανισμός δεν καλύπτει θέματα ασφάλειας, παρέχει μία στοιχειώδη μέθοδο πιστοποίησης των χρηστών από ένα διακομιστή Ιστού.

Η άλλη μέθοδος για την αναγνώριση των χρηστών από τους εξυπηρετητές Ιστού είναι τα cookies. Πολλές τοποθεσίες του Ιστού χρησιμοποιούν cookies, ενώ κάποιες άλλες όχι. Ακολουθεί ένα παράδειγμα στο οποίο γίνεται χρήση cookies. Εφόσον ένας πελάτης στείλει για πρώτη φορά μία αίτηση σε έναν Ιστότοπο, τότε ο κατάλληλος εξυπηρετητής περιλαμβάνει στο μήνυμα απόκρισής του την επικεφαλίδα Set-cookie, η οποία περιέχει έναν αναγνωριστικό αριθμό, που δημιουργείται από αυτόν τον εξυπηρετητή. Μόλις ο παραπάνω HTTP πελάτης λάβει την απόκριση αυτή, προσθέτει μία γραμμή στο τοπικό αρχείο ενός χρήστη στο οποίο περιέχονται τα cookies. Η γραμμή αυτή συνήθως περιλαμβάνει το όνομα του πιο πάνω χρήστη και τον αναγνωριστικό αριθμό του μηνύματος απόκρισης. Αν μετά από κάποιο χρονικό διάστημα ο συγκεκριμένος πελάτης θέλει να ξαναστείλει μία αίτηση στον ίδιο εξυπηρετητή, τότε ανακτά από το προαναφερόμενο αρχείο με τα cookies τον προαναφερόμενο αναγνωριστικό αριθμό και στο μήνυμα της τρέχουσας αίτησης περικλείει μία Cookie επικεφαλίδα, με την οποία παρέχει στον παρόν εξυπηρετητή τον αναγνωριστικό αυτόν αριθμό. Με τον τρόπο αυτόν ο παρών εξυπηρετητής δε χρειάζεται να γνωρίζει το όνομα και τον κωδικό του παρόντος χρήστη, αλλά του αρκεί να γνωρίζει ότι ο χρήστης αυτός είναι ο ίδιος με αυτόν που είχε κάνει την πρώτη αίτηση. Οι διακομιστές Ιστού χρησιμοποιούν τα cookies για διάφορους λόγους, μερικοί από τους οποίους είναι οι παρακάτω:

- Ο εκάστοτε εξυπηρετητής Ιστού χρειάζεται πιστοποίηση για τον εκάστοτε χρήστη, αλλά δεν επιθυμεί κάθε φορά να ζητά από τον χρήστη αυτό την παροχή ονόματος και κωδικού (πχ όταν ένας φοιτητής πραγματοποιεί είσοδο στην ηλεκτρονική βιβλιοθήκη του πανεπιστημίου του).
- Ο εκάστοτε εξυπηρετητής Ιστού χρειάζεται να θυμάται τις προτιμήσεις του εκάστοτε χρήστη, ώστε να του αποστέλλει περιεχόμενο ανάλογο με τις προτιμήσεις αυτές (πχ στην περίπτωση που ένας χρήστης πραγματοποιεί πολύ συχνά αναζητήσεις αθλητικού περιεχομένου).
- Ο εκάστοτε εξυπηρετητής Ιστού χρειάζεται να κρατά την κατάσταση για κάποιες από τις ενέργειες που έχει κάνει ο εκάστοτε χρήστης (πχ στην περίπτωση που ένας χρήστης έχει αποθηκεύσει στο καλάθι του συγκεκριμένα προϊόντα ενός ηλεκτρονικού καταστήματος, κατά τη διαδικασία υλοποίησης μίας αγοράς στο Διαδίκτυο).

Από την παραπάνω ανάλυση, ωστόσο, προκύπτει και το βασικό μειονέκτημα των cookies: ένας χρήστης αναγνωρίζεται ως διαφορετικός, αν αλλάξει το εκάστοτε μηχάνημα με το οποίο συνδέεται με έναν Ιστότοπο.

[13] [14]

2.3.7 Το Πρωτόκολλο HTTPS

Το πρωτόκολλο HTTPS (Hypertext Transfer Protocol Secure) (απόδοση στα Ελληνικά: Ασφαλές Πρωτόκολλο Μεταφοράς Υπερκείμενου) χρησιμοποιείται στην Πληροφορική, με σκοπό να δηλώνει μία ασφαλή διαδικτυακή HTTP σύνδεση. Ένας σύνδεσμος URL, που αρχίζει με το πρόθεμα https, υποδηλώνει ότι θα χρησιμοποιηθεί κανονικά το πρωτόκολλο HTTP, αλλά η σύνδεση θα γίνει σε διαφορετική πόρτα (443, αντί 80) και τα δεδομένα θα ανταλλάσσονται κρυπτογραφημένα. Το σύστημα αυτό σχεδιάστηκε αρχικά από την εταιρία Netscape Communications Corporation, για να βρίσκει εφαρμογή σε Ιστοχώρους όπου απαιτείται αυθεντικοποίηση χρηστών και κρυπτογραφημένη επικοινωνία. Σήμερα χρησιμοποιείται ευρέως στο Διαδίκτυο, όπου χρειάζεται αυξημένη ασφάλεια, διότι διακινούνται ευαίσθητες πληροφορίες (πχ κωδικοί κάθε είδους, αριθμοί πιστωτικών καρτών).

Το HTTPS δεν είναι ξεχωριστό πρωτόκολλο, όπως μερικοί νομίζουν, αλλά αναφέρεται στον συνδυασμό του απλού HTTP πρωτοκόλλου και των δυνατοτήτων κρυπτογράφησης που παρέχει το πρωτόκολλο SSL (Secure Sockets Layer). Η κρυπτογράφηση που διατίθεται διασφαλίζει πως τα κρυπτογραφημένα δεδομένα δε θα μπορούν να υποκλαπούν από άλλους κακόβουλους χρήστες ή από man-in-the-middle επιθέσεις.

Για να χρησιμοποιηθεί το πρωτόκολλο HTTPS σε έναν εξυπηρετητή Ιστού, θα πρέπει ο διαχειριστής του να εκδώσει ένα πιστοποιητικό δημοσίου κλειδιού. Στη συνέχεια το πιστοποιητικό αυτό θα πρέπει να υπογραφεί από μία αρχή πιστοποίησης (certificate authority), η οποία θα πιστοποιεί πως ο εκδότης του πιστοποιητικού είναι νομότυπος και ότι το πιστοποιητικό είναι έγκυρο. Με τον τρόπο αυτόν οι εκάστοτε χρήστες θα είναι ικανοί να δουν την υπογραφή της προαναφερόμενης αρχής πιστοποίησης και να βεβαιωθούν ότι το πιστοποιητικό είναι έγκυρο και ότι κανένας κακόβουλος χρήστης δεν το έχει πλαστογραφήσει.

[15]

2.3.8 Σύγκριση του HTTP

Το HTTP είναι, ίσως, το βασικότερο πρωτόκολλο επικοινωνίας στο Διαδίκτυο. Στη λογική του στηρίζονται πολλά νεότερα πρωτόκολλα επικοινωνίας όπως είναι, για παράδειγμα το SOAP (Simple Object Access Protocol) και το SIP (Session Initiation Protocol). Υπάρχουν, ωστόσο, και άλλα θεμελιώδη πρωτόκολλα επικοινωνίας στον Παγκόσμιο Ιστό, που η γέννηση τους συνάδει με αυτή του ίδιου του Ιστού ή είναι ακόμα αρχαιότερη. Ένα τέτοιο πρωτόκολλο ιδιάζουσας σημασίας, παλαιότερο κατά μία δεκαετία περίπου από το HTTP, σχετικά με τη μεταφορά κάθε μορφής πληροφορίας πάνω από το Διαδίκτυο, είναι το πρωτόκολλο FTP (File Transfer Protocol) (απόδοση στα Ελληνικά: Πρωτόκολλο Μεταφοράς Αρχείου).

Παρόλο που τα πρωτόκολλα HTTP και το FTP έχουν αρκετές ομοιότητες στην αρχιτεκτονική και τον τρόπο λειτουργίας τους, εμφανίζουν, ωστόσο, και κάποιες διαφορές είτε λεπτομερείς είτε πιο κρίσιμες. Η σύγκριση των δύο αυτών πρωτοκόλλων, λοιπόν, επικεντρώνεται στα ακόλουθα σημεία:

- Το πρωτόκολλο FTP μεταφέρει δεδομένα σε διάφορες μορφές, όπως είναι για παράδειγμα η δυαδική (binary) ή του συστήματος χαρακτήρων ASCII (American Standard Code for Information Interchange), ενώ το πρωτόκολλο HTTP μεταφέρει δεδομένα αποκλειστικά σε δυαδική μορφή.
- Η μεταφορά ενός αρχείου με χρήση του πρωτοκόλλου FTP δεν εμπλέκει μεταδεδομένα (metadatas). Αντίθετα, το πρωτόκολλο HTTP με τη μεταφορά ενός αρχείου κουβαλάει και μεταδεδομένα, τα οποία σχετίζονται κυρίως με τις επικεφαλίδες των HTTP μηνυμάτων.
- Το πρωτόκολλο HTTP υποστηρίζει την τεχνική pipeling, σύμφωνα με την οποία ένας πελάτης στην ίδια TCP σύνδεση δύναται να αιτηθεί τη μεταφορά ενός αρχείου, πριν να ολοκληρωθεί η μεταφορά ενός άλλου αρχείου, που ήδη πραγματοποιείται. Από την πλευρά του, το πρωτόκολλο FTP δεν εκμεταλλεύεται τη συγκεκριμένη τεχνική, ωστόσο στο FTP πολλαπλά αρχεία μπορούν να μεταφερθούν ταυτόχρονα με χρήση ξεχωριστών TCP συνδέσεων για τη μεταφορά καθενός από τα αρχεία αυτά.
- Για τη μεταφορά ενός μόνο αρχείου στο πρωτόκολλο HTTP απαιτείται αποκλειστικά ένα μήνυμα αίτησης και ένα μήνυμα απάντησης. Στον αντίποδα, για τη μεταφορά ενός μόνο αρχείου στο πρωτόκολλο FTP είναι αναγκαία μία σειρά εντολών, γεγονός που καθυστερεί αρκετά τη μεταφορά αυτή.
- Το πρωτόκολλο FTP χρησιμοποιεί δύο διαφορετικές συνδέσεις για την επικοινωνία στο Διαδίκτυο: μία για την αποστολή των πληροφοριών ελέγχου και μία για τα δεδομένα, ενώ το πρωτόκολλο HTTP υλοποιεί και τις δύο συγκεκριμένες ενέργειες σε μία μόνο σύνδεση.
- Στο πρωτόκολλο HTTP ένας πελάτης δύναται να συντηρεί μία και μόνο σύνδεση με έναν εξυπηρετητή Ιστού, προκειμένου να λάβει ολόκληρο το περιεχόμενο ενός Ιστοτόπου μέσω μίας ακολουθίας διαδοχικών μεταφορών. Στην αντίπερα όχθη, στο πρωτόκολλο FTP ένας πελάτης πρέπει να εγκαθιδρύει μία καινούργια σύνδεση κάθε φορά που αιτείται διαφορετικό αντικείμενο από έναν Ιστοχώρο, φαινόμενο που οδηγεί συχνά σε συμφόρηση του υπάρχοντος σε κάθε περίπτωση δικτύου.
- Στις περιπτώσεις που η πιστοποίηση χρηστών είναι απαραίτητη και τα δύο πρωτόκολλα στέλνουν τα ονόματα και τους κωδικούς αυτών με τη μορφή απλού κειμένου. Πάραυτα το HTTP διαθέτει περισσότερες μεθόδους για να κρυπτογραφήσει τα προηγούμενα στοιχεία σε σχέση με τις μεθόδους που παρέχει το FTP.
- Τόσο το πρωτόκολλο FTP όσο και το πρωτόκολλο HTTP υποστηρίζουν επαναλαμβανόμενες μεταβιβάσεις είτε προς ένα διακομιστή Ιστού είτε προς έναν πελάτη, αλλά το HTTP έχει προχωρημένες μεθόδους τεμαχισμού του περιεχομένου μίας Ιστοσελίδας σε μικρά κομμάτια. Έτσι, μπορεί να ανταποκρίνεται ευκολότερα σε αιτήματα εύρους bytes αναφορικά με το FTP.

[12]

2.4 Η Αρχιτεκτονική Ανάπτυξης Λογισμικού, Πελάτη-Εξυπηρετητή

2.4.1 Εισαγωγή στην Αρχιτεκτονική Πελάτη-Εξυπηρετητή

Η αρχιτεκτονική ανάπτυξης λογισμικού πελάτη-εξυπηρετητή (client-server) αναφέρεται σε μία βασική αλλαγή στο στυλ των υπολογιστών, την αλλαγή από τα συστήματα που βασίζονται στα μηχανήματα στα συστήματα που βασίζονται στους χρήστες. Ειδικότερα, ένα σύστημα πελάτη-εξυπηρετητή είναι ένα σύστημα στο οποίο ένα δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε ο πελάτης να μπορεί να ζητά υπηρεσίες από το διακομιστή, ο οποίος προσφέρει πληροφορίες ή επιπρόσθετη υπολογιστική ισχύ. Με άλλα λόγια, σύμφωνα με μοντέλο πελάτη-εξυπηρετητή ο πελάτης θέτει μία αίτηση και ο κατάλληλος σε κάθε περίπτωση διακομιστής επιστρέφει μία ανταπόκριση ή κάνει μία σειρά από ενέργειες. Ο διακομιστής μπορεί να ενεργοποιείται άμεσα για την αίτηση αυτή ή να την προσθέτει σε μία ουρά. Η άμεση ενεργοποίηση για την παραπάνω αίτηση μπορεί, για παράδειγμα, να σημαίνει ότι ο εξυπηρετητής υπολογίζει έναν αριθμό και τον επιστρέφει αμέσως στον πελάτη. Η τοποθέτηση της συγκεκριμένης αίτησης σε μία ουρά συνεπάγεται πως αυτή τίθεται σε αναμονή, για να εξυπηρετηθεί μελλοντικά. Ένα αντιπροσωπευτικό παράδειγμα αυτής της περίπτωσης είναι η εκτύπωση σε έναν εκτυπωτή δικτύου. Ένας διακομιστής τοποθετεί μία αίτηση εκτύπωσης ενός πελάτη σε μία ουρά μαζί με αιτήσεις εκτυπώσεων και από άλλους πελάτες. Μετά επεξεργάζεται την αίτηση αυτή με βάση μία σειρά προτεραιότητας, η οποία καθορίζεται από τη σειρά με την οποία ο προαναφερόμενος διακομιστής παρέλαβε την απαίτηση.

Συνοπτικά, η διαδικασία πελάτη-εξυπηρετητή απλοποιείται στα εξής βήματα:

- Ο χρήστης δημιουργεί μία αίτηση ή ένα ερώτημα.
- Ο πελάτης μορφοποιεί το ερώτημα και το στέλνει στον εξυπηρετητή.
- Ο διακομιστής ελέγχει την πιθανότητα πρόσβασης του χρήστη.
- Ο εξυπηρετητής επεξεργάζεται το ερώτημα και επιστρέφει τα αποτελέσματα.
- Ο πελάτης λαμβάνει την ανταπόκριση και τη μορφοποιεί για τον χρήστη.
- Ο χρήστης βλέπει και χειρίζεται την πληροφορία.

Η αρχιτεκτονική πελάτη-εξυπηρετητή είναι πολύ σημαντική, διότι επιτυγχάνει τα εξής:

- Αποτελεσματική χρήση της υπολογιστικής ισχύος.
- Μείωση του κόστους συντήρησης, δημιουργώντας συστήματα πελάτη-εξυπηρετητή που απαιτούν λιγότερη συντήρηση και κοστίζουν λιγότερο στην αναβάθμιση.
- Αύξηση της παραγωγικότητας, προσφέροντας στους χρήστες ξεκάθαρη πρόσβαση σε αναγκαίες πληροφορίες μέσω σταθερών και εύκολων στην χρήση διασυνδέσεων.
- Αύξηση της ευελιξίας και της προοπτικής δημιουργίας συστημάτων που υποστηρίζουν πολλά περιβάλλοντα.

Το βασικό μοντέλο πελάτη-εξυπηρετητή έχει την εξής κεντρική ιδέα: η πλευρά του πελάτη πρώτα στέλνει ένα μήνυμα για να καλέσει σε ετοιμότητα τον διακομιστή. Από τη στιγμή που ο πελάτης και ο εξυπηρετητής έχουν επικοινωνία μεταξύ τους, ο πελάτης μπορεί να υποβάλλει την αίτησή του.

Από τη μία πλευρά, ο πελάτης είναι ο αιτών των υπηρεσιών. Ο πελάτης δεν μπορεί παρά να είναι ένας υπολογιστής. Οι υπηρεσίες που ζητούνται από τον πελάτη είναι εφικτό να υπάρχουν στους ίδιους σταθμούς εργασίας ή σε απομακρυσμένους σταθμούς εργασίας, που συνδέονται μεταξύ τους μέσω ενός δικτύου. Ο πελάτης ξεκινάει πάντα την επικοινωνία. Μία μηχανή, λοιπόν, που παίζει το ρόλο του πελάτη πρέπει να είναι θέση να υλοποιεί τα επόμενα:

- Να τρέχει το λογισμικό Γραφικών Διεπαφών Χρηστών (Graphical Users Interfaces-GUIs).
- Να δημιουργεί τις αιτήσεις για πληροφορίες και να τις στέλνει στον εξυπηρετητή.
- Να αποθηκεύει τις επιστρεφόμενες πληροφορίες.

Από την άλλη πλευρά ο εξυπηρετητής απαντάει στις αιτήσεις, οι οποίες γίνονται από τους πελάτες. Οι εξυπηρετητές δεν ξεκινάνε τις επικοινωνίες, αλλά περιμένουν τις αιτήσεις των πελατών. Έτσι, στο παραπάνω παράδειγμα του διακομιστή εκτυπώσεων ενός δικτύου, ο πελάτης ζητάει από τον εξυπηρετητή να εκτυπώσει ένα κείμενο σε ένα συγκεκριμένο εκτυπωτή και ο διακομιστής προσθέτει την εκτύπωση σε μία ουρά και ενημερώνει τον πελάτη, μόλις το κείμενο εκτυπωθεί επιτυχημένα. Η διαδικασία του πελάτη δύναται να ανήκει στον ίδιο σταθμό εργασίας με την διαδικασία του διακομιστή. Στο συγκεκριμένο παράδειγμα μία εντολή εκτύπωσης εκδίδεται στον εξυπηρετητή του σταθμού εργασίας του δικτύου, κάνοντας χρήση την διαδικασία του διακομιστή εκτυπώσεων σε αυτόν τον σταθμό εργασίας. Μία μηχανή, λοιπόν, που διαδραματίζει το ρόλο του εξυπηρετητή πρέπει να είναι σε θέση να εκτελεί τα παρακάτω:

- Να αποθηκεύει, να ανακτά, να προστατεύει πληροφορίες και να επιθεωρεί τις αιτήσεις των πελατών.

- Να δημιουργεί εφαρμογές διαχείρισης πληροφοριών, όπως η δημιουργία αντιγράφων, ασφάλειας και άλλες.
- Να διαχειρίζεται πληροφορίες.

Στην εξίσωση πελάτη-εξυπηρετητή είναι αθέμιτο να παραληφθεί η παράμετρος των δικτύων. Τα δίκτυα ελέγχονται από κατάλληλο λογισμικό διαχείρισης, με στόχο να διασφαλίζουν τις υπηρεσίες επικοινωνίας του διακομιστή και να προστατεύουν τα προγράμματα τόσο του πελάτη όσο και του εξυπηρετητή από το να έχουν άμεση σύνδεση μεταξύ τους. Συνεπώς, ένα λογισμικό διαχείρισης εστιάζει στη παροχή αξιόπιστων υπηρεσιών, στην ελαχιστοποίηση των προβλημάτων σε ένα δίκτυο και στην ελαχιστοποίηση των χρόνων πτώσης του δικτύου.

[16] [17]

2.4.2 Τα Συστατικά της Αρχιτεκτονικής Πελάτη-Εξυπηρετητή

Τα κύρια συστατικά μίας εφαρμογής πελάτη-εξυπηρετητή είναι, όπως μαρτυρά, εξάλλου, και το όνομα της, ο πελάτης και ο εξυπηρετητής.

Το τμήμα του πελάτη μίας εφαρμογής πελάτη-εξυπηρετητή απαρτίζεται από τα παρακάτω μέρη:

- Το υλικό (hardware).
- Το λειτουργικό σύστημα (operating system).
- Το δίκτυο (network).
- Τη Γραφική Διεπαφή του Χρήστη (Graphical User Interface-GUI).
- Το λογισμικό (software).

Στην αρχιτεκτονική πελάτη-εξυπηρετητή ο πελάτης παίζει τους ακόλουθους βασικούς ρόλους:

- Προσφέρει μία εύκολη στη χρήση διασύνδεση χρηστών.
- Στέλνει αιτήσεις.
- Δέχεται ανταποκρίσεις.
- Επιτρέπει στον εκάστοτε χρήστη να βλέπει και να χειρίζεται τις πληροφορίες.

Παρόλο που ο πελάτης κατέχει ένα μεγάλο μέρος της προσοχής, αφού με αυτόν αλληλεπιδρούν οι χρήστες, ο διακομιστής είναι η καρδιά του συστήματος πελάτη-εξυπηρετητή. Οι εξυπηρετητές είναι τα σημεία όπου αποθηκεύονται οι πληροφορίες και εκτελούνται οι εργασίες. Σήμερα, οι διακομιστές μπορούν να είναι οποιαδήποτε μορφή υπολογιστή. Ωστόσο, η αύξηση της ισχύς και η μείωση του κόστους των προσωπικών υπολογιστών τους καθιστά γενικά την πιο συμφέρουσα οικονομικά επιλογή. Ακόμα και αν ο εκάστοτε εξυπηρετητής είναι ένας σταθερός προσωπικός υπολογιστής, αυτό που κάνει τη διαφορά από ένα σταθερό προσωπικό σύστημα, είναι πως είναι εξειδικευμένος και έχει συγκεκριμένες πρωτοβουλίες.

Οι εξυπηρετητές μπορούν να διαιρεθούν στους εξής τύπους:

- Εξυπηρετητές εφαρμογών (application servers): Τρέχουν λογισμικό εφαρμογών, που είναι πολύ σημαντικό, όταν διανέμονται λογικές εφαρμογών μεταξύ ενός πελάτη και ενός εξυπηρετητή. Η τοποθέτηση εφαρμογών στον εξυπηρετητή σημαίνει πως αυτές οι εφαρμογές είναι διαθέσιμες σε πολλούς πελάτες. Πολλοί πελάτες έχουν τη δυνατότητα να χρησιμοποιούν Κλήσεις Απομακρυσμένης Διαδικασίας (Remote Procedure Calls-RPCs), ώστε να θέτουν σε λειτουργία μία επεξεργασία στο διακομιστή. Επίσης, πολλοί εξυπηρετητές εφαρμογών μπορούν να συνεργάζονται, προκειμένου να απαντούν στις αιτήσεις των πελατών. Κάθε εξυπηρετητής μπορεί να τρέχει ένα διαφορετικό λειτουργικό σύστημα σε μία διαφορετική πλατφόρμα υλικού, αλλά αυτές οι λεπτομέρειες είναι ξεκάθαρες στον πελάτη, ο οποίος μπορεί να κάνει αιτήσεις, δίχως να υπολογίζει τον τύπο της μηχανής που θα ανταποκριθεί.
- Εξυπηρετητές πληροφοριών (data servers): Χρησιμοποιούνται αποκλειστικά για αποθήκευση και διαχείριση πληροφοριών. Αυτοί οι εξυπηρετητές ερευνούν και ελέγχουν την αξιοπιστία των πληροφοριών, αλλά γενικά δε μεταβιβάζουν μεγάλη ποσότητα βασικών πληροφοριών σε ένα δίκτυο.
- Εξυπηρετητές υπολογισμών (compute servers): Παίρνουν τις αιτήσεις των πελατών για πληροφορίες από έναν εξυπηρετητή πληροφοριών και μετά προωθούν τα αποτελέσματα των αιτήσεων πίσω στους πελάτες.
- Εξυπηρετητές βάσεων δεδομένων (database servers): Είναι τυπικά συστήματα πελάτη-εξυπηρετητή και έχουν να κάνουν την ίδια εργασία με αυτήν που κάνουν οι εξυπηρετητές

πληροφοριών και υπολογισμών μαζί. Οι εξυπηρετητές βάσεων δεδομένων τρέχουν DBMS λογισμικό και πολύ πιθανό και κάποια λογική εφαρμογή πελάτη-εξυπηρετητή, που σημαίνει ότι αυτός ο τύπος διακομιστών χρειάζεται περισσότερη ισχύ. Τα ΣΔΒΔ (Συστήματα Διαχείρισης Βάσεων Δεδομένων) (DBMS-Database Management Systems) προσφέρουν εξειδικευμένες υπηρεσίες, όπως είναι, για παράδειγμα, η δυνατότητα ανάκτησης και διαχείρισης πληροφοριών. Οι διακομιστές που συνδυάζουν τις λειτουργίες των εξυπηρετητών βάσεων δεδομένων και των εξυπηρετητών εφαρμογών είναι, επίσης, γνωστοί ως εξυπηρετητές συναλλαγών (transaction servers).

- Εξυπηρετητές πόρων ή επικοινωνιών (resources or communications servers): Οι εξυπηρετητές πόρων επιτρέπουν σε πολλούς πελάτες την προσπέλαση συγκεκριμένων πόρων, που είναι ουσιαστικά πολύ ακριβοί, για να βρίσκονται σε έναν πελάτη. Για παράδειγμα, οι εξυπηρετητές εκτύπωσης (print servers) συνδέουν πολλούς πελάτες με πολλούς εκτυπωτές. Επίσης, οι εξυπηρετητές πόρων έχουν την ικανότητα να συνδέουν πελάτες με άλλες συσκευές, όπως είναι, για παράδειγμα, τα πολυμέσα. Από τη στιγμή που οι εξυπηρετητές πόρων είναι συνδεδεμένοι σε μία συγκεκριμένη συσκευή, δεν απαιτείται συνήθως τόσο πολύ ισχύ όση αυτή των διακομιστών που προσφέρουν περισσότερο περίπλοκες υπηρεσίες. Οι εξυπηρετητές επικοινωνιών συνδέουν απομακρυσμένα συστήματα.

Ο τύπος του διακομιστή που χρησιμοποιείται σε κάθε περίπτωση εξαρτάται από την απαιτούμενη εργασία. Επιπλέον, οι παραπάνω έξι ρόλοι μπορούν να συνδυαστούν σε ένα σύστημα ή να διαιρεθούν σε περισσότερα. Για παράδειγμα, μία μηχανή μπορεί να εξυπηρετεί είτε ως εξυπηρετητής εφαρμογών είτε ως εξυπηρετητής βάσεων δεδομένων.

Το τμήμα του εξυπηρετητή μίας εφαρμογής πελάτη-εξυπηρετητή απαρτίζεται από τα παρακάτω στοιχεία:

- Το υλικό (hardware).
- Το λειτουργικό σύστημα (operating system).
- Τη βάση δεδομένων (database).
- Το λογισμικό (software)

Στην αρχιτεκτονική πελάτη-εξυπηρετητή ο εξυπηρετητής επιτελεί τέσσερις βαρυσήμαντες λειτουργίες, οι οποίες είναι οι εξής:

- Ακούει την αίτηση του πελάτη .
- Ελέγχει την δυναμικότητα πρόσβασης του χρήστη.
- Επεξεργάζεται την αίτηση του πελάτη.
- Επιστρέφει τα αποτελέσματα στον χρήστη.

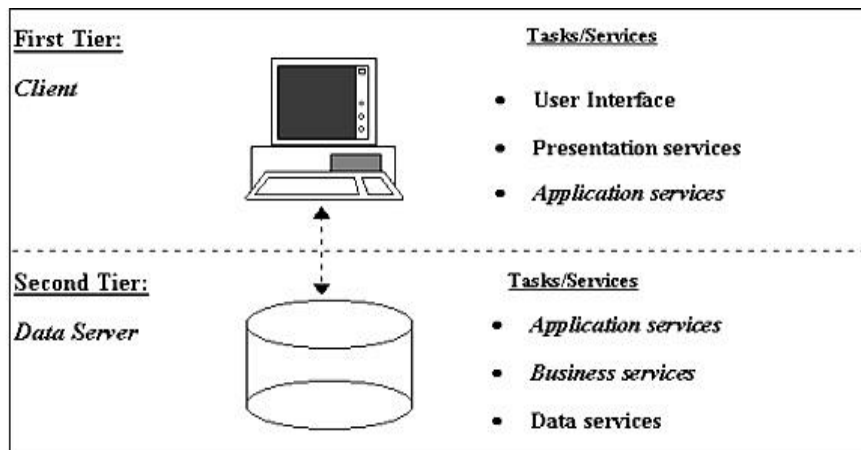
Ο διακομιστής δεν εγκαινιάζει καμιά ενέργεια. Αντίθετα, ο διακομιστής περιμένει παθητικά να φτάσουν οι αιτήσεις του πελάτη μέσω του δικτύου. Ο εξυπηρετητής πρέπει πάντα να απαντάει στους πελάτες, ακόμα και όταν πολλοί πελάτες κάνουν ταυτόχρονες αιτήσεις. Από τη στιγμή που ο διακομιστής δέχεται από τον πελάτη την απαίτηση, ο εξυπηρετητής πρέπει να βεβαιωθεί ότι ο πελάτης είναι εξουσιοδοτημένος να λάβει την πληροφορία ή την απάντηση. Αν ο πελάτης δεν είναι εξουσιοδοτημένος, τότε ο εξυπηρετητής απορρίπτει την αίτηση και στέλνει κατάλληλο μήνυμα στον πελάτη. Αν ο πελάτης είναι εξουσιοδοτημένος, ο εξυπηρετητής συνεχίζει και επεξεργάζεται την αίτηση. Η επεξεργασία της αίτησης περιλαμβάνει την παραλαβή της αίτησης του πελάτη, την μετατροπή της σε μία μορφή που μπορεί ο διακομιστής να χρησιμοποιεί και τον χειρισμό αυτής. Όταν η επεξεργασία ολοκληρώνεται, τότε ο εξυπηρετητής στέλνει τα αποτελέσματα πίσω στον πελάτη. Μετά, ο πελάτης δύναται να μεταφράζει και να χρησιμοποιεί τις πληροφορίες. Δεν υπάρχει προκαθορισμένος διαχωρισμός στις ευθύνες για τις εφαρμογές πελάτη-εξυπηρετητή. Ανάλογα με τις ανάγκες του εκάστοτε χρήστη γίνεται και ο διαχωρισμός της εκάστοτε εφαρμογής. Το ισχυρό μοντέλο πελάτη δίνει πιο πολλές λειτουργίες στον πελάτη, ενώ το ισχυρό μοντέλο εξυπηρετητή δίνει περισσότερες λειτουργίες στον εξυπηρετητή. Οι διακομιστές εφαρμογών και συναλλαγών τείνουν να είναι ισχυροί εξυπηρετητές, ενώ οι διακομιστές βάσεων δεδομένων και αρχείων τείνουν να έχουν ισχυρούς πελάτες. Ανεξάρτητα, πάντως, του πως διαχωρίζεται μία εφαρμογή, η βασική ευθύνη του διακομιστή παραμένει η ίδια: να εξυπηρετεί τους πελάτες που κάνουν αιτήσεις.

[16] [17] [18]

2.4.3 Οι Κατηγορίες της Αρχιτεκτονικής Πελάτη-Εξυπηρετητή

Οι εφαρμογές πελάτη-εξυπηρετητή διακρίνονται ανάλογα με τον τρόπο με τον οποίο η εκάστοτε εφαρμογή κατανέμεται μεταξύ του πελάτη και του εξυπηρετητή σε δύο κατηγορίες, οι οποίες περιγράφονται αμέσως παρακάτω:

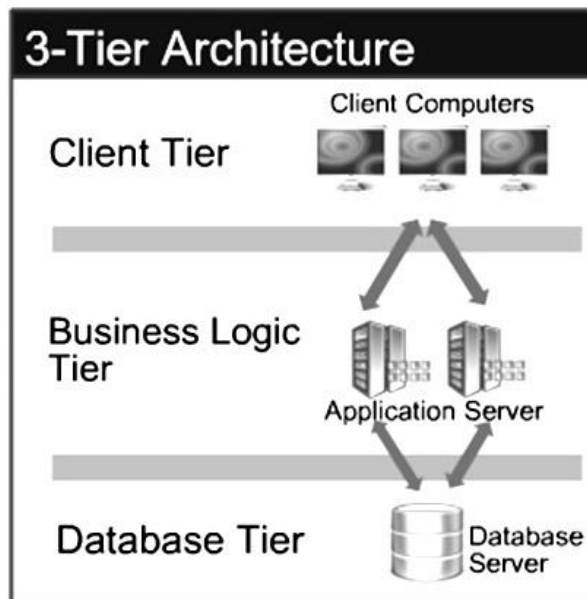
Πρώτον, η αρχιτεκτονική δύο στρωμάτων (two-tier): Οι εφαρμογές πελάτη-εξυπηρετητή πρώτης γενιάς εκτελούνται γενικά με δύο λογικά επίπεδα. Η συγκεκριμένη αρχιτεκτονική διαιρεί μία εφαρμογή σε δύο συγκεκριμένα τμήματα, όπου ένα τμήμα τρέχει στον πελάτη και ένα ξεχωριστό τμήμα τρέχει στο διακομιστή, όπως φαίνεται και στην παρακάτω Εικόνα:



Εικόνα 4: Η αρχιτεκτονική πελάτη-εξυπηρετητή δύο στρωμάτων.

Αξίζει να σημειωθεί ότι στην τρέχουσα αρχιτεκτονική ο κώδικας του πελάτη και του εξυπηρετητή δεν ενημερώνει ούτε καν γνωρίζει, αν αυτοί τρέχουν κάθε φορά στον ίδιο υπολογιστή ή όχι. Επιπλέον, η εφαρμογή διαιρείται κατά μήκος του πελάτη και του εξυπηρετητή. Η ποσότητα της εκάστοτε λογικής εφαρμογής που λειτουργεί στον πελάτη ή στον διακομιστή καθορίζει, αν αυτό το επίπεδο-τμήμα είναι αδύνατο ή ισχυρό. Το αδύνατο τμήμα παρουσιάζει μικρή ανάπτυξη της εφαρμογής, αντίθετα το ισχυρό τμήμα εμφανίζει μεγάλη ανάπτυξη της εφαρμογής. Υπάρχουν ποικίλες διαβαθμίσεις αδυνάτου και ισχυρού. Μάλιστα, οι αδύνατοι πελάτες είναι ελκυστικοί, όταν ο υπολογιστής-πελάτης έχει περιορισμένη επίδοση. Η αρχιτεκτονική δύο στρωμάτων εμφανίζεται να είναι πιο δύσκολο να αναπτυχθεί και να διατηρηθεί σε σχέση με τι κανονικά προσδοκάται. Επιπλέον, οι εφαρμογές δύο επιπέδων δεν εμφανίζουν καλή κλιμάκωση και τα εργαλεία της αρχιτεκτονικής πελάτη-εξυπηρετητή δύο στρωμάτων πήραν χρόνο για να αναπτυχθούν. Η Visual Basic, η Delphi και το PowerBuilder είναι συγκεκριμένα παραδείγματα αυτού του τύπου των εργαλείων.

Δεύτερον, η αρχιτεκτονική τριών στρωμάτων (three-tier): Ο πιο πρόσφατος τύπος αρχιτεκτονικής πελάτη-εξυπηρετητή που αναπτύσσεται είναι ο τριών στρωμάτων (three-tier), ο οποίος χωρίζει μία εφαρμογή σε τρία επιμέρους στρώματα, όπως απεικονίζεται και στην κατώτερη Εικόνα:



Εικόνα 5: Η αρχιτεκτονική πελάτη-εξυπηρετητή τριών στρωμάτων.

Μερικοί άνθρωποι, επίσης, χρησιμοποιούν πολλαπλά στρώματα (multi-tier), για να περιλάβουν οποιαδήποτε προσέγγιση η οποία χρησιμοποιεί πάνω από δύο λογικά τμήματα. Μία πρωταρχική διαφορά μεταξύ των εφαρμογών δύο και τριών επιπέδων είναι η επιπλέον επίστρωση λογισμικού στο διακομιστή. Εκεί που οι εφαρμογές δύο επιπέδων τείνουν να τοποθετούν την λογική στον πελάτη και να περνούν εγγραφές ή δεδομένα σε μία βάση δεδομένων, όπου αποθηκευμένες διαδικασίες εκτελούν την λογική της εφαρμογής, οι εφαρμογές τριών επιπέδων τείνουν να περνούν μήνυμα μεταξύ των τμημάτων κώδικα του πελάτη και του εξυπηρετητή της εφαρμογής. Το τμήμα του διακομιστή εφαρμόζει τη λογική της εφαρμογής, η οποία συνήθως, καλείται Business Logic, και κατόπιν τη στέλνει σε μία βάση δεδομένων. Οι εφαρμογές πελάτη-εξυπηρετητή τριών στρωμάτων προσθέτουν κάποια πολυπλοκότητα, διότι πρόκειται για ένα επιπρόσθετο κομμάτι κώδικα που αναπτύσσεται. Μολονότι συμβαίνει αυτό, υπάρχουν ορισμένα πλεονεκτήματα σε μία προσέγγιση πελάτη-εξυπηρετητή τριών στρωμάτων σε σχέση με την αντίστοιχη δύο στρωμάτων, τα οποία περιγράφονται παρακάτω:

- Η κλιμάκωση βελτιώνεται, εφόσον ο κώδικας του διακομιστή και η εκάστοτε βάση δεδομένων είναι χωρισμένα. Πολλαπλές εφαρμογές εξυπηρετητών μπορούν να επικοινωνούν με μία κεντρική βάση δεδομένων ή μία εφαρμογή διακομιστή μπορεί να εξυπηρετεί τους πελάτες, ενώ προσπελάζονται οι πολλαπλές βάσεις δεδομένων, όσο το υφιστάμενο κάθε φορά σύστημα αυξομειώνεται.
- Τα δίκτυα διανομής εμφανίζουν χαμηλότερα προβλήματα, φαινόμενο που απορρέει από το πέρασμα μικρών μηνυμάτων στην εκάστοτε εφαρμογή παρά από ολόκληρες εγγραφές πληροφοριών.
- Η ευλυγισία κερδίζεται, γιατί ο πελάτης, ο εξυπηρετητής και τα συστήματα βάσεων δεδομένων μπορούν να αντικαθίστανται, χωρίς να επηρεάζουν τα άλλα κομμάτια, από τη στιγμή που η διασύνδεση δεν αλλάζει. Πολλές από τις σημερινές εμπορικές εφαρμογές, που κάνουν χρήση βάσεων δεδομένων, είναι βασισμένες στην αρχιτεκτονική πελάτη-εξυπηρετητή τριών στρωμάτων, με στόχο να κερδίζουν το ενδεχόμενο αυξομείωσης και ευλυγισίας. Ακόμα, οι εφαρμογές Διαδικτύου είναι γενικά τριών επιπέδων.

[17] [19]

2.4.4 Σύγκριση της Αρχιτεκτονικής Πελάτη-Εξυπηρετητή

Η αρχιτεκτονική πελάτη-εξυπηρετητή πλεονεκτεί έναντι των άλλων κλασσικών μοντέλων αρχιτεκτονικών στην εκμετάλλευση των δυνατοτήτων που παρέχουν οι πελάτες, οι διακομιστές και το εκάστοτε δίκτυο, κατά τη διαδικασία ανάπτυξης μίας εφαρμογής. Για να επωφεληθεί κάποιος από αυτές τις δυνατότητες, πρώτα πρέπει να γνωρίζει ότι η πιο συνηθισμένη λειτουργία μίας τέτοιας εφαρμογής

είναι η παροχή δυνατότητας πρόσβασης των χρηστών στις πληροφορίες, αποτελεσματικά και εύκολα. Είναι αναγκαίο να ενοποιούνται ομαλά τα GUIs, οι καταναμημένες εφαρμογές, οι συγγενικές βάσεις δεδομένων και τα δίκτυα. Οι πληροφορίες δεν αποθηκεύονται ή ελέγχονται από κεντρικούς μεγάλους υπολογιστές (mainframes). Αντίθετα, είναι εύκολα προσπελάσιμες στους εξυπηρετητές του δικτύου.

Από τη στιγμή που το μοντέλο πελάτη-εξυπηρετητή είναι ανώτερο από τα κλασσικά μοντέλα, αυτό ενισχύει τις αδυναμίες των παραδοσιακών μεθοδολογιών ανάπτυξης συστημάτων. Οι πληροφορίες που ελέγχονται από αυτό το σύστημα είναι περισσότερες από τις προηγούμενες αρχιτεκτονικές. Έπειτα, η ασφάλεια που παρέχεται σε κάθε στρώμα, κάθε χρονική στιγμή ρυθμίζεται σε διάφορα επίπεδα συμπεριλαμβανομένων των σταθμών εργασιών, των πληροφοριών και του χρόνου. Και τέλος, κάτι ιδιαίτερα σημαντικό: οι πελάτες και οι διακομιστές προσδιορίζονται από το εκάστοτε λογισμικό και όχι από το εκάστοτε υλικό.

[20]

2.5 Η Αρχιτεκτονική Ανάπτυξης Λογισμικού, Πράκτορες Λογισμικού

2.5.1 Εισαγωγή στους Πράκτορες Λογισμικού

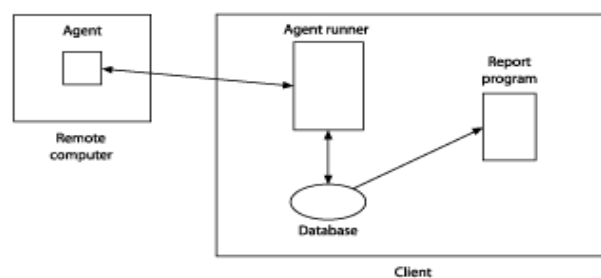
Με τον γενικό όρο πράκτορας λογισμικού (software agent) γίνεται αναφορά σε μία αυτόνομη οντότητα λογισμικού, η οποία αυτοματοποιεί κάποιες ενέργειες που ένας άνθρωπος ή μία διεργασία λογισμικού μπορεί να μεταβιβάσει σε αυτή. Κάθε πράκτορας λογισμικού έχει τη δική του πληροφορία κατάστασης, τη δική του συμπεριφορά και το δικό του νήμα ελέγχου και μπορεί να αλληλεπιδρά ελεύθερα με άλλες οντότητες με σκοπό την επίλυση κάποιου συγκεκριμένου προβλήματος.

[21]

2.5.2 Οι Κατηγορίες των Πρακτόρων Λογισμικού

Οι πράκτορες λογισμικού διαχωρίζονται σε δύο βασικές κατηγορίες, που είναι οι επόμενες:

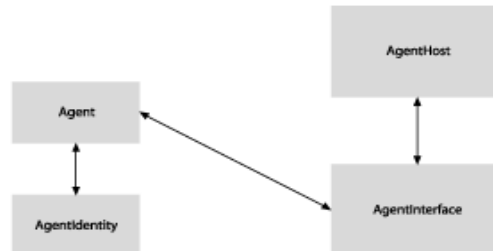
Πρώτον, οι μη-κινητοί πράκτορες (non-mobile agents) λογισμικού, δηλαδή προγράμματα που εκτελούνται στον εκάστοτε πελάτη και διατρέχουν έναν απομακρυσμένο Ιστοχώρο, με τα ενδιάμεσα αποτελέσματα από την εκτέλεση του πράκτορα να επιστρέφουν στον πελάτη. Η αρχιτεκτονική ενός μη-κινητού πράκτορα λογισμικού φαίνεται στην παρακάτω Εικόνα:



Εικόνα 6.: Η αρχιτεκτονική μη-κινητού πράκτορα λογισμικού

Το πρώτο από τα παραπάνω διαγράμματα παρουσιάζει ένα σύστημα, όπου αναπτύσσεται μόνο ένας μη-συνεργατικός πράκτορας λογισμικού. Η αρχιτεκτονική αποτελείται από ένα πλαίσιο που χειρίζεται τη δημιουργία και την καταστροφή ενός πράκτορα, την αποστολή, τη δημιουργία και τη συντήρηση μίας βάσης δεδομένων για όλα τα δεδομένα, που είχαν ανακτηθεί από αυτόν τον πράκτορα. Ένα πρόγραμμα αναφοράς, το οποίο συχνά περιέχεται σε ένα κείμενο του Παγκοσμίου Ιστού, παρουσιάζει τα δεδομένα με ένα φιλικό προς τον εκάστοτε χρήστη τρόπο. Το δεύτερο από τα πιο πάνω διαγράμματα δείχνει ένα σύστημα με πολλούς μη-κινητούς πράκτορες, οι οποίοι συνεργάζονται, για να εκτελέσουν μία συγκεκριμένη εργασία. Εδώ ένα πρόγραμμα συντονιστής ενεργεί ως το σημείο επικοινωνίας για τους πράκτορες και μάλιστα δρα ως ένας πίνακας που περιέχει πληροφορίες για τον καθένα πράκτορα καθώς και μία λίστα από αρμοδιότητες που πρέπει αυτός να φέρει εις πέρας.

Δεύτερον, οι κινητοί πράκτορες (mobile agents) λογισμικού, με άλλα λόγια εφαρμογές που ταξιδεύουν στους υπολογιστές ενός δικτύου, προκειμένου να υλοποιούν διάφορες εργασίες αυτού. Το κυριότερο πλεονέκτημά τους είναι η εξοικονόμηση εύρους ζώνης: δεν υπάρχει ανάγκη για αποστολή ενδιάμεσων αποτελεσμάτων στον εκάστοτε υπολογιστή-πελάτη για επεξεργασία. Τα πάντα επεξεργάζονται στους υπολογιστές που φιλοξενούν τους πράκτορες. Μία τυπική αρχιτεκτονική για ένα κινητό πράκτορα λογισμικού φαίνεται στην πιο κάτω Εικόνα:



Εικόνα 7: Η αρχιτεκτονική κινητού πράκτορα λογισμικού

Το παραπάνω σχήμα απεικονίζει τους δομικούς λίθους ενός συστήματος κινητών πρακτόρων λογισμικού, που είναι οι εξής:

- Η κλάση Agent, που δημιουργεί έναν πράκτορα και καθορίζει τη λειτουργικότητά του.
- Η κλάση AgentHost, η οποία καθορίζει τις ιδιότητες λογισμικού που προσφέρονται από τους υπολογιστές που φιλοξενούν τον πράκτορα. Εκτελεί πολλές εργασίες, συμπεριλαμβανομένων της εκτέλεσης του κινητού πράκτορα, της τροποποίησής του και της παροχής δεδομένων σε αυτόν.
- Η κλάση AgentInterface, που καθορίζει τη διεπαφή για τον πράκτορα, με την οποία ο πράκτορας μπορεί να επικοινωνεί με τον υπολογιστή που τον φιλοξενεί.
- Η κλάση AgentIdentity, η οποία παρέχει πληροφορίες σχετικά με την ταυτότητα ενός πράκτορα, όπως τι δυνατότητες έχει και μέσω ποιων διεπαφών μπορεί να δουλέψει.

Στο σημείο αυτό πρέπει να τονιστεί πως στη συνέχεια θα αναλυθεί περαιτέρω η τεχνολογία των μη-κινητών πρακτόρων λογισμικού, γιατί, εξαιτίας των χαρακτηριστικών τους, χρησιμοποιούνται ευρέως τόσο σε εμπορικές όσο και σε διαδικτυακές εφαρμογές.

[22]

2.5.3 Τα Χαρακτηριστικά των Κινητών Πρακτόρων Λογισμικού

Όπως προαναφέρθηκε, ένας πράκτορας λογισμικού ορίζεται ως ένα αντικείμενο που έχει τη δική του συμπεριφορά, κατάσταση καθώς και τοποθεσία. Οι κινητοί πράκτορες λογισμικού έχουν την ιδιότητα να μεταναστεύουν από έναν υπολογιστή σε έναν άλλο και να συνεχίζουν εκεί την εκτέλεσή τους από το σημείο που σταμάτησαν (κινητικότητα). Κατά τη μετακίνησή τους αυτή, εκτός από τον κώδικα, οι κινητοί πράκτορες παίρνει μαζί τους και την πληροφορία κατάστασης (πχ ορίσματα και τιμές τοπικών μεταβλητών ρουτινών που έχουν κληθεί στο ίδιο νήμα εκτέλεσης με αυτούς και δεν έχουν τερματιστεί). Ένα από τα πλέον βασικά χαρακτηριστικά τους είναι η αυτονομία, αφού από τη στιγμή που ξεκινά η εκτέλεσή τους μπορούν ανεξάρτητα να αποφασίζουν σε ποιες τοποθεσίες θα μεταναστεύουν και ποιον κώδικα θα εκτελούν, βασιζόμενοι σε μεταδεδομένα μετακίνησης (mobility metadata). Οι κινητοί πράκτορες αποτελούν μορφή κινητού κώδικα. Το χαρακτηριστικό αυτό της αυτονομίας τους συμβάλλει και στην βελτίωση του κόστους ενός δικτύου, εφόσον η αλληλεπίδραση ενός κινητού πράκτορα με την υφιστάμενη σε κάθε περίπτωση πηγή προέλευσής του ελαττώνεται στην απολύτως απαραίτητη. Σε αντίθεση με τα κλασσικά παραδείγματα της απομακρυσμένης αξιολόγησης (remote evaluation) και του κατά απαίτηση κώδικα (code on demand), οι κινητοί πράκτορες είναι ενεργοί, υπό την έννοια πως μπορούν να μετακινούνται μεταξύ υπολογιστών οποιαδήποτε στιγμή κατά την εκτέλεσή τους. Αυτή η ιδιότητα τους καθιστά ένα ιδιαίτερα ισχυρό εργαλείο στην ανάπτυξη κατανεμημένων εφαρμογών πάνω σε ένα δίκτυο υπολογιστών. Εκτός από την κινητικότητα (mobility), την αυτονομία (independence) και την ενεργητικότητα (activity), που αποτελούν τα βασικότερα χαρακτηριστικά τα

ποία πρέπει να έχει ένας κινητός πράκτορας, υπάρχει και ένας αριθμός από άλλα στοιχεία που τον χαρακτηρίζουν και είναι τα εξής:

- Διαδραστικότητα (interactivity): Ένας κινητός πράκτορας λογισμικού έχει τη δικαιοδοσία να επικοινωνεί και να αλληλεπιδρά με το εκάστοτε περιβάλλον του καθώς και με άλλους κινητούς πράκτορες.
- Προσαρμοστικότητα (adaptability): Οι κινητοί πράκτορες λογισμικού αποκρίνονται στο εκάστοτε περιβάλλον τους ή σε άλλους κινητούς πράκτορες και προσαρμόζονται και οι ίδιοι κατάλληλα.
- Πρωτοβουλία (initiative): Κάθε κινητός πράκτορας λογισμικού είναι προσανατολισμένος προς κάποιο συγκεκριμένο στόχο και λαμβάνει κατάλληλες πρωτοβουλίες κάθε φορά που ανταποκρίνεται στο εκάστοτε περιβάλλον του.
- Νοημοσύνη (intelligence): Οι κινητοί πράκτορες λογισμικού διαθέτουν νοημοσύνη βασισμένη σε κάποια γνώση, ώστε να γίνεται αποτελεσματικότερη η αλληλεπίδραση με το εκάστοτε περιβάλλον τους.
- Συντονισμός (coordination): Οι κινητοί πράκτορες λογισμικού έχουν την ικανότητα να πραγματοποιούν μεταφορά δεδομένων που μοιράζονται μεταξύ των κινητών πρακτόρων ενός περιβάλλοντος.
- Μάθηση (learning): Αναφέρεται στην ικανότητα ενός κινητού πράκτορα λογισμικού να παίρνει πληροφορίες σχετικά με το εκάστοτε περιβάλλον, οι οποίες του επιτρέπουν να προσαρμόζει ανάλογα τη συμπεριφορά του.
- Συνεργασία (cooperation): Οι κινητοί πράκτορες λογισμικού συντονίζονται και συνεργάζονται μεταξύ τους για την επίτευξη ενός κοινού στόχου.
- Ικανότητα ανάκαμψης (recovery ability): Ένας κινητός πράκτορας λογισμικού αντιμετωπίζει τα όποια σφάλματα εμφανίζονται κατά τη διάρκεια ζωής τους.

[23]

2.5.4 Εφαρμογές των Κινητών Πρακτόρων Λογισμικού

Όπως αναφέρθηκε προηγουμένα, η τεχνολογία των κινητών πρακτόρων λογισμικού προσφέρει σημαντικά πλεονεκτήματα σε ένα εκτεταμένο εύρος εφαρμογών, οι οποίες απαιτούν καταναμημένους υπολογισμούς και ασύγχρονη επικοινωνία. Ο καταναμημένος χαρακτήρας των κινητών πρακτόρων οδηγεί σε βελτιστοποιημένη χρήση υπολογιστικών πόρων. Η δυνατότητα ασύγχρονης επικοινωνίας ελαττώνει την ανάγκη για επικοινωνία σε πραγματικό χρόνο για τον εκάστοτε τελικό χρήστη. Η δυνατότητα αυτόνομης μεταφοράς και εκτέλεσης κώδικα σε διαφορετικά περιβάλλοντα συμβάλλει στην ελάττωση του κόστους του εύρους ενός δικτύου και στην ανάπτυξη εφαρμογών με αυξημένη διαλειτουργικότητα. Αρχικά, λόγω των χαρακτηριστικών της μεταφοράς, της αυτονομίας και της ασύγχρονης επικοινωνίας, οι κινητοί πράκτορες χρησιμοποιούνται, για να παρέχουν ανεκτικότητα σε λάθη δικτύου (network fault tolerance). Επίσης, η προσαρμοστικότητα που παρουσιάζουν σε αλλαγές τόσο της κατάστασης ενός δικτύου όσο και του περιβάλλοντος αυτού του δικτύου κάνει την τεχνολογία των κινητών πρακτόρων κατάλληλη για την προστασία κρίσιμων εφαρμογών από σφάλματα που προέρχονται από αναξιόπιστα δίκτυα.

Μία πολύ σημαντική εφαρμογή της τεχνολογίας των κινητών πρακτόρων λογισμικού έγκειται στη συλλογή δεδομένων πάνω σε ένα δίκτυο (πχ το Διαδίκτυο). Για παράδειγμα, ένας κινητός πράκτορας μπορεί να αποστέλλεται σε έναν εξυπηρετητή Ιστού, όπου υπάρχει μία μεγάλη βάση δεδομένων, να εκτελεί τοπικά μία αναζήτηση και να επιστρέφει με το αποτέλεσμα στον υπολογιστή που ξεκίνησε η εκτέλεσή του. Επιπρόσθετα, κινητοί πράκτορες διατίθενται για την αποδοτική ανανέωση απομακρυσμένων υπηρεσιών σε καταναμημένες εφαρμογές. Για παράδειγμα, ένας κινητός πράκτορας μπορεί να αξιοποιείται ως ένα κεντρικό σημείο ελέγχου για την ανανέωση των λειτουργιών των κόμβων ενός δικτύου. Ακόμα, οι κινητοί πράκτορες είναι κατάλληλοι για εφαρμογές που απαιτούν συμπαγή επικοινωνία (robustness) μέσα από μη αξιόπιστα δίκτυα και συνδέσεις χαμηλού εύρους ζώνης δικτύου. Παραδείγματα εφαρμογών τους περιλαμβάνουν μεταξύ άλλων διαχείριση δικτύων, εφαρμογές ηλεκτρονικού εμπορίου και ηλεκτρονικές δημοπρασίες. Συγκεκριμένα, σε ό,τι αφορά το ηλεκτρονικό εμπόριο, ανοίγονται νέες προοπτικές. Για παράδειγμα, κάποιος πράκτορας έχει την ικανότητα να αναλαμβάνει να ενημερώνει διαφορετικούς Ιστοτόπους για νέα προϊόντα, χωρίς να είναι απαραίτητη η ύπαρξη κοινής βάσης δεδομένων ή άλλης υποδομής επικοινωνίας ανάμεσα σε αυτούς τους Τύπους.

Από την πλευρά των χρηστών, κάποιοι κινητός πράκτορας λογισμικού μπορεί να αναλαμβάνει να αναζητάει προϊόντα τα οποία ταιριάζουν με κάποια κριτήρια αναζήτησης αυτών των χρηστών και να τους παρουσιάζει τα αποτελέσματα πραγματοποιώντας και κάποια σύγκριση μεταξύ τους. Οι κινητοί πράκτορες που χρησιμοποιούνται σε τέτοιου είδους εφαρμογές πρέπει να είναι αξιόπιστοι, τόσο από άποψη ασφάλειας όσο και ανοχής σε σφάλματα. Επιπρόσθετα, η ίδια η τεχνολογία των κινητών πρακτόρων οδηγεί σε ένα νέο είδος ηλεκτρονικού εμπορίου, το λεγόμενο κινητό εμπόριο (mobile commerce).

Αποδοτική εφαρμογή των κινητών πρακτόρων λογισμικού συνιστά και η περίπτωση των κινητών χρηστών ενός δικτύου. Αυτό που χαρακτηρίζει αυτήν την κατηγορία χρηστών είναι η δαπανηρή και ευαίσθητη σύνδεσή τους με το εκάστοτε δίκτυο καθώς και ο κατά κανόνα ελλιπής σε πόρους υπολογιστής που χρησιμοποιούν για τη σύνδεση. Για την υποστήριξη τέτοιων κινητών χρηστών οι κινητοί πράκτορες είναι ιδανική λύση, αφού ταιριάζουν ιδανικά σε περιπτώσεις όπου δεν είναι δυνατή μία συνεχόμενη και σταθερή σύνδεση. Προς αυτήν την κατεύθυνση έχουν γίνει προσπάθειες για την εκμετάλλευση του μοντέλου των κινητών πρακτόρων στην διαχείριση δικτύων κινητών συσκευών. Τέλος, πολύ σημαντικός είναι ο ρόλος των κινητών πρακτόρων και στην ταχεία ανάπτυξη νέων τεχνολογιών, όπως είναι, για παράδειγμα, η πλατφόρμα κατανεμημένης επικοινωνίας CORBA (Common Object Request Broker) Τέλος, σε κάποιες πρόσφατες ερευνητικές προσπάθειες έχει γίνει προσπάθεια για τη χρήση των κινητών πρακτόρων στην ανάπτυξη πλεγμάτων υποδομών.

[21] [23]

2.5.5 Σύγκριση των Κινητών Πρακτόρων Λογισμικού

Η τεχνολογία των κινητών πρακτόρων λογισμικού λύνει ή τουλάχιστον υπόσχεται ότι θα λύσει άμεσα διάφορα προβλήματα σε ποικίλα μέτωπα, που άλλες αρχιτεκτονικές λογισμικού δεν κατάφεραν. Συγκεκριμένα, η παρούσα αρχιτεκτονική παρουσιάζει τα παρακάτω σημαντικά πλεονεκτήματα:

- Ασύγχρονη και αυτόνομη εκτέλεση διεργασιών: Μετά από την είσοδο ενός κινητού πράκτορα λογισμικού σε ένα κατανεμημένο υπολογιστικό σύστημα ενός δικτύου, ο εκάστοτε χρήστης δύναται να εκτελεί άλλες δραστηριότητες, χωρίς να απαιτείται να αλληλεπιδρά με τον πράκτορα αυτό, ακόμα και να αποσυνδέεται από το τρέχον δίκτυο μετά την αποστολή του συγκεκριμένου πράκτορα στον έναν απομακρυσμένο υπολογιστή του παρόντος συστήματος, εφόσον πρόκειται για πρόσβαση σε απομακρυσμένες εφαρμογές. Για την υποστήριξη τέτοιου είδους συμπεριφορών χρησιμοποιείται ένας μηχανισμός αποθήκευσης και προώθησης των κινητών πρακτόρων.
- Μείωση της εκάστοτε δικτυακής κυκλοφορίας και της απαιτούμενης κάθε φορά υπολογιστικής ισχύος, από τη στη μεριά του εκάστοτε πελάτη: Στην περίπτωση των κινητών πρακτόρων ανταλλαγές δεδομένων πολύ μεγάλου όγκου γίνονται τοπικά από τους εκάστοτε κόμβους που περιέχουν αυτήν την πληροφορία. Συνεπώς, οι υπολογιστές στην πλευρά του εκάστοτε πελάτη επικεντρώνονται στην εκτέλεση σχετικά απλών λειτουργιών. Αυτό το γεγονός επιτρέπει την υλοποίηση «λεπτών πελατών» (“thin clients”). Με αυτόν τον τρόπο, οι κινητοί πράκτορες είναι σε θέση να μειώνουν το κόστος, σχετικά με εύρος ζώνης ενός δικτύου, συγκρινόμενοι με παλαιότερες μεθόδους κατανεμημένης επικοινωνίας.
- Μείωση των χρονικών καθυστερήσεων, λόγω δικτύου: Οι κινητοί πράκτορες, λόγω των κινητών και προσαρμοστικών ιδιοτήτων τους, προσφέρουν μεγάλες δυνατότητες για απόκριση σε πραγματικού χρόνου συστήματα όπου η χρονική καθυστέρηση δεν μπορεί να γίνεται αποδεκτή και δημιουργεί προβλήματα.
- Αυξημένη ευρωστία: Η μείωση της εξάρτησης μίας κατανεμημένης εφαρμογής από την διαθεσιμότητα ενός δικτύου είναι ευπρόσδεκτη και λύνει ένα από τα βασικά προβλήματα που αντιμετωπίζει η αρχιτεκτονική πελάτη-εξυπηρετητή. Στα συστήματα των κινητών πρακτόρων επέρχεται αυτή η μείωση της εξάρτησης, διότι, από τη στιγμή που ο κώδικας του εκάστοτε πελάτη βρίσκεται στον ίδιο κόμβο με εκείνον του εξυπηρετητή, η επικοινωνία είναι τοπική.
- Αυτοματοποίηση της επεξεργασίας κατανεμημένων υπολογισμών: Οι κινητοί πράκτορες έχουν ενσωματωμένα δρομολόγια τα οποία καθορίζουν τι είδους δραστηριότητες πρέπει να εκτελούνται κατά περίπτωση, με αποτέλεσμα να μην απαιτείται συνεχής αλληλεπίδραση με τον εκάστοτε χρήστη.

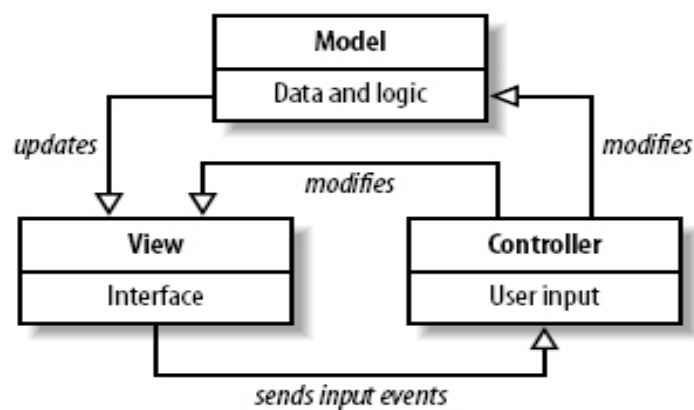
- Αποκεντρωμένη και κατά τόπους επεξεργασία (έλεγχος και διαχείριση): Η κλωνοποίηση των πρακτόρων επιτρέπει την (αυτοματοποιημένη) κατανομή των υπολογιστικών συνιστωσών προηγούμενων συγκεντρωτικών προγραμμάτων.
- Ικανότητα εκτέλεσης σε ετερογενή περιβάλλοντα: Οι κινητοί πράκτορες έχουν την ικανότητα να εκτελούνται σε περιβάλλοντα με διαφορετικές υποδομές. Έτσι, επεκτείνεται η διαλειτουργικότητα σε κατανεμημένα συστήματα και πληθώρα εφαρμογών μπορούν να εκτελούνται σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα και να επικοινωνούν μέσω των κινητών πρακτόρων.
- Ευελιξία: Η κατά απαίτηση διανομή λογισμικού ή η παροχή υπηρεσιών γίνεται εφικτή, μίας και κινητοί πράκτορες μπορούν να κατεβούν στιγμιαία σε οποιοδήποτε κόμβο ενός πελάτη ή ενός εξυπηρετητή.
- Κλιμάκωση εφαρμογών: Λόγω της δυναμικής εκτέλεσης των κινητών πρακτόρων, διευκολύνεται η ανάπτυξη περισσότερο κλιμακούμενων εφαρμογών.
- Ανασχεδιασμό του συστήματος και μετά την κατασκευή ενός δικτύου: Κατά το σχεδιασμό ενός συστήματος με βάση την αρχιτεκτονική πελάτη-εξυπηρετητή καθορίζονται οι ρόλοι του πελάτη και του εξυπηρετητή, παίρνοντας εξαρχής αποφάσεις σχετικά με τους περιορισμούς του εύρους ζώνης, την κίνηση ενός δικτύου, τον αριθμό των πελατών και των εξυπηρετητών και άλλων συναφών παραγόντων. Αν οι αρχικές προδιαγραφές είναι λανθασμένες, τότε η εκάστοτε εφαρμογή δεν ανταποκρίνεται σωστά. Η τεχνολογία των κινητών πρακτόρων επιτρέπει να λαμβάνονται λιγότερες αποφάσεις κατά την αρχική σχεδίαση ενός συστήματος, το οποίο πλέον μπορεί εύκολα στο μέλλον να ανασχεδιασθεί.

[22] [23]

2.6 Η Αρχιτεκτονική Ανάπτυξης Λογισμικού, MVC

2.6.1 Εισαγωγή στην MVC

Η αρχιτεκτονική MVC (Model-View-Controller) (απόδοση στα Ελληνικά: Μοντέλου-Προβολής-Ελεγκτή) είναι ένα μοντέλο αρχιτεκτονικής ανάπτυξης λογισμικού, το οποίο χρησιμοποιείται για την δημιουργία περιβαλλόντων αλληλεπίδρασης με τους χρήστες. Στο μοντέλο αυτό η εκάστοτε εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη, όπως φαίνεται και στην παρακάτω Εικόνα:



Εικόνα 8: Η διασύνδεση των τμημάτων της αρχιτεκτονικής MVC

Σύμφωνα με την ανώτερη Εικόνα στην τρέχουσα αρχιτεκτονική διαχωρίζεται η παρουσίαση της εκάστοτε πληροφορίας στον εκάστοτε χρήση από τη μορφή που αποθηκεύεται στο εκάστοτε σύστημα. Το κύριο μέρος του συγκεκριμένου μοντέλου είναι το αντικείμενο Model το οποίο διαχειρίζεται την ανάκτηση-αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο View χρησιμοποιείται μόνο, για να παρουσιάζεται η πληροφορία στον εκάστοτε χρήστη (πχ με γραφικό τρόπο). Το τρίτο μέρος του είναι το αντικείμενο Controller που δέχεται κατάλληλη σε κάθε περίπτωση είσοδο και στέλνει εντολές στα αντικείμενα Model και View.

Εκτός από το να διαιρείται η εκάστοτε εφαρμογή σε τρία διασυνδεδεμένα μέρη, η σχεδίαση MVC ορίζει και τις αλληλεπιδράσεις των τμημάτων αυτών, όπως στη συνέχεια:

- Το αντικείμενο Model ενημερώνει τα αντικείμενα View και Controller, όταν υπάρχει αλλαγή στα δεδομένα. Αυτή η ενημέρωση επιτρέπει στα View και Controller να ενημερώνουν με τη σειρά τους την προβαλλόμενη σε κάθε περίπτωση γραφική απεικόνιση.
- Το αντικείμενο View αναπαριστά με γραφικό τρόπο την εκάστοτε πληροφορία που περιέχει το αντικείμενο Model, δημιουργώντας γραφική παρουσίαση στον εκάστοτε χρήστη.
- Το αντικείμενο Controller στέλνει εντολές στο αντικείμενο Model και ενημερώνει την κατάστασή του. Μπορεί, επίσης, να στέλνει εντολές, ώστε να πραγματοποιείται η κατάλληλη αναπαράσταση σε κάθε φάση των δεδομένων του αντικειμένου Model μέσω του αντικειμένου View.

[24]

2.6.2 Εφαρμογές της MVC

Παρόλο που η αρχιτεκτονική MVC αρχικά αναπτύχθηκε για επιτραπέζιους υπολογιστές, το συγκεκριμένο μοντέλο έχει υιοθετηθεί σε μεγάλο βαθμό ως μία αρχιτεκτονική για διαδικτυακές εφαρμογές σε μεγάλες γλώσσες προγραμματισμού. Τα πιο πολλά εμπορικά και μη εμπορικά πλαίσια λογισμικού (software frameworks) διαδικτυακών εφαρμογών έχουν δημιουργηθεί χάρη στη συγκεκριμένη αρχιτεκτονική. Αν, τώρα, αναλογιστεί κανείς πόσο σημαντικός είναι ο ρόλος των πλαισίων για την υλοποίηση εφαρμογών Διαδικτύου, εφόσον κάνουν αυτή την υλοποίηση περισσότερο ασφαλή, γρήγορη και ευχάριστη, διότι διαθέτουν ένα σύνολο προκαθορισμένων (by default) λειτουργιών, τότε αντιλαμβάνεται πόσο ιδιαίζουσας σημασίας είναι η αρχιτεκτονική MVC.

[24]

2.6.3 Σύγκριση της MVC

Η αρχιτεκτονική MVC έχει πολλά οφέλη αναφορικά με άλλες αρχιτεκτονικές ανάπτυξης λογισμικού, από τα οποία τα κυριότερα αναφέρονται παρακάτω:

- Ο διαχωρισμός των προβλημάτων (separation of concerns): Αυτό είναι και το πιο βασικό πλεονέκτημα της MVC. Ουσιαστικά με χρήση της συγκεκριμένης αρχιτεκτονικής δημιουργείται μία εφαρμογή η οποία, όπως προαναφέρθηκε, έχει τρία επίπεδα, το επίπεδο Model, το επίπεδο View και το επίπεδο Controller. Το καθένα από τα τρία αυτά επίπεδα επιτελεί ξεχωριστό έργο σε ταυτόχρονη συνεργασία με τα άλλα επίπεδα. Μία σωστή MVC εφαρμογή είναι εκείνη που τα τρία επίπεδα είναι ξεκάθαρα καθορισμένα και δεν συμπλέκονται
- Η επεκτασιμότητα (modularity): Τα προγράμματα που είναι φτιαγμένα με MVC αρχιτεκτονική έχουν βασικό χαρακτηριστικό ότι είναι επεκτάσιμα, δηλαδή υπάρχει η δυνατότητα να προστίθενται λειτουργίες σε αυτά ή να τροποποιούνται κάποιες από τις ήδη υπάρχουσες λειτουργίες τους και να λαμβάνονται διαφορετικά αποτελέσματα.
- Η ελεγχιμότητα (testability): Οι MVC εφαρμογές είναι ελέγξιμες και με τον τρόπο αυτόν συντηρούνται πιο εύκολα.

[24]

2.7 Το Πλαίσιο Λογισμικού

2.7.1 Εισαγωγή στο Πλαίσιο Λογισμικού

Στην επιστήμη της Πληροφορικής ένα πλαίσιο λογισμικού (software framework) είναι μία αφηρημένη έννοια στην οποία κομμάτια λογισμικού που παρέχουν γενικές λειτουργίες δύναται να αλλάξουν επιλεκτικά με την προσθήκη αποσπασμάτων κώδικα του εκάστοτε προγραμματιστή. Με τον τρόπο αυτό ένα γενικό λογισμικό εξειδικεύεται, ώστε να εξυπηρετήσει της ανάγκες μία συγκεκριμένης εφαρμογής. Ένα πλαίσιο λογισμικού είναι ένα καθολικό, επαναχρησιμοποιήσιμο περιβάλλον λογισμικού, που παρέχει ιδιαίτερη λειτουργικότητα, προκειμένου να διευκολύνει την ανάπτυξη ειδικού λογισμικού το οποίο θα υποστηρίζει τις εξατομικευμένες ανάγκες εφαρμογών. Για το σκοπό αυτόν τα πλαίσια λογισμικού παρέχουν υποστηρικτικά προγράμματα, μεταγλωττιστές, βιβλιοθήκες κώδικα, και Διεπαφές

Προγραμματισμού Εφαρμογών (Application Programming Interfaces-APIs, ώστε να συγκεντρώνουν όλα τα απαραίτητα συστατικά που καθιστούν δυνατή την υλοποίηση ενός έργου.

Τα πλαίσια λογισμικού, λοιπόν, παρέχουν ένα σύνολο διακριτών γνωρισμάτων που τα κάνουν να ξεχωρίζουν από τις κοινές βιβλιοθήκες κώδικα και τα συνήθη προγράμματα, από τα οποία τα βασικότερα είναι τα μετέπειτα:

- Αντιστροφή ελέγχου (inversion of control): Σε ένα πλαίσιο λογισμικού ο έλεγχος της ροής του συνολικού προγράμματος δεν υπαγορεύεται από τον εκάστοτε προγραμματιστή, αλλά από το παρόν πλαίσιο.
- Προκαθορισμένη συμπεριφορά (default behavior): Ένα πλαίσιο λογισμικού ακολουθεί μία προκαθορισμένη συμπεριφορά. Προσφέρει, δηλαδή, προκαθορισμένες προγραμματιστικές κατευθύνσεις, οι οποίες οφείλουν σε κάθε περίπτωση να είναι χρήσιμες και λειτουργικές και να συντελούν με τον καλύτερο δυνατόν τρόπο στην ολοκλήρωση ενός συγκεκριμένου, λογισμικού.
- Επεκτασιμότητα (extensibility): Ένα πλαίσιο λογισμικού μπορεί να εξατομικεύσει τις γενικευμένες λειτουργίες του, με στόχο αυτές να ανταποκρίνονται στα ζητούμενα μίας εξειδικευμένης εφαρμογής. Κάτι τέτοιο είναι εφικτό από τον εκάστοτε προγραμματιστή με την προσθήκη κώδικα στο πλαίσιο ή
- Μη τροποποιήσιμο κώδικα (non-modifiable code): Σε γενικές γραμμές ο κώδικας ενός πλαισίου λογισμικού δεν πρέπει να επιδέχεται τροποποιήσεις από τον εκάστοτε προγραμματιστή, παρόλο που, όπως μόλις αναφέρθηκε, ο εκάστοτε προγραμματιστής δύναται να τον επεκτείνει.

[25]

2.7.2 Το Πλαίσιο Λογισμικού Διαδικτυακών Εφαρμογών, Apache Struts 1

Το πλαίσιο λογισμικού διαδικτυακών εφαρμογών Apache Struts 1 είναι ένα πλαίσιο λογισμικού ανοιχτού κώδικα ειδικά κατασκευασμένο να βοηθάει στην υλοποίηση εφαρμογών του Διαδικτύου. Πρόκειται για ένα έργο της εταιρείας ASF (Apache Software Foundation), γνωστό, επίσης στο ευρύ κοινό με την ονομασία Jakarta Struts. Το παρόν πλαίσιο χρησιμοποιεί και επεκτείνει την Java Servlet API και παροτρύνει τους προγραμματιστές Διαδικτύου να υιοθετήσουν την MVC αρχιτεκτονική στις κατασκευές τους.

Σε μία τυπική εφαρμογή του Παγκόσμιου Ιστού προγραμματισμένη σε γλώσσα Java, ο ο εκάστοτε πελάτης θα καλέσει τον εκάστοτε εξυπηρετητή Ιστού μέσω μίας φόρμας του Παγκόσμιου Ιστού. Στη συνέχεια, η πληροφορία που μόλις γεννήθηκε είτε μεταβιβάζεται σε μία Java Servlet, η οποία αλληλεπιδρά με μία βάση δεδομένων, είτε μεταβιβάζεται σε μία JSP, προκειμένου να παράγει ένα HTML έγγραφο απόκρισης. Οι δύο προαναφερθείσες προσεγγίσεις θεωρούνται συχνά ανεπαρκείς για μεγάλα έργα, επειδή συγχέουν τη λογική ενός έργου με την παρουσίασή του και κατά επέκταση δυσκολεύουν τη συντήρησή του. Ο σκοπός και ο στόχος, λοιπόν, του πλαισίου λογισμικού διαδικτυακών εφαρμογών Apache Struts 1 έγκειται στο διαχωρισμό μεταξύ των αντικειμένων Model, View και Controller σε εφαρμογές που εκμεταλλεύονται την αρχιτεκτονική MVC. Έτσι, το τρέχον πλαίσιο παρέχει το αντικείμενο Controller ως μία servlet, που πολύ συχνά είναι γνωστή με το όνομα “ActionServlet” και διευκολύνει τη συγγραφή προτύπων για το αντικείμενο View συνήθως με χρήση της JSP, αλλά και της XML ή της Velocity. Ο εκάστοτε προγραμματιστής, τώρα, είναι υπεύθυνος για τη συγγραφή του κώδικα του αντικειμένου Model μίας εφαρμογής και για τη δημιουργία ενός κύριου αρχείου διαμόρφωσης, το οποίο συνήθως διατίθεται με την ονομασία “struts-config.xml”, και αποσκοπεί στο δέσιμο των προαναφερόμενων τριών αντικειμένων (Model-View-Controller) μεταξύ τους. Ύστερα, τα αιτήματα των πελατών αποστέλλονται και αποθηκεύονται στο αντικείμενο Controller, σε μία φόρμα που τις πιο πολλές φορές καλείται “Actions” και καθορίζεται στο προαναφερόμενο αρχείο διαμόρφωσης. Για κάθε αίτημα που λαμβάνει το αντικείμενο Controller καλεί την κατάλληλη κλάση “Action” που αλληλεπιδρά με το κατάλληλο απόσπασμα κώδικα του αντικειμένου Model. Έτσι, επιστρέφεται ένα αλφαριθμητικό που οδηγεί το αντικείμενο Controller να εμφανίσει την αρμόζουσα σε κάθε περίπτωση Ιστοσελίδα στον εκάστοτε χρήστη που αιτήθηκε περιεχόμενο από τον εκάστοτε διακομιστή. Εδώ, να επισημανθεί ότι η διαθέσιμη κάθε φορά πληροφορία περνάει ανάμεσα στα αντικείμενα Model και View με τη μορφή συγκεκριμένων JavaBeans στοιχείων.

[26]

2.8 Η Γλώσσα Μοντελοποίησης Λογισμικού, UML

2.8.1 Εισαγωγή στη UML

Η γλώσσα UML (Unified Modeling Language), (απόδοση στα Ελληνικά: Ενοποιημένη Γλώσσα Μοντελοποίησης) είναι μία γραφική γλώσσα η οποία χρησιμοποιείται για την οπτική παράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε λογισμικό. Η UML στοχεύει στο σχεδιασμό αντικειμενοστραφών συστημάτων. Το εκάστοτε σχέδιο είναι μία απλοποιημένη παράσταση της εκάστοτε πραγματικότητας.

Ο σχεδιασμός αποσκοπεί στην κατανόηση του εκάστοτε συστήματος που πρόκειται να αναπτυχθεί. Έτσι, δημιουργώντας ένα σχέδιο επιτυγχάνονται τέσσερις στόχοι, που είναι οι παρακάτω:

- I. Παριστάνεται οπτικά το εκάστοτε υπό κατασκευή σύστημα.
- II. Προσδιορίζεται η δομή και η συμπεριφορά του εκάστοτε υπό κατασκευή συστήματος.
- III. Δημιουργείται το πρότυπο-βάση για την κατασκευή του εκάστοτε συστήματος.
- IV. Τεκμηριώνονται οι ληφθείσες αποφάσεις για την υλοποίηση του εκάστοτε υπό κατασκευή συστήματος.

Σε όλους τους τεχνολογικούς τομείς ο εκάστοτε σχεδιασμός βασίζεται σε τέσσερις βασικές αρχές, οι οποίες είναι οι επόμενες:

- I. Η επιλογή μίας κατηγορίας σχεδίων έχει επίπτωση στον τρόπο και την μορφή επίλυσης ενός προβλήματος.
- II. Όλα τα σχέδια εκφράζονται σε διαφορετικές βαθμίδες ακρίβειας.
- III. Τα καλύτερα σχέδια σχετίζονται με την πραγματικότητα.
- IV. Ένα είδος σχεδίων δεν είναι ποτέ αρκετό.

[27]

2.8.2 Τα Βασικά Συστατικά της UML

Η γλώσσα UML περιλαμβάνει τρία βασικά στοιχεία, που αναφέρονται στη συνέχεια:

- I. Οντότητες.
- II. Σχέσεις.
- III. Διαγράμματα.

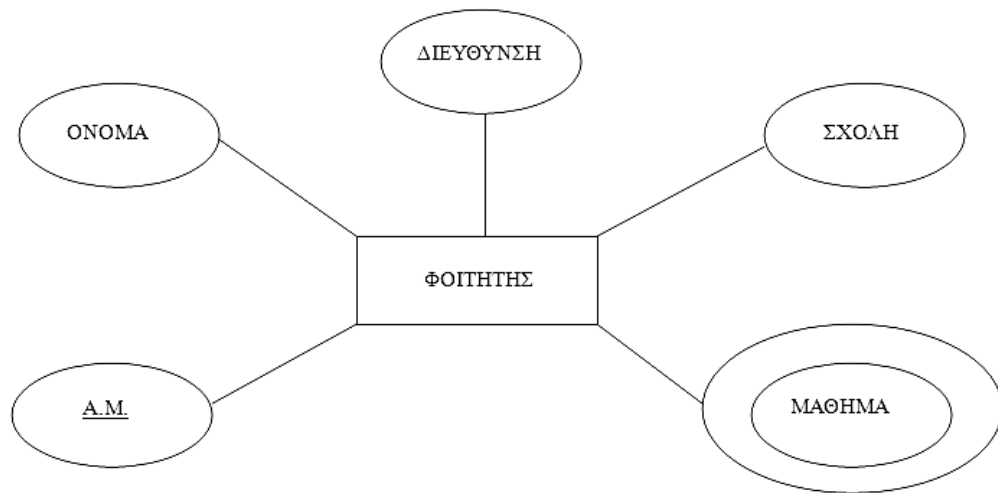
Ας ριχθεί μία στενότερη ματιά σε καθένα από τα παραπάνω στοιχεία:

- I. Μία οντότητα (entity) αποτελεί ένα αντικείμενο ενδιαφέροντος στον πραγματικό κόσμο το οποίο μπορεί να είναι, για παράδειγμα, άνθρωποι, μέρη, αντικείμενα, γεγονότα, έννοιες. Το στιγμιότυπο (instance) μίας οντότητας είναι μία συγκεκριμένη περίπτωση ενός τύπου μίας οντότητας. Ο τύπος μίας οντότητας είναι μία συλλογή χαρακτηριστικών που περιγράφουν την οντότητα. Οι διάφοροι τύποι οντοτήτων (πχ ΕΡΓΑΖΟΜΕΝΟΣ, ΦΟΙΤΗΤΗΣ) παριστάνονται στο Μοντέλο Οντοτήτων-Συχνοτήτων (Μοντέλο Ο\Σ) (Entity-Relationship Model-ER Model) με ένα ορθογώνιο, όπως αντικατοπτρίζεται στο παράδειγμα της παρακάτω Εικόνας:



Εικόνα 9: Παράδειγμα μίας οντότητας του Μοντέλου Ο\Σ της UML

Κάθε οντότητα έχει διάφορα στοιχεία που την προσδιορίζουν. Ένα τέτοιο στοιχείο ονομάζεται ιδιότητα\χαρακτηριστικό\πεδίο (property\attribute\field). της οντότητας. Τα χαρακτηριστικά χωρίζονται σε μονότιμα (single-valued), τα οποία έχουν μόνο μία τιμή, και πλειότιμα (multi-valued), τα οποία έχουν ένα σύνολο από τιμές. Στο Μοντέλο Ο\Σ, οι ιδιότητες που έχει μία οντότητα παριστάνονται μέσα σε έλλειψη, με υπογραμμισμένο το πρωτεύον κλειδί. Τα πλειότιμα χαρακτηριστικά μίας οντότητας (πχ ΦΟΙΤΗΤΗΣ) παριστάνονται μέσα σε έλλειψη με διπλό περίγραμμα, όπως φαίνεται και στο παράδειγμα της κατώτερης Εικόνας:



Εικόνα 10: Παράδειγμα των ιδιοτήτων μίας οντότητας του Μοντέλου Ο\Σ της UML

- II. Στη γλώσσα UML ορίζονται τέσσερις βασικές σχέσεις, που παρουσιάζονται πιο κάτω:
- A. Εξάρτηση (dependency): Δηλώνει πως μία αλλαγή σε μία οντότητα θα επηρεάσει μία άλλη, αλλά όχι απαραίτητα και το αντίστροφο. Παριστάνεται με μία διακεκομμένη γραμμή με ανοιχτό βέλος που δείχνει προς την οντότητα που υπάρχει εξάρτηση, όπως δείχνεται και στην επόμενη Εικόνα:

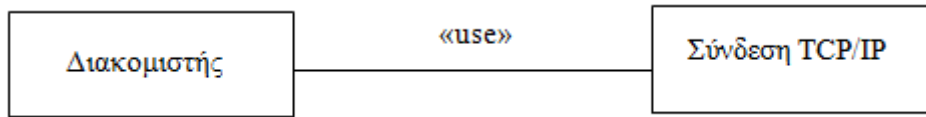


Εικόνα 11: Το σύμβολο της εξάρτησης της UML

Διακρίνουμε 13 μορφές εξάρτησης, οι οποίες είναι οι μετέπειτα:

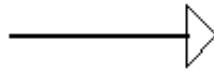
- i. Πρόσβαση (access): Άδεια σε κάποια στοιχεία από ένα τμήμα να έχουν πρόσβαση σε στοιχεία από άλλο τμήμα (access).
- ii. Σύνδεση (binding): Παροχή τιμών σε ένα πρότυπο, για να δημιουργήσει ένα νέο στοιχείο (bind).
- iii. Κλήση (call): Μία μέθοδος καλεί μία άλλη (call).
- iv. Απόρροια (derivation): Ένα στοιχείο μπορεί να υπολογιστεί από κάποιο άλλο (derive).
- v. Φίλος (friend): Ένα στοιχείο μπορεί να έχει πρόσβαση σε στοιχεία άλλης κλάσης παρά τους όποιους περιορισμούς (friend).
- vi. Εισαγωγή (import): Άδεια σε ένα τμήμα να εισάγει και να χρησιμοποιήσει τα στοιχεία ενός άλλου τμήματος (import).
- vii. Δημιουργία (instantiation): Μία μέθοδος μίας κλάσης δημιουργεί αντικείμενα μίας άλλης κλάσης (instantiate).
- viii. Παράμετρος (parameter) Σχέση ανάμεσα σε μία λειτουργία και τις παραμέτρους της (parameter).
- ix. Δημιουργία (realization): Σχέση ανάμεσα σε μία προδιαγραφή και την υλοποίησή της (realize).
- x. Εκλέπτυνση (refinement): Δήλωση για την ύπαρξη απεικόνισης ανάμεσα σε δύο σημασιολογικά επίπεδα (refine).
- xi. Αποστολή (send): Σχέση ανάμεσα στον αποστολέα και τον παραλήπτη ενός μηνύματος (send).
- xii. Ίχνος (trace): Σχέση ανάμεσα σε δύο στοιχεία δύο διαφορετικών μοντέλων, που δεν αποτελεί, όμως, απεικόνιση (trace).
- xiii. Χρήση (usage): Ένα στοιχείο απαιτεί την ύπαρξη ενός άλλου στοιχείου για τη λειτουργία του (πχ call, instantiate, parameter, send) (use).

Σε ένα UML διάγραμμα γράφεται μέσα σε διπλά εισαγωγικά (") το είδος της εξάρτησης (αυτό που εμφανίζεται στο τέλος του ορισμού, στα είδη εξαρτήσεων που αναφέρθηκαν παραπάνω) πάνω από την αντίστοιχη γραμμή με το βέλος, όπως και αναπαρίσταται στο παράδειγμα της παρακάτω Εικόνας:



Εικόνα 12: Παράδειγμα εξάρτησης της UML

- B. Γενίκευση (generalisation): Σημαίνει μία σχέση ανάμεσα σε κάτι γενικό (μία βασική κλάση ή αλλιώς «γονέα») και κάτι ειδικό (μία υποκλάση ή αλλιώς «παιδί» της). Απεικονίζεται με μία συνεχή γραμμή με κλειστό βέλος που δείχνει προς τη βασική κλάση, όπως απεικονίζεται και στην ακόλουθη Εικόνα:



Εικόνα 13: Το σύμβολο της γενίκευσης της UML

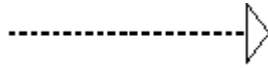
- C. Σύνδεση (association): Αναφέρεται σε αντικείμενα τα οποία συνδέονται με κάποιο τρόπο με άλλα. Εφόσον δύο κλάσεις είναι συνδεδεμένες, τότε μπορεί κανείς να μεταβεί από αντικείμενα της μίας σε αντικείμενα της άλλης. Αποδίδεται συμβολικά με μία ευθεία γραμμή ανάμεσα σε δύο αντικείμενα, όπως δείχνεται και στην πιο κάτω Εικόνα:



Εικόνα 14: Το σύμβολο της σύνδεσης της UML

Σχετικά με την εκάστοτε σύνδεση δύο αντικειμένων στη γλώσσα UML μπορεί κανείς να παρατηρήσει τα εξής:

- Αν μία σύνδεση δεν είναι αμφίδρομη, τότε η κατεύθυνσή της μπορεί να οριστεί με ένα ανοιχτό βέλος.
 - Το όνομα μίας σύνδεσης μπορεί να γραφεί πάνω από τη γραμμή, ενώ η κατεύθυνση του ονόματος αυτής ορίζεται από ένα βέλος πλάι στο όνομα. Ο αριθμός που δηλώνει πόσα αντικείμενα αντιστοιχούν σε κάθε αντικείμενο στην άλλη άκρη μίας σχέσης σύνδεσης (πολλαπλότητα (multiplicity)) δηλώνεται από έναν αριθμό (πχ 2) ή μία περιοχή αριθμών (πχ 1..* (για ένα ως πολλά)) πάνω από την αντίστοιχη άκρη της γραμμής της προαναφερόμενης σχέσης.
 - Ένα X πάνω σε μία άκρη της γραμμής μίας σύνδεσης δηλώνει πως δεν προσφέρεται μετάβαση (navigation) προς τη συγκεκριμένη κατεύθυνση.
 - Αν σε μία σχέση τα αντικείμενα απαρτίζουν τμήματα ενός «όλου», τότε αυτή απεικονίζεται ως συγκρότημα (aggregation) με την παράσταση ενός άδειου διαμαντιού στην άκρη του «όλου».
 - Αν σε μία σχέση τα αντικείμενα που απαρτίζουν τμήματα ενός «όλου» έχουν την ίδια διάρκεια ζωής με το «όλο», τότε αυτή απεικονίζεται ως σύνθεση (composition) με την παράσταση ενός γεμάτου διαμαντιού στην άκρη του «όλου».
- D. Υλοποίηση (realisation): Υποδεικνύει πως ο εκάστοτε εξυπηρετούμενος υποστηρίζει μία διεπαφή που ορίζεται από τον εκάστοτε πάροχο. Αναπαρίσταται με μία διακεκομμένη γραμμή με ανοιχτό βέλος που δείχνει από τον πάροχο προς τον εξυπηρετούμενο, όπως αντικατοπτρίζεται και στη μετέπειτα Εικόνα:



Εικόνα 15: Το σύμβολο της υλοποίησης της UML

- III. Η γλώσσα UML ορίζει τέσσερα γραφικά διαγράμματα σχετικά με τις όψεις ενός Μοντέλου ΟΣ, που είναι τα παρακάτω:
- A. Διάγραμμα περιπτώσεων χρήσης (use case diagram): Μοντελοποιεί το πλαίσιο λειτουργίας ενός συστήματος καθώς και τις προδιαγραφές του. Είναι πολύ σημαντικό διάγραμμα στη UML και περιλαμβάνει τέσσερα συστατικά, που είναι τα μετέπειτα:
 - i. Περιπτώσεις χρήσης.
 - ii. Δρώντες (αυτοί που είναι έξω από το σύστημα).
 - iii. Σχέσεις εξάρτησης, γενίκευσης, σύνδεσης.
 - iv. Τα όρια του συστήματος.
 - B. Διάγραμμα δομής (structure diagram), το οποίο εμπεριέχει δύο είδη, τα οποία τεκμηριώνονται στη συνέχεια:
 - i. Διάγραμμα κλάσεων (class diagram): Δείχνει τη στατική δομή των κλάσεων ενός συστήματος και των σχέσεων μεταξύ τους. Συνιστά το βασικότερο διάγραμμα στη UML και συνήθως αποτελείται από τέσσερα στοιχεία, που ταξινομούνται ως εξής:
 - a. Κλάσεις (classes).
 - b. Διεπαφές (interfaces).
 - c. Συνεργασίες (collaborations).
 - d. Συσχετίσεις (relationships).Το μοντέλο των κλάσεων χρησιμοποιείται τόσο κατά τη διάρκεια της ανάλυσης ενός συστήματος, ώστε να περιγράψει τις λειτουργικές απαιτήσεις (functional requirements) αυτού του συστήματος όσο και κατά τη διάρκεια του σχεδιασμού ενός συστήματος, προκειμένου να περιγράψει το λεξιλόγιο του συστήματος (system vocabulary) τις συνεργασίες (collaborations) και το λογικό σχήμα της βάσης δεδομένων (logical database schema).
 - ii. Διάγραμμα αντικειμένων (object diagram): Περιγράφει ένα σύνολο αντικειμένων καθώς και τις σχέσεις τους μία δεδομένη χρονική στιγμή. Συγκαταλέγει αντικείμενα (objects) και συνδέσμους (links) και εφαρμόζεται, για να καταγράψει στατικές δομές αντικειμένων, όντας στην ουσία ένα στιγμιότυπο ενός διαγράμματος κλάσεων ή η στατική όψη ενός διαγράμματος συνεργασίας.
 - C. Διάγραμμα συμπεριφοράς (behavior diagram), που διακρίνεται σε δύο κατηγορίες, οι οποίες είναι οι επόμενες:
 - i. Διάγραμμα καταστάσεων (statechart diagram): Αναλύει ενεργά (reactive) αντικείμενα και προσδιορίζει μία μηχανή καταστάσεων δίνοντας έμφαση στην ροή του ελέγχου από μία κατάσταση σε μία άλλη. Συμπεριλαμβάνει καταστάσεις (states) και μεταβάσεις (transitions). Μία μηχανή καταστάσεων προσδιορίζει τις καταστάσεις (states) που μπορεί να βρεθεί ένα αντικείμενο, τα γεγονότα (events) στα οποία αντιδρά ένα αντικείμενο καθώς και την απόκριση (response) αυτού του αντικειμένου στα συγκεκριμένα γεγονότα.
 - ii. Διάγραμμα δραστηριοτήτων (activity diagram): Παρουσιάζει τη ροή των εργασιών μέσα σε ένα σύστημα και βασικά περιέχει δραστηριότητες (activities), ενέργειες (actions) και μεταβάσεις (transitions). Έτσι, τα διαγράμματα δραστηριοτήτων είναι διαθέσιμα κυρίως για την περιγραφή τεσσάρων στοιχείων, τα οποία είναι τα εξής:
 - a. Περιπτώσεις χρήσης (use cases).
 - b. Ροές εργασιών (workflows).
 - c. Διεργασίες (processes).
 - d. Επιχειρηματικές διαδικασίες (business processes).

- iii. Διάγραμμα αλληλεπίδρασης (interaction diagram) που ταξινομείται σε δύο είδη, τα οποία στοιχειοθετούνται στη συνέχεια:
 - a. Διάγραμμα ακολουθίας (sequence diagram): Απεικονίζει την αλληλεπίδραση των αντικειμένων, μέσω της ανταλλαγής μηνυμάτων, δίνοντας έμφαση στη χρονική αλληλουχία αυτών των μηνυμάτων. Αποτελείται από χειριστές, αντικείμενα και μηνύματα που ανταλλάσσουν τα αντικείμενα μεταξύ τους και βρίσκει εφαρμογή στην περιγραφή του κύκλου ζωής των αντικειμένων αυτών.
 - b. Διάγραμμα συνεργασίας (collaboration diagram): Δείχνει την αλληλεπίδραση των αντικειμένων, μέσω της ανταλλαγής μηνυμάτων, δίνοντας έμφαση στη χρονική αλληλουχία αυτών των μηνυμάτων. Περιέχει χειριστές, αντικείμενα και μηνύματα που ανταλλάσσουν τα αντικείμενα και είναι εκμεταλλεύσιμο, για να περιγράψει τη ροή του ελέγχου μέσα στο εκάστοτε σύστημα.
- D. Διάγραμμα δομής υλοποίησης (composite structure diagram), που έχει δύο κατηγορίες, οι οποίες αναπτύσσονται πιο κάτω:
 - i. Διάγραμμα στοιχείων (component diagram): Περιγράφει τη φυσική διάσταση ενός συστήματος με όρους εκτελέσιμων αρχείων, βιβλιοθηκών κώδικα, HTML Ιστοσελίδων, εγγράφων και πινάκων βάσεων δεδομένων. Περικλείει τόσο στοιχεία (components) όσο και διεπαφές (interfaces) και διατίθεται, με σκοπό τη μοντελοποίηση του εκάστοτε πηγαίο κώδικα, την παραγωγή των εκτελέσιμων προγραμμάτων, τη δομή μίας βάσης δεδομένων και τη σχέση του εκάστοτε τρέχοντος συστήματος με άλλα εξωτερικά συστήματα.
 - ii. Διάγραμμα ανάπτυξης (deployment diagram): Αποδίδει τον τρόπο με τον οποίο διατάσσεται το λογισμικό (software) και το υλικό (hardware) ενός συστήματος για την τελική λειτουργία αυτού του συστήματος. Περιέχει κόμβους (nodes) και συσχετίσεις και προσφέρεται για την αρχιτεκτονική περιγραφή σύνθετων συστημάτων πελάτη-εξυπηρετητή client-server systems), συστημάτων πραγματικού χρόνου (real-time systems) και κατανεμημένων συστημάτων (distributed systems).

[28] [29] [30]

2.8.3 Σύγκριση της UML

Θα πρέπει να γίνει σαφές πως η γλώσσα UML δεν είναι ριζικά διαφορετική από τις γλώσσες μοντελοποίησης Booch, OMT (Object-Modeling Technique) και OOSE (Object-Oriented Software Engineering, αλλά μία διάδοχος κατάστασή τους. Αυτό σημαίνει πως ένας χρήστης μίας από τις πιο πάνω μεθοδολογίες θα μπορεί να χρησιμοποιεί τις υπάρχουσες γνώσεις του στη UML, καθώς αποτελεί ένα εξελικτικό βήμα τους. Οι χρήστες άλλων μεθοδολογιών θα δύνανται, επίσης, να μάθουν εύκολα UML, αλλά θα πρέπει να επιλέγουν μόνοι τους, αν θα χρησιμοποιούν τις έννοιες και τους συμβολισμούς της UML κάτω από τις δικές τους μεθοδολογίες.

Η μεθοδολογία της γλώσσας UML είναι πιο εκφραστική, αλλά και πιο ξεκάθαρη και ενοποιημένη από συναφείς μεθοδολογίες. Αυτό σημαίνει ότι, όταν μεταφέρεται κάποιος στη UML έχει όφελος, γιατί μπορεί να μοντελοποιήσει έργα που δεν μπορούσε πριν. Επίσης, υπάρχει κέρδος, γιατί αφαιρούνται οι άχρηστες διαφορές στο συμβολισμό και την ορολογία που δεν αφήνουν να φανούν οι ομοιότητες των προσεγγίσεων αυτών.

Σε σχέση με άλλες οπτικές γλώσσες μοντελοποίησης, συμπεριλαμβανομένων της μοντελοποίησης οντοτήτων-συσχετίσεων, της BRP (Business Reengineering Process), των διαγραμμάτων ροής και των γλωσσών που προσανατολίζονται στις καταστάσεις, το UML μοντέλο προσφέρει επιπλέον εκφραστικότητα και ολιστική ακεραιότητα.

Οι χρήστες των υπάρχουσών μεθοδολογιών θα αντιμετωπίσουν μικροαλλαγές στους συμβολισμούς, αλλά αυτό δε σημαίνει τόσο ότι πρέπει να μάθουν εκ νέο κάποια πράγματα όσο να ξεκαθαρίσουν κάποιες έννοιες στην υποκείμενη σημασιολογία. Αν έχουν επιτευχθεί οι στόχοι της ενοποίησης, η

γλώσσα UML αποτελεί την προφανή επιλογή, για να ξεκινήσει κάποιος ένα νέο έργο, ειδικά όσο υπάρχουν όλο και περισσότερα εργαλεία, βιβλία και παροχή κατάρτισης. Πολλά από τα εργαλεία οπτικής μοντελοποίησης υποστηρίζουν υπάρχοντες συμβολισμούς ως όψεις του υποκείμενου μοντέλου, όπως είναι, για παράδειγμα, οι γλώσσες Booch, OMT, OOSE. Όταν τα εργαλεία αυτά προσφέρουν και την υποστήριξη στη UML, οι χρήστες θα είναι ικανοί να μεταφέρουν τα υπάρχοντα μοντέλα τους στο συμβολισμό της UML, χωρίς να χάνουν καθόλου πληροφορία.

[29] [30]

2.9 Το Μοντέλο Αντικειμένου Εγγράφου

2.9.1 Εισαγωγή στο Μοντέλο Αντικειμένου Εγγράφου

Το Μοντέλο Αντικειμένου Εγγράφου (Document Object Model-DOM) είναι μία καλοσχεδιασμένη Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface-API) προορισμένη κυρίως για έγγραφα της γλώσσας HTML και της γλώσσας XML. Το DOM καθορίζει τη λογική δομή τέτοιου είδους εγγράφων και τον τρόπο με τον οποίο αυτά γίνονται προσβάσιμα και διαχειρίσιμα.

Το βασικό πλεονέκτημα του DOM έγκειται στο ότι δίνει τη δυνατότητα στους εκάστοτε προγραμματιστές όχι απλά να δημιουργήσουν και να τεκμηριώσουν διάφορα HTML και XML έγγραφα, αλλά κυρίως να επέμβουν στη δομή αυτών των εγγράφων και να προσθέσουν, να τροποποιήσουν ή να διαγράψουν στοιχεία και περιεχόμενο τους. Έτσι, οτιδήποτε περιλαμβάνεται σε ένα HTML ή XML έγγραφο μπορεί να προσπελαστεί, να αλλάξει ή να σβηστεί με χρήση του DOM.

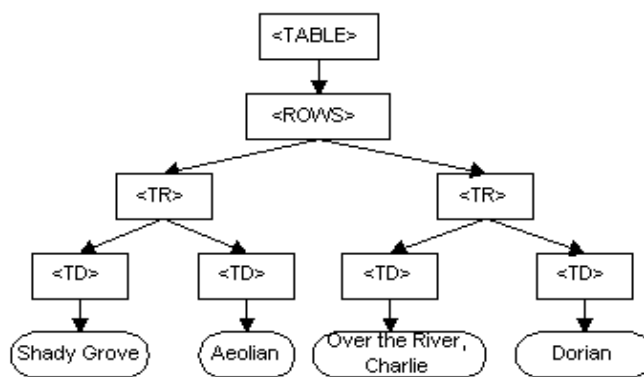
[31]

2.9.2 Η Αναπαράσταση του Μοντέλου Αντικειμένου Εγγράφου

Το DOM στην ουσία ταυτίζεται με τη δομή των εγγράφων που μοντελοποιεί. Στη συνέχεια παρουσιάζεται, για παράδειγμα, το απόσπασμα κώδικα ενός HTML εγγράφου για την εμφάνιση ενός πίνακα σε μία Ιστοσελίδα:

```
<TABLE>
  <ROWS>
    <TR>
      <TD>Shady Grove</TD>
      <TD>Aeolian</TD>
    </TR>
    <TR>
      <TD>Over the River, Charlie</TD>
      <TD>Dorian</TD>
    </TR>
  </ROWS>
</TABLE>
```

Το DOM που αναπαριστά τον πίνακα του παραπάνω παραδείγματος φαίνεται στην παρακάτω Εικόνα:



Εικόνα16: Παράδειγμα αναπαράστασης ενός HTML εγγράφου μέσω του DOM.

Είναι προφανές, επομένως, ότι στο DOM τα έγγραφα έχουν μία λογική δομή, η οποία μοιάζει πολύ με ένα δένδρο και μάλιστα ακόμα περισσότερο με ένα δάσος, που μπορεί να περιέχει πάνω από ένα δένδρα. Ωστόσο, στο σημείο αυτό, πρέπει να γίνει ξεκάθαρο πως το DOM δεν προδιαγράφει πως τα εκάστοτε HTML ή XML έγγραφα θα υλοποιηθούν με μία μορφή δένδρου ή δάσους και δεν καθορίζει σε καμία περίπτωση τις σχέσεις που συνδέουν μεταξύ τους τα οποιαδήποτε αντικείμενα των εγγράφων αυτών. Με άλλα λόγια, το DOM προσδιορίζει εξολοκλήρου το λογικό μοντέλο μίας διεπαφής προγραμματισμού, το οποίο με τη σειρά του θα αναπτυχθεί με τον κατάλληλο τρόπο, που κρίνει απαραίτητο η υφιστάμενη κάθε φορά εφαρμογή.

Επιπρόσθετα, ένα ιδιαίτερο χαρακτηριστικό του DOM είναι ο διαρθρωτικός ισομορφισμός σύμφωνα με τον οποίο, αν δύο ξεχωριστές υλοποιήσεις του DOM χρησιμοποιούνται, για να μοντελοποιήσουν τη δομή ενός εγγράφου, τότε θα της αποδώσουν όμοια αναπαράσταση, σε κάθε περίπτωση με ακριβώς τα ίδια αντικείμενα και τις ίδιες συνδέσεις μεταξύ αυτών.

[31]

2.9.3 Οι Διεπαφές του Μοντέλου Αντικειμένου Εγγράφου

Το όνομα Μοντέλο Αντικειμένου επιλέχθηκε, εφόσον είναι ένα μοντέλο αντικειμένου που χρησιμοποιεί η παραδοσιακή λογική σχεδιασμού με βάση τα αντικείμενα: τα εκάστοτε έγγραφα μοντελοποιούνται με χρήση αντικειμένων και το μοντέλο αυτό περιλαμβάνει όχι μόνο τη δομή ενός εγγράφου, αλλά και τη συμπεριφορά του, όπως, επιπλέον, και τα αντικείμενα που το απαρτίζουν. Με άλλα λόγια οι κόμβοι στο παραπάνω διάγραμμα δεν αντιπροσωπεύουν απλά μία δομή δεδομένων, αλλά αντιπροσωπεύουν, επιπρόσθετα, αντικείμενα που έχουν λειτουργίες και ταυτότητα. Έτσι, ως ένα μοντέλο αντικειμένου, το DOM προσδιορίζει τα εξής στοιχεία:

- Τις διεπαφές και τα αντικείμενα που χρησιμοποιούνται, ώστε να αναπαραστήσουν και να χειριστούν ένα έγγραφο.
- Τη σημασιολογία αυτών των διεπαφών και αντικειμένων, συμπεριλαμβανομένων τόσο της συμπεριφοράς των αντικειμένων όσο και των χαρακτηριστικών γνωρισμάτων τους.
- Τις σχέσεις και τις συνεργασίες μεταξύ των συγκεκριμένων διεπαφών και αντικειμένων.

Η δομή ενός εγγράφου της γλώσσας SGML (Standard Generalized Markup Language) (απόδοση στα Ελληνικά: Πρότυπης Γενικευμένης Γλώσσας Σήμανσης) παραδοσιακά αναπαριστανόταν από ένα αφηρημένο μοντέλο δεδομένων. Σε ένα τέτοιο αφηρημένο μοντέλο δεδομένων, το μοντέλο είναι στο κέντρο και τα δεδομένα γύρω του. Στις γλώσσες προγραμματισμού με βάση τα αντικείμενα, τα δεδομένα ενθυλακώνονται στα αντικείμενα και κρύβονται μέσα τους, προκειμένου να μένουν προστατευμένα από απευθείας εξωτερικούς χειρισμούς. Οι λειτουργίες, λοιπόν, που σχετίζονται με τα συγκεκριμένα αντικείμενα και καθορίζουν το χειρισμό αυτών των αντικειμένων είναι μέρος του Μοντέλου Αντικειμένου.

Το DOM ορίζει τις διεπαφές που χρησιμοποιούνται, για να διαχειριστούν HTML ή XML έγγραφα. Είναι σημαντικό να αντιληφθεί κάποιος ότι οι διεπαφές αυτές είναι ένα μέσον καθορισμού ενός τρόπου πρόσβασης και χειρισμού της εσωτερικής αναπαράστασης τέτοιων εγγράφων για τις εκάστοτε

εφαρμογές. Πάνω από όλα, οι συγκεκριμένες διεπαφές δε συνεπάγονται μία ειδική συμπαγής υλοποίηση. Μάλιστα, κάθε εφαρμογή του DOM είναι ελεύθερη να υποστηρίξει έγγραφα σε οποιαδήποτε κατάλληλη αναπαράσταση, όσο οι προδιαγεγραμμένες διεπαφές υποστηρίζονται. Ειδικότερα:

- Τα χαρακτηριστικά που ορίζονται στη γλώσσα OMG IDL (Object Management Group Interface Definition Language) (απόδοση στα Ελληνικά Γλώσσα Καθορισμού Διεπαφής της Ομάδας Διαχείρισης Αντικειμένων) δεν υποδηλώνουν ότι συγκεκριμένα αντικείμενα πρέπει να έχουν μοναδικά χαρακτηριστικά σύμφωνα με τις δεσμεύσεις της εκάστοτε γλώσσας προγραμματισμού.
- Οι εφαρμογές του DOM πρέπει να παρέχουν πρόσθετες διεπαφές και αντικείμενα, τα οποία, αν και δεν προσδιορίζονται επακριβώς από τη γλώσσα OMG IDL, ωστόσο να θεωρούνται συμβατές με το ίδιο το μοντέλο.

Σε γενικές γραμμές, το DOM δε γνωρίζει ποιους κατασκευαστές να χρησιμοποιήσει, ώστε να υλοποιήσει μία εφαρμογή, διότι σε κάθε περίπτωση προδιαγράφονται οι διεπαφές και όχι τα αντικείμενα που πρόκειται να δημιουργηθούν. Οπότε, οι χρήστες του DOM καλούν κάθε φορά την κατάλληλη μέθοδο κατασκευής της κλάσης ενός εγγράφου, με στόχο να δημιουργήσουν τη δομή αυτού του εγγράφου και οι εφαρμογές του DOM με τη σειρά τους φτιάχνουν τις δικές τους εσωτερικές αναπαραστάσεις αυτών των δομών.

[31]

2.10 Η Γλώσσα Σήμανσης Περιεχομένου Ιστοσελίδων, HTML

2.10.1 Εισαγωγή στην HTML

Η γλώσσα HTML (HyperText Markup Language) (απόδοση στα Ελληνικά: Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης των Ιστοσελίδων. Τα στοιχεία της HTML είναι τα βασικά δομικά στοιχεία των Σελίδων του Παγκόσμιου Ιστού.

Ο σκοπός ενός περιηγητή του Παγκοσμίου Ιστού είναι να διαβάσει τα HTML έγγραφα και τα συνθέσει σε Ιστοσελίδες, που μπορεί κανείς να δει ή να ακούσει. Ο εκάστοτε πλοηγός δεν εμφανίζει τις HTML ετικέτες, αλλά τις χρησιμοποιεί, για να ερμηνεύσει το περιεχόμενο της υφιστάμενης κάθε φορά Σελίδας.

Τα στοιχεία (elements) της γλώσσας HTML χρησιμοποιούνται, προκειμένου να κατασκευαστεί η επιθυμητή κάθε φορά Ιστοσελίδα. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα σε μία Ιστοσελίδα και μπορεί να χρησιμοποιηθεί, ώστε να εμφανίσει διαδραστικές διαδικτυακές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων, δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης αυτού του περιεχομένου, καθορίζοντας σημαντικά δομικά στοιχεία για τα τυχόν κείμενα που απαντώνται στο συγκεκριμένο περιεχόμενο, όπως είναι, για παράδειγμα, οι κεφαλίδες, οι παράγραφοι, οι λίστες, οι σύνδεσμοι. Είναι σε θέση, επίσης, να ενσωματώνει εντολές γλωσσών σεναρίων, όπως είναι, για παράδειγμα, η γλώσσα JavaScript, και να επηρεάζει με αυτό τον τρόπο τη συμπεριφορά των Ιστοσελίδων, προσδίδοντας τους ένα δυναμικό χαρακτήρα.

[31]

2.10.2 Τα Βασικά Συστατικά της HTML

Τα πιο βασικά στοιχεία που συνθέτουν τη γλώσσα HTML είναι τα εξής:

- Οι ετικέτες (tags): Η γλώσσα HTML καθορίζει τη μορφή με την οποία ο εκάστοτε πλοηγός απεικονίζει τα διάφορα έγγραφα. Με τον όρο έγγραφα εννοούμε κάθε είδους αρχείου, όπως είναι, για παράδειγμα, τα αρχεία κειμένου, οι εικόνες, τα κινούμενα γραφικά και τα βίντεο. Τα αρχεία της HTML είναι απλά αρχεία κειμένου στα οποία υπάρχουν ειδικοί κώδικες που ονομάζονται ετικέτες (tags) και δηλώνουν στον εκάστοτε περιηγητή τον τρόπο με τον οποίο αυτά θα εμφανιστούν. Οι ετικέτες έχουν την παρακάτω γενική δομή: <όνομα ετικέτας>κείμενο</όνομα ετικέτας>. Η ετικέτα <όνομα ετικέτας>, που εκμεταλλεύεται σε κάθε περίπτωση, δηλώνει στον εκάστοτε φυλλομετρητή μία ενέργεια που θα εκτελεστεί στο κείμενο που περικλείεται εντός

αυτής της ετικέτας. Για να δηλωθεί, τώρα, στον εκάστοτε φυλλομετρητή το σημείο ενός κειμένου που θα σταματήσει να δρα μία ετικέτα, διατίθεται τις περισσότερες φορές για κάθε ετικέτα <όνομα ετικέτας> η αντίστοιχη ετικέτα κλεισίματος </όνομα ετικέτας>. Υπάρχουν, ωστόσο, περιπτώσεις HTML ετικετών που δεν έχουν ετικέτα κλεισίματος, όπως είναι, για παράδειγμα, η ετικέτα εισαγωγής μίας εικόνας (η οποία θα αναπτυχθεί παρακάτω). Αξίζει να τονιστεί σε αυτό το σημείο ότι οι ετικέτες στην HTML είναι δυνατό να είναι ένθετες, δηλαδή να βρίσκονται μέσα σε άλλες ετικέτες, όπως είναι, για παράδειγμα η ετικέτα <body>...</body> εντός της <html>...</html> ή η ετικέτα <title>...</title> εντός της <head>...</head>. Ιδιαίτερη προσοχή πρέπει να δοθεί στη σειρά των ετικετών τερματισμού, η οποία πρέπει να είναι η αντίστροφη της σειράς δήλωσης των αντίστοιχων ετικετών, όπως φαίνεται, για παράδειγμα, στο επόμενο απόσπασμα κώδικα:

```
<head>
<title>
.....
</title>
</head>
```

Τέλος, αξίζει να αναφερθεί πως στο όνομα μίας ετικέτας δε γίνεται διάκριση ανάμεσα σε πεζούς και κεφαλαίους χαρακτήρες. Οποιοσδήποτε συνδυασμός είναι αποδεκτός. Έτσι, για παράδειγμα, οι ετικέτες <html>, <HTML>, <HtMl> είναι ισοδύναμες.

- Η δομή ενός HTML εγγράφου (HTML document structure): Ένα HTML αρχείο περιέχεται μέσα στην ετικέτα <html>...</html> και αποτελείται από δύο ξεχωριστά μέρη: το <head>...</head>, στο οποίο περιέχονται πληροφορίες που αφορούν το ίδιο το αρχείο και όχι το τι θα εμφανιστεί στην οθόνη του σε κάθε περίπτωση υφιστάμενου χρήστη και το <body>...</body>, στο οποίο περιέχεται πως θα παρουσιαστεί στην εκάστοτε Ιστοσελίδα. Τη δομή ενός HTML εγγράφου κλείνει η ετικέτα <title>...</title> που τοποθετείται εντός της ετικέτας <head>...</head>. Η ετικέτα <title>...</title> εμφανίζει τον τίτλο στην μπάρα τίτλου του εκάστοτε πλοηγού, δίχως να δείχνει κάποιο κείμενο στην οθόνη του εκάστοτε χρήστη.
- Οι επικεφαλίδες (headings): Οι επικεφαλίδες χρησιμοποιούνται για την μορφοποίηση του κειμένου που εμφανίζεται σε μία Ιστοσελίδα. Για να οριστεί μία επικεφαλίδα, διατίθεται η ετικέτα <hN>...</hN>, όπου το N είναι ένας ακέραιος αριθμός από το 1 ως και το 6. Όσο αυξάνεται αυτό το N τόσο λιγότερο σημαντική γίνεται μία επικεφαλίδα. Έτσι, η επικεφαλίδα <h1>...</h1> είναι η πιο σημαντική, ενώ η επικεφαλίδα <h6>...</h6> η λιγότερο σημαντική.
- Οι διαχωριστικές ετικέτες (breaking tags): Η γλώσσα HTML γενικά αγνοεί τα κενά διαστήματα ή και τις αλλαγές γραμμής. Για τη δημιουργία τμημάτων κειμένου τα οποία θα είναι αυτοδύναμα, γίνεται χρήση διαχωριστικών. Ένα διαχωριστικό ενός κειμένου είναι οι παράγραφοι. Οι παράγραφοι διαχωρίζουν το εκάστοτε κείμενο μέσω της εμφάνισης μίας κενής γραμμής και δηλώνονται με τη χρήση της ετικέτας <p>...</p>. Στις επικεφαλίδες δεν είναι απαραίτητη η χρήση της ετικέτας παραγράφου, αφού η επικεφαλίδα είναι από μόνη της ένα ξεχωριστό κομμάτι κειμένου. Τέλος, αν και η ετικέτα τερματισμού </p> είναι προαιρετική, καλό είναι να χρησιμοποιείται, διότι σε κάποιες περιπτώσεις είναι υποχρεωτική, όπως είναι, για παράδειγμα, εφόσον γίνεται χρήση κάποιας μεθόδου στοίχισης κειμένου. Ένας άλλος τρόπος διαχωρισμού ενός κειμένου είναι και οι οριζόντιες γραμμές (horizontal rules). Οι οριζόντιες γραμμές δηλώνονται μέσω της ετικέτας <hr>. Η ετικέτα αυτή δεν έχει αντίστοιχη ετικέτα τερματισμού. Παρόμοια λειτουργία με την παράγραφο έχει και η ετικέτα αλλαγής γραμμής
, στην οποία, σε αντίθεση με την παράγραφο, δεν αφήνεται κενή γραμμή.
- Οι ετικέτες μορφοποίησης (format tags): Εκτός από τις παραγράφους και τις επικεφαλίδες, αφότου είναι επιθυμητή η μορφοποίηση του κειμένου μίας Ιστοσελίδας, διατίθενται οι ετικέτες μορφοποίησης της γλώσσας HTML. Οι πιο βασικές είναι η ..., που κάνει το εκάστοτε κείμενο να εμφανίζεται με έντονη γραφή, η <i>...</i>, η οποία κάνει το εκάστοτε κείμενο να εμφανίζεται με πλάγια γραφή και η <u>...</u>, που κάνει το εκάστοτε κείμενο να εμφανίζεται υπογραμμισμένο.

- Οι ετικέτες φράσεων (phrase tags): Οι ετικέτες φράσεων επιτελούν όμοια λειτουργία με αυτή των ετικετών μορφοποίησης και έχουν να κάνουν με τον τρόπο εμφάνισης ενός κειμένου σε μία Ιστοσελίδα. Οι πιο κύριες ετικέτες μορφοποίησης είναι η `...`, που επισημαίνει ένα κείμενο ως σημαντικό, η `...`, η οποία δίνει έμφαση σε ένα κείμενο και η `<abr>...</abr>`, που δημιουργεί μία συντομογραφία για ένα στοιχείο ενός κειμένου. Να σημειωθεί, εδώ, ότι αν τοποθετηθεί για λίγο χρονικό διάστημα ο κέρσορας πάνω σε αυτή τη συντομογραφία, τότε θα εμφανιστεί η πλήρη περιγραφή της.
- Οι λίστες (lists): Η γλώσσα HTML παρέχει διάφορες μορφές λιστών για την κατάταξη των αντικειμένων μίας Ιστοσελίδας. Μία μη αριθμημένη λίστα (unordered list) δηλώνεται με την ετικέτα `...`, ενώ κάθε στοιχείο της πρέπει να περικλείεται μέσα σε μία ετικέτα `...`. Κάθε στοιχείο αυτής της λίστας αναπαρίσταται με μία βούλα (ή με κάποιο άλλο κοινό σύμβολο) μπροστά του. Στις αριθμημένες λίστες (ordered lists) υπάρχει ένας αριθμός μπροστά από κάθε στοιχείο αυτού του είδους της λίστας. Η μόνη διαφορά στη δημιουργία των αριθμημένων λιστών, σε σχέση με τις μη αριθμημένες, είναι ότι χρησιμοποιείται η ετικέτα `...`, αντί της `...`. Σε αυτό το σημείο θα ήταν σκόπιμο να παρατηρηθεί ότι οι λίστες μπορούν να είναι ένθετες, δηλαδή μία λίστα είναι ικανή να περιέχεται εντός κάποιας άλλης. Σε περίπτωση που υπάρχουν ένθετες λίστες του ίδιου τύπου, το διακριτικό μπροστά από κάθε λίστα αλλάζει, ώστε να ξεχωρίζουν τα στοιχεία της κάθε λίστας. Πέρα από τις αριθμημένες και τις μη αριθμημένες λίστες, η HTML έχει και μία τρίτη κατηγορία λιστών, τις λίστες ορισμών (definition lists), οι οποίες επιτρέπουν την δημιουργία επεξηγήσεων για διάφορους ορισμούς. Στις λίστες αυτές δεν εμφανίζεται κάποιο διακριτικό σημάδι, αλλά, αντίθετα, η επεξήγηση του κάθε ορισμού παρουσιάζεται πιο μέσα από το υπόλοιπο κείμενο. Μία λίστα ορισμών δηλώνεται με την ετικέτα `<dl>...</dl>` και περιέχει ζεύγη ορισμού τίτλων μέσω της ετικέτας `<dt>...</dt>` καθώς και αντίστοιχης επεξήγησης μέσω της ετικέτας `<dd>...</dd>`.
- Οι εικόνες (images): Για την ενσωμάτωση μίας εικόνας σε μία Ιστοσελίδα χρησιμοποιείται η ετικέτα ``. Στην ετικέτα αυτή καθορίζεται το όνομα του αρχείου της εικόνας που σε κάθε περίπτωση είναι επιθυμητή η ενσωμάτωσή της σε μία Σελίδα. Στη γλώσσα HTML κάθε ετικέτα μπορεί να έχει διάφορα χαρακτηριστικά (γνωρίσματα)\ιδιότητες (attributes). Τα χαρακτηριστικά αυτά περιγράφουν με μεγαλύτερη ακρίβεια τον τρόπο με τον οποίο θα εμφανιστεί ένα συγκεκριμένο τμήμα μίας Σελίδας στον εκάστοτε φυλλομετρητή. Το χαρακτηριστικό `src` της ετικέτας ``, που μόλις προαναφέρθηκε, είναι ένα παράδειγμα για τον τρόπο δήλωσης των χαρακτηριστικών γνωρισμάτων που εκμεταλλεύεται η HTML: τα χαρακτηριστικά γνωρίσματα κάθε ετικέτας πρέπει να βρίσκονται εντός των συμβόλων `<` και `>` της αρχικής δήλωσης αυτής της ετικέτας (και όχι της ετικέτας τερματισμού, αν αυτή χρειάζεται), να διαχωρίζονται από το όνομα της εκάστοτε ετικέτας και από τα άλλα χαρακτηριστικά, που τυχόν υπάρχουν στην ίδια κάθε φορά ετικέτα και να είναι της μορφής: `όνομα χαρακτηριστικού=τιμή χαρακτηριστικού`. Μία από τις σημαντικότερες ιδιότητες της ετικέτας ``, εκτός της `src`, είναι η `alt`. Με τη βοήθεια της ιδιότητας αυτής καθορίζεται το εκάστοτε κείμενο που θα εμφανιστεί στη θέση μίας εικόνας σε πλοηγούς που δεν υποστηρίζουν γραφικά. Επιπλέον, δύο άλλες ιδιότητες της ετικέτας `` που χρησιμοποιούνται συχνά είναι οι `height` και `width`, με τις οποίες καθορίζεται, αντίστοιχα, το ύψος και το πλάτος της εκάστοτε εμφανιζόμενης εικόνας. Τέλος, θα ήταν ωφέλιμο να τονιστεί πως η HTML παρέχει και την ευχέρεια διαφορετικά τμήματα μίας εικόνας να αντιστοιχούν σε διαφορετικές ενέργειες, όπως, για παράδειγμα, σε διαφορετικούς δεσμούς υπερκειμένου (οι οποίοι θα αναπτυχθούν αμέσως παρακάτω). Τέτοιες εικόνες ονομάζονται χάρτες γραφικών (image maps).
- Τα σχόλια (comments): Ένα σχόλιο είναι ένα κείμενο μέσα σε μία Ιστοσελίδα το οποίο για τον περιηγητή που διατίθεται σε κάθε περίπτωση είναι σαν να μην υπάρχει, αλλά χρησιμοποιείται από τον εκάστοτε δημιουργό της συγκεκριμένης Σελίδας ως διευκρίνιση για αυτόν που διαβάσει τον κώδικά της. Ένα σχόλιο περιέχεται μέσα στα σύμβολα `<!--` και `-->` και μπορεί να καταλαμβάνει πάνω από μία γραμμές.

[32] [33] [34]

2.10.3 Οι Δεσμοί Υπερκειμένου της HTML

Στην παραπάνω ενότητα έγινε μία αναφορά στα πιο βασικά στοιχεία της γλώσσας HTML για τη μορφοποίηση σε κάθε περίπτωση του επιθυμητού υπό εμφάνιση κειμένου μίας Ιστοσελίδα καθώς και την ενσωμάτωση εικόνων σε αυτή τη Σελίδα. Όμως, η πραγματική δύναμη της HTML είναι η δυνατότητα δημιουργίας δεσμών υπερκειμένου (hypertext links) προς άλλα έγγραφα, τα οποία μπορεί να είναι κάθε μορφής πολυμεσική πληροφορία ή ακόμα και άλλες Σελίδες. Ο εκάστοτε χρήστης, με ένα απλό κλικ του ποντικιού σε ένα σημείο μίας Ιστοσελίδας, μπορεί να μεταβεί σε οποιαδήποτε σημείο είτε αυτής της Σελίδας είτε άλλης τόσο του Ιστοτόπου που οι προαναφερόμενες Ιστοσελίδες ανήκουν όσο και διαφορετικού Ιστοχώρου. Αφότου στο Διαδίκτυο γίνεται χρήση πλοηγών με γραφικό περιβάλλον, οι δεσμοί υπερκειμένου συνήθως, εμφανίζονται υπογραμμισμένοι με μπλε χρώμα, όπως είναι, για παράδειγμα, ο δεσμός <http://www.epp.teicrete.gr/>. Σε περίπτωση που έχει πραγματοποιηθεί ήδη επίσκεψη ενός δεσμού, τότε το χρώμα του αλλάζει και τις πιο πολλές φορές γίνεται μωβ.

Οι δεσμοί υπερκειμένου ταξινομούνται στις ακόλουθες τρεις κατηγορίες:

- I. Δεσμοί υπερκειμένου σε τοπικά αρχεία: Νοοούνται οι δεσμοί προς αρχεία που βρίσκονται στο ίδιο μηχάνημα με αυτό της εκάστοτε Ιστοσελίδας που περιέχει αυτούς τους δεσμούς. Ο απλούστερος δεσμός σε τοπικά αρχεία που δύναται να υπάρχει είναι εκείνος ο δεσμός προς ένα άλλο αρχείο HTML, που βρίσκεται στον ίδιο υποκατάλογο με το εκάστοτε έγγραφο που συναντάται ο συγκεκριμένος δεσμός. Για έναν δεσμό, όπως ο προαναφερόμενος, χρησιμοποιείται η ετικέτα `κείμενο δεσμού`. Στο κείμενο ενός δεσμού μπορούν να εφαρμοστούν ετικέτες μορφοποίησης καθώς και εικόνες. Για να υλοποιηθούν δεσμοί προς αρχεία σε άλλους υποκαταλόγους, πρέπει, πέραν από το όνομα του εκάστοτε αρχείου, να δηλωθεί και ο τρέχον κάθε φορά κατάλογος στον οποίο αυτό βρίσκεται. Αν, λοιπόν, βρίσκεται σε υποκατάλογο που περιέχεται στον κατάλογο της εκάστοτε παρούσας Σελίδας, τότε αρκεί να καθοριστεί το όνομα αυτού του υποκαταλόγου, να χρησιμοποιηθεί το σύμβολο / και στη συνέχεια να γραφεί το όνομα του αρχείου. Σε περίπτωση που είναι θεμιτό να εκμεταλλευτεί δεσμός προς Σελίδα ανωτέρου επιπέδου, είναι ωφέλιμο να χρησιμοποιηθεί η ακολουθία χαρακτήρων ../ για κάθε ένα επίπεδο της ιεραρχίας των υφιστάμενων σε κάθε φάση καταλόγων. Τα σύμβολα / και ../ μπορούν να εφαρμοστούν περισσότερες από μία φορές σε μία ζεύξη και είναι δυνατό να εμφανίζονται με οποιαδήποτε σειρά.
- II. Δεσμοί υπερκειμένου σε Ιστοτόπους: Για τη δημιουργία ενός δεσμού σε Ιστοτόπους χρησιμοποιείται η ετικέτα `<a>`, όπως και στην περίπτωση των δεσμών σε τοπικά αρχεία. Στην περίπτωση αυτή, όμως, στη θέση του εκάστοτε ονόματος αρχείου χρησιμοποιείται η URL του εκάστοτε Τόπου σύμφωνα με το σχήμα: `κείμενο δεσμού`.
- III. Δεσμοί υπερκειμένου σε τμήματα της εκάστοτε ίδιας Σελίδας: Πολλές φορές είναι επιθυμητό να δημιουργηθεί ένας Ιστότοπος ο οποίος να περιέχει μεγάλη ποσότητα κειμένου. Ένας τρόπος οργάνωσής ενός τέτοιου Τόπου είναι ο χωρισμός του σε επιμέρους Ιστοσελίδες και η δημιουργία ενός πίνακα περιεχομένων. Εκτός από αυτόν τον τρόπο οργάνωσης, η γλώσσα HTML παρέχει τη δυνατότητα δημιουργίας δεσμών υπερκειμένου προς κάποιο τμήμα του σε κάθε περίπτωση ίδιου Ιστοχώρου. Εξαιτίας της δυνατότητας αυτής, δύναται να οργανωθεί ένας Ιστότοπος διαφορετικά: στην αρχή είναι εφικτή η τοποθέτηση ενός πίνακα περιεχομένων, που θα χρησιμοποιεί δεσμούς προς άλλα τμήματα του συγκεκριμένου Τόπου, και στη συνέχεια να γραφεί το κατάλληλο σε κάθε φάση κείμενο. Αυτό προσφέρει στον εκάστοτε χρήστη του ενός τέτοιου Ιστοχώρου μεγαλύτερη ευκολία ανάγνωσης ολόκληρου του περιεχομένου του, αλλά και τη δυνατότητα αποθήκευσής του, χωρίς την ανάγκη αποθήκευσης πολλών διαφορετικών επιμέρους Σελίδων. Η HTML προσφέρει το παραπάνω πλεονέκτημα δίνοντας κρυφά διακριτικά ονόματα σε μέρη του κειμένου ενός εγγράφου και στη συνέχεια κάνοντας αναφορές (δεσμούς) στα συγκεκριμένα μέρη. Για να αποδοθεί, λοιπόν, ένα διακριτικό κρυφό όνομα, όπως είναι, για παράδειγμα, το "NAME", σε ένα κείμενο, διατίθεται η ετικέτα `κείμενο`. Για την αναφορά, τώρα, στο προαναφερόμενο κείμενο, με το κρυφό διακριτικό όνομα NAME, διατίθεται η ετικέτα `κείμενο δεσμού`. Τέλος, είναι ικανή η δημιουργία και δεσμών προς τμήματα άλλων Ιστοτόπων. Για το σκοπό αυτό παρέχεται πάλι η ετικέτα `<a>` με το χαρακτηριστικό href να έχει τη μορφή: URL#NAME. Στον παραπάνω συμβολισμό η URL μπορεί να αναφέρεται σε ένα αρχείο στο εκάστοτε τοπικό μηχάνημα ή σε ένα αρχείο στο Διαδίκτυο και το NAME αφορά το κρυφό συμβολικό όνομα μίας ενότητας ενός από τα προγενέστερα αρχεία. Σε αυτή την περίπτωση η ετικέτα έχει τη μορφή: `κείμενο δεσμού`.

2.10.4 Άλλες Δυνατότητες της HTML

Εκτός από τα κύρια συστατικά της γλώσσας HTML, που αναφέρθηκαν ακριβώς παραπάνω, η συγκεκριμένη γλώσσα έχει και άλλες δυνατότητες, οι σημαντικότερες από τις οποίες παρατίθενται στη συνέχεια:

- Η στοίχιση κειμένου (text align): Στη γλώσσα HTML παρέχεται η δυνατότητα στοίχισης του κειμένου μίας Ιστοσελίδας, με άλλα λόγια η δυνατότητα ένα κείμενο να εμφανίζεται στο κέντρο μίας γραμμής μίας Σελίδας ή να είναι στοιχισμένο μόνο αριστερά ή μόνο δεξιά ή αριστερά και δεξιά ταυτόχρονα (πλήρη στοίχιση). Για να επιτευχθεί κάτι τέτοιο καταχωρούνται οι τιμές center, left, right, justify, αντίστοιχα, στην ιδιότητα align σε μία σειρά από HTML ετικέτες, όπως είναι, για παράδειγμα η ετικέτα παραγράφου ή οι ετικέτες επικεφαλίδων. Ανάλογη διευθέτηση μπορεί να επιτευχθεί για τις εικόνες μίας Σελίδας, τοποθετώντας την ιδιότητα align στην ετικέτα . Στην προκειμένη περίπτωση το κατάλληλο HTML κείμενο, που συνοδεύει την εκάστοτε εικόνα γεμίζει τον περιβάλλοντα χώρο της εικόνας αυτής. Αντίθετα, αν δε γίνει χρήση της ιδιότητας align, το προαναφερόμενο κείμενο συνεχίζει σε καινούργια γραμμή, κάτω από την προαναφερόμενη εικόνα.
- Το χρώμα (color): Οι περισσότεροι από τους σύγχρονους πλοηγούς έχουν τη δυνατότητα απεικόνισης χρώματος τόσο για τα κείμενα όσο και για το εκάστοτε φόντο μίας Ιστοσελίδας. Για την αναπαράσταση των χρωμάτων εφαρμόζεται η RGB (Red-Green-Blue) κωδικοποίηση χρωμάτων. Στην κωδικοποίηση αυτή αποδίδονται οι τιμές από 0 ως και 255 στα τρία βασικά χρώματα (κόκκινο-πράσινο-μπλε) για την απεικόνιση της έντασης του κάθε χρώματος στην επιθυμητή σε κάθε περίπτωση τελική απόχρωση ενός αντικειμένου μίας Σελίδας. Αν χρησιμοποιηθεί η τιμή 0 σε ένα χρώμα, τότε το χρώμα αυτό δεν συμμετέχει στην προαναφερόμενη τελική απόχρωση, ενώ αν χρησιμοποιηθεί η τιμή 255, τότε το συγκεκριμένο χρώμα συμμετέχει στο μέγιστο βαθμό στην εν λόγω τελική απόχρωση. Όταν και τα τρία βασικά χρώματα έχουν την τιμή 0, τότε παράγεται το μαύρο, ενώ, εφόσον και τα τρία έχουν την τιμή 255, το λευκό. Στη γλώσσα HTML το κάθε χρώμα αναπαρίσταται από την ακολουθία #xxxxxx, όπου τα δύο πρώτα xx αντιστοιχούν στην τιμή του κόκκινου (στο δεκαεξαδικό σύστημα αρίθμησης). Αντίστοιχα, τα επόμενα ζεύγη xx αντιστοιχούν στο πράσινο και στο μπλε. Πλην της παραπάνω αναπαράστασης των χρωμάτων, η HTML περιέχει και συμβολικά ονόματα-κωδικούς για κάποια βασικά χρώματα (πχ το μαύρο αντιστοιχεί στον κωδικό #000000). Για να αλλαχθεί το φόντο μίας Σελίδας εκμεταλλεύεται το χαρακτηριστικό bgcolor της ετικέτας <body>...</body>. Στην ετικέτα <body>...</body> υπάρχουν και άλλες ιδιότητες για τον έλεγχο των χρωμάτων μίας Σελίδας, όπως είναι, για παράδειγμα, η link, για το χρώμα ενός απλού κειμένου, η ιδιότητα text, για το χρώμα των δεσμών υπερκειμένου, η vlink, για τους δεσμούς υπερκειμένου τους οποίους έχουν ήδη επισκεφθεί οι χρήστες. Όλες οι προγενέστερες ιδιότητες δέχονται ως τιμή το συμβολικό όνομα ή την ακολουθία #xxxxxx του κατάλληλου χρώματος του οποίου είναι επιθυμητή η χρήση σε κάθε περίπτωση. Στην περίπτωση που είναι χρήσιμη η αλλαγή των προκαθορισμένων χρωμάτων που χρησιμοποιεί ο εκάστοτε περιηγητής για την αναπαράσταση των διαφόρων στοιχείων μίας Σελίδας, ιδιαίτερη προσοχή πρέπει να δοθεί στην τιμή των χρωμάτων που θα επιλεγθούν. Μία λανθασμένη επιλογή χρωμάτων μπορεί να οδηγήσει τους χρήστες σε προβλήματα ανάγνωσης του περιεχομένου της συγκεκριμένης Σελίδας. Τέλος, υπάρχει η δυνατότητα να οριστεί ως φόντο μίας Ιστοσελίδας κάποια εικόνα. Στην περίπτωση αυτή γίνεται εκμετάλλευση του χαρακτηριστικού background της ετικέτας <body>...</body>. Η τιμή του χαρακτηριστικού αυτού δηλώνεται με τη μορφή: <body background="image">...</body>. Σε περίπτωση που η προαναφερόμενη εικόνα φόντου έχει μικρότερο μέγεθος από το παράθυρο του εκάστοτε πλοηγού, η εικόνα αυτή επαναλαμβάνεται τόσο κατά τον κατακόρυφο όσο και κατά τον οριζόντιο άξονα, ώστε να καλύψει ολόκληρη την επιφάνεια του παραπάνω παραθύρου. Ιδιαίτερη προσοχή πρέπει να δοθεί στο μέγεθος του αρχείου της επιθυμητής σε κάθε περίπτωση εικόνας φόντου μίας Ιστοσελίδας, προκειμένου να μην καθυστερεί υπερβολικά η εμφάνιση αυτής της Σελίδας στο φυλλομετρητή του εκάστοτε

χρήστη. Πολλές φορές μία λανθασμένη επιλογή φόντου μπορεί να προκαλέσει προβλήματα στην αναγνωσιμότητα μίας Ιστοσελίδας.

- Η γραμματοσειρά (font): Η HTML ετικέτα που διατίθεται για τεχνικές μορφοποίησης του εμφανιζόμενου κειμένου στην εκάστοτε Ιστοσελίδα είναι η `...`. Ανάλογα με τη μορφοποίηση που είναι αρεστή κάθε φορά να επιτευχθεί, διατίθενται μία σειρά από ιδιότητες της ετικέτας `...`. Χρησιμοποιώντας, λοιπόν, το χαρακτηριστικό `size` της ετικέτας `...`, πραγματοποιείται αλλαγή του μεγέθους του σε κάθε φάση προς εμφάνιση κειμένου μίας Σελίδας, σύμφωνα με τη μορφή: `...`, όπου η μεταβλητή `N` μπορεί να πάρει τις τιμές από 1 ως και 7, με την τιμή 3 να αντιστοιχεί στο κανονικό μέγεθος της εκάστοτε εκμεταλλεζόμενης γραμματοσειράς. Οι τιμές 1 και 2 αντιστοιχούν σε μικρότερα μεγέθη, ενώ οι τιμές 4, 5, 6 και 7 σε μεγαλύτερα μεγέθη. Επίσης, στο χαρακτηριστικό `size` της ετικέτας `...` υπάρχει η ευχέρεια να αποδοθεί το σχετικό, σε σχέση με το κανονικό, μέγεθος της εκάστοτε χρησιμοποιούμενης γραμματοσειράς. Έτσι, αν είναι επιθυμητό το κείμενο μίας Σελίδας να εμφανίζεται με μικρότερη γραμματοσειρά κατά `X` (όπου `X=1,2,3,...`) τάξεις μεγέθους, από το κανονικό της μέγεθος, τότε γίνεται χρήση της σύνταξης: `...`. Στην περίπτωση που χρειάζεται να εμφανίζεται το προηγούμενο κείμενο μεγαλύτερο κατά `Y` (όπου `Y=1,2,3`) τάξεις μεγέθους, αναφορικά με το κανονικό του μέγεθος, ακολουθείται η σύνταξη: `...`. Επιπλέον, για να εμφανιστεί κάποιο συγκεκριμένο χρώμα σε ένα κομμάτι κειμένου μίας Σελίδας, γίνεται χρήση της ιδιότητας `color` της ετικέτας `...`, κατόπιν του αποσπάσματος κώδικα: `...`, όπου `xxxxxx` ο κωδικός του επιθυμητού σε κάθε περίπτωση χρώματος. Για την παρουσίαση, τώρα, του κειμένου μίας Ιστοσελίδας, ο εκάστοτε πλοηγός χρησιμοποιεί κάποιες προκαθορισμένες γραμματοσειρές. Αν προτιμάται ένα τμήμα κειμένου μίας Ιστοσελίδας να παρουσιάζεται με κάποια συγκεκριμένη γραμματοσειρά, τότε πρέπει να γίνει εκμετάλλευση του χαρακτηριστικού `face` της ετικέτας `...`. Το χαρακτηριστικό αυτό δέχεται ως τιμή μία λίστα από ονόματα γραμματοσειρών. Ο κατάλληλος κάθε στιγμή φυλλομετρητής εξετάζει, αν είναι εγκατεστημένες στο σύστημα του εκάστοτε χρήστη οι γραμματοσειρές που δηλώνονται στο συγκεκριμένο χαρακτηριστικό, με την ίδια σειρά με την οποία δηλώνονται. Σε περίπτωση που στο πρωτότερο σύστημα βρεθεί κάποια από τις προαναφερόμενες γραμματοσειρές, τότε χρησιμοποιείται για την εμφάνιση εκείνου του κομματιού κειμένου που είναι επιθυμητό να έχει τη συγκεκριμένη γραμματοσειρά, διαφορετικά και για το κομμάτι αυτό χρησιμοποιείται η προκαθορισμένη γραμματοσειρά με την οποία εμφανίζεται όλη η υπόλοιπη Ιστοσελίδα. Συνεπώς πρέπει να δίνεται ιδιαίτερη προσοχή, ώστε να επιλέγονται γραμματοσειρές που είναι κοινές στους περισσότερους χρήστες. Στο σημείο τούτο, αξίζει να τονιστεί ότι, αν είναι ωφέλιμο να αλλαχθεί η προκαθορισμένη γραμματοσειρά ολόκληρου του κειμένου που εμφανίζεται σε μία Ιστοσελίδα, ως προς το μέγεθος, το χρώμα και τον τύπο, τότε είναι εφικτό μέσα στην ετικέτα `<head>...</head>` να ενταχθεί η ετικέτα `<basefont>` και να καταχωρηθούν στα χαρακτηριστικά της ετικέτας `<basefont>`, `size`, `color` και `face`, οι επιθυμητές τιμές μεγέθους, χρώματος και τύπου γραμματοσειράς, αντίστοιχα. Κλείνοντας, στη γλώσσα HTML μπορούν να γραφούν δείκτες και εκθέτες, που απαντώνται συχνά σε μαθηματικούς τύπους. Προκειμένου, λοιπόν, να δηλωθεί ένας εκθέτης (superscript) διατίθεται η ετικέτα `^{...}`, ενώ, για να δηλωθεί ένας δείκτης (subscript), η ετικέτα `_{...}`.
- Οι πίνακες (tables): Οι πίνακες στη γλώσσα HTML χρησιμοποιούνται για την εκπλήρωση δύο βασικών επιδιώξεων: Πρώτον, για τη συγκεντρωμένη οργάνωση άλλων στοιχείων της HTML και δεύτερον, για τη διαμόρφωση της εμφάνισης των Ιστοσελίδων. Γενικά στην HTML οι πίνακες αποτελούνται από γραμμές, οι οποίες με τη σειρά τους αποτελούνται από κελιά. Για να οριστούν τα στοιχεία ενός πίνακα ορίζεται, αρχικά, η πρώτη γραμμή, μετά η δεύτερη, η τρίτη και όλες οι υπόλοιπες μέχρι να οριστούν όλες οι γραμμές. Στον ορισμό της κάθε γραμμής ενός πίνακα περιέχεται ο ορισμός όλων των στοιχείων-κελιών που αυτή περιλαμβάνει. Για να δηλωθεί ένας πίνακας, εφαρμόζεται η ετικέτα `<table>...</table>`. Μέσα στην ετικέτα ενός πίνακα δηλώνονται οι ετικέτες της κάθε γραμμής του, `<tr>...</tr>`, ενώ μέσα στην ετικέτα της κάθε γραμμής του πίνακα ορίζονται οι ετικέτες των στοιχείων-κελιών της κάθε γραμμής του. Υπάρχουν δύο ετικέτες στοιχείων για ένα πίνακα: η `<td>...</td>` και η `<th>...</th>`. Στο εσωτερικό των παραπάνω ετικετών καθορίζονται τα δεδομένα κάθε στοιχείου αυτού του πίνακα. Η μόνη διαφορά μεταξύ των δύο προαναφερόμενων ετικετών είναι πως τα δεδομένα

της δεύτερης εμφανίζονται κεντραρισμένα και με έντονα γράμματα. Τα δεδομένα ενός κελιού ενός πίνακα είναι εφικτό να είναι οποιοδήποτε τμήμα κώδικα της HTML, όπως είναι, για παράδειγμα, μία εικόνα ή ένας δεσμός υπερκειμένου. Η HTML παρέχει τη δυνατότητα ορισμού του μεγέθους των περιγραμμάτων ενός πίνακα, χρησιμοποιώντας το χαρακτηριστικό border της ετικέτας `<table>...</table>`, για το συγκεκριμένο πίνακα. Ένα σημείο που πρέπει να τονιστεί είναι η ύπαρξη κενών κελιών σε έναν πίνακα, διότι όλοι οι πλοηγοί δε συμπεριφέρονται με τον ίδιο τρόπο σε μία τέτοια κατάσταση. Για τη σωστή αναπαράσταση των πινάκων που έχουν κενά κελιά, τα κελιά αυτά πρέπει να δηλώνονται με χρήση του ειδικού χαρακτήρα ` ` (non-breaking space). Επομένως, ο σωστός τρόπος δήλωσης των κενών κελιών ενός πίνακα συμφωνεί με το σχήμα: `<td> </td>`. Όπως αναφέρθηκε οι πίνακες είναι δυνατό να χρησιμοποιηθούν και για τη διαμόρφωση της εμφάνισης Σελίδων. Προκειμένου να αναγκαστούν τα περιεχόμενα μίας Σελίδας να στοιχιστούν με ένα νοητό πλαίσιο, δημιουργείται ένας πίνακας με μηδενικό πάχος περιγράμματος, δηλαδή ένας «αόρατος» πίνακας, όπως συνήθως αποκαλείται ένας τέτοιος πίνακας, κατά τη μορφή: `<table border=0>...</table>` και στη συνέχεια δηλώνονται όλα τα στοιχεία αυτού του πίνακα. Όλες οι παράμετροι-χαρακτηριστικά για την μορφοποίηση των πινάκων, που αναφέρθηκαν προηγουμένως, μπορούν να εκμεταλλευτούν και στην περίπτωση των «αόρατων» πινάκων. Για την στοίχιση του κειμένου σε ένα πίνακα χρησιμοποιείται η ιδιότητα `align`. Η ιδιότητα αυτή εφαρμόζεται και σε όλες τις ετικέτες που αναφέρθηκαν πιο πάνω. Το περιεχόμενο των πινάκων μπορεί να στοιχιστεί και ως προς τον κατακόρυφο άξονα. Για το σκοπό αυτό χρησιμοποιείται το χαρακτηριστικό `valign`, το οποίο παίρνει τις τιμές `top`, `bottom` και `middle`, όσον αφορά τη στοίχιση του κειμένου στην κορυφή, στην κάτω περιοχή ή στο μέσο του κελιού, αντίστοιχα. Η HTML παρέχει τη δυνατότητα για συνδυασμό των παραπάνω χαρακτηριστικών, στοχεύοντας στην καλύτερη εμφάνιση των περιεχομένων ενός πίνακα. Με τον τρόπο δημιουργίας των πινάκων που περιγράφηκε παραπάνω υλοποιούνται πίνακες που έχουν σε όλες τις γραμμές και σε όλες τους τις στήλες τον ίδιο αριθμό κελιών. Σε περίπτωση, όμως, που είναι χρήσιμο να δημιουργηθούν κελιά για ένα πίνακα που επεκτείνονται σε περισσότερες από μία στήλες ή σε περισσότερες από μία γραμμές, πρέπει κάτι τέτοιο να δηλωθεί ρητά, κάνοντας χρήση των χαρακτηριστικών `colspan` και `rowspan` στις ετικέτες `<td>...</td>` και `<th>...</th>`. Ανάλογα με τη τιμή που θα δοθεί σε κάθε περίπτωση στα παραπάνω δύο χαρακτηριστικά, δημιουργούνται κελιά που καταλαμβάνουν συγκεκριμένο αριθμό γραμμών και στηλών του πίνακα. Για την μορφοποίηση των πινάκων προσφέρονται και άλλα χαρακτηριστικά των ετικετών που αναφέρθηκαν παραπάνω. Έτσι, με την ιδιότητα `cellpadding` καθορίζεται η απόσταση (σε pixels) που θα έχουν τα περιεχόμενα των κελιών ενός πίνακα από το περίγραμμα του κάθε κελιού. Ακόμα, υπάρχει το περιθώριο να οριστεί και η απόσταση (σε pixels) μεταξύ των ίδιων των κελιών ενός πίνακα, κάνοντας χρήση της ιδιότητας `cellspacing`. Επίσης, μπορεί να δηλωθεί και το μέγεθος ενός πίνακα στην οθόνη ενός πλοηγού. Αν δε δηλωθεί ξεκάθαρα το προαναφερόμενο μέγεθος, τότε ο προηγούμενος πίνακας θα εμφανιστεί να έχει μέγεθος που εξαρτάται από το μέγεθος των περιεχομένων των κελιών του. Για να οριστεί ρητά το μέγεθος ενός πίνακα χρησιμοποιείται το χαρακτηριστικό `width` της ετικέτας `<table>`. Το πλάτος ενός πίνακα μπορεί να καθοριστεί είτε απόλυτα (σε αριθμό pixels) είτε σχετικά (με το μέγεθος της οθόνης του εκάστοτε περιηγητή, σε ποσοστό επί τις εκατό). Σε έναν πίνακα γίνεται να αποδοθεί τίτλος, χρησιμοποιώντας την ετικέτα `<caption>...</caption>`. Η ετικέτα αυτή πρέπει να βρίσκεται μέσα στην ετικέτα `<table>...</table>` και δύναται να περικλείει οποιοδήποτε HTML κώδικα. Ο παραπάνω τίτλος εμφανίζεται κεντραρισμένος στην πάνω περιοχή του προαναφερόμενου πίνακα. Αν είναι αρεστό ο συγκεκριμένος τίτλος να εμφανίζεται στο κάτω μέρος του τρέχοντος πίνακα, τότε διατίθεται η ιδιότητα `align` της ετικέτας `caption` με τιμή `bottom`, σύμφωνα με τη μορφή: `<caption align="bottom">...</caption>`.

- Τα πλαίσια (frames): Όταν πραγματοποιείται ένα κλικ πάνω σε ένα δεσμό υπερκειμένου μίας Ιστοσελίδας, τότε η γλώσσα HTML παρέχει τη δυνατότητα της εμφάνισης των περιεχομένων αυτού του δεσμού είτε σε καινούργια παράθυρα του εκάστοτε περιηγητή είτε σε (ορθογώνια) τμήματα του παρόντος σε κάθε περίπτωση παραθύρου του προαναφερόμενου περιηγητή. Σχετική έννοια με τα προηγούμενα είναι και η έννοια των πλαισίων (frames). Τα πλαίσια κάνουν μία Σελίδα να μοιάζει με ένα αρχείο, ενώ στην ουσία αποτελείται από περισσότερα. Αυτό έχει ως πλεονέκτημα ορισμένα τμήματα της συγκεκριμένης Σελίδας να παραμένουν

σταθερά, ενώ κάποια άλλα τμήματά της μεταβάλλονται (πχ σε ένα τμήμα του παραθύρου-πλαίσιου ενός φυλλομετρητή μπορεί να υπάρχει ένας πίνακας περιεχομένων που να παραμένει σταθερός, ενώ σε άλλο παράθυρο-πλαίσιο του πρωτύπου περιηγητή δύναται να εμφανίζονται τα περιεχόμενα της Σελίδας). Ως μειονέκτημα της τεχνικής των πλαισίων συχνά αναφέρεται η ανάγκη δημιουργίας και συντήρησης περισσότερων HTML αρχείων καθώς και η επιπλέον καθυστέρηση στη φόρτωση μίας Ιστοσελίδας από τον εκάστοτε πλοηγό. Το πρώτο βήμα για τη δημιουργία μίας Ιστοσελίδας, που θα περιέχει πλαίσια, είναι η από την αρχή δήλωση του τρόπου με τον οποίο θα χωρίζεται η οθόνη του εκάστοτε πλοηγού σε ένα ξεχωριστό αρχείο. Το αρχείο αυτό θα περιέχει την ετικέτα <frameset>...</frameset> στη θέση της ετικέτας <body>...</body> στον κώδικα της προαναφερόμενης Σελίδας. Για τον ορισμό της δομής των πλαισίων, η ετικέτα <frameset>...</frameset> θα πρέπει να περιέχει το χαρακτηριστικό rows ή το χαρακτηριστικό cols, ώστε να είναι δυνατή η διάσπαση μίας Σελίδας σε πλαίσια, κατά τον κατακόρυφο ή τον οριζόντιο άξονα, αντίστοιχα. Σε περίπτωση που είναι αναγκαία η δημιουργία μίας πιο περίπλοκης δομής πλαισίων, χρησιμοποιούνται ένθετες ετικέτες <frameset>...</frameset>. Οι τιμές που εισάγονται στα χαρακτηριστικά rows και cols, καθορίζουν τον αριθμό και τις διαστάσεις των γραμμών ή των στηλών μίας Σελίδας, αντίστοιχα, και δίνονται (διαχωριζόμενες με κόμμα) ως λίστα των διαστάσεων των πλαισίων είτε σε απόλυτο αριθμό pixels είτε σε ποσοστό επί τοις εκατό (%) του παραθύρου του εκάστοτε πλοηγού. Στο εσωτερικό της ετικέτας <frameset>...</frameset> οφείλει να υπάρχει αριθμός ετικετών <frame> ίσος με τον αριθμό των πλαισίων που καθορίστηκαν στην ετικέτα <frameset>...</frameset> Κάθε ετικέτα <frame> περιέχει την ιδιότητα src, η τιμή της οποίας είναι μία URL ή ένα τοπικό αρχείο.

- Το προσχηματισμένο κείμενο (preformatted text): Ένα από τα υπερτερήματα της γλώσσας HTML είναι η εμφάνιση του κειμένου μίας Ιστοσελίδας, από τη στιγμή που αυτό δεν χωράει σε μία γραμμή αυτής της Σελίδας στην επόμενη γραμμή της. Επιπρόσθετα, το εκάστοτε πρόγραμμα περιήγησης αγνοεί τυχόν κενά διαστήματα, που παρουσιάζονται ανάμεσα στις διάφορες λέξεις ενός κειμένου μίας Σελίδας. Υπάρχουν, ωστόσο, περιπτώσεις στις οποίες είναι θεμιτή η απενεργοποίηση αυτής της δυνατότητας της HTML. Σε τέτοιες περιπτώσεις εφαρμόζεται η ετικέτα προσχηματισμένου κειμένου <pre>...</pre>, η οποία έχει τη δυνατότητα να αναπαριστά τον κώδικα μίας Σελίδας, δίχως να αλλοιώνει τη μορφή του. Εντός αυτής της ετικέτας μπορεί να χρησιμοποιηθούν και άλλες ετικέτες της HTML. Μία σημαντική, λοιπόν, χρήση της τρέχουσας ετικέτας είναι η αναπαράσταση του κώδικα μίας Σελίδας, χωρίς να τροποποιείται η μορφή του.
- Οι ειδικοί χαρακτήρες (special characters): Σε πολλές περιπτώσεις είναι απαραίτητη η εμφάνιση ενός ειδικού χαρακτήρα στο κείμενο μίας Ιστοσελίδας. Ένα τέτοιο παράδειγμα συνιστά το σύμβολο © του Copyright, που πολλοί κατασκευαστές Σελίδων χρησιμοποιούν στο περιεχόμενο των Σελίδων. Στον παρακάτω Πίνακα φαίνονται οι ειδικοί χαρακτήρες που αποτυπώνονται συχνότερα σε μία Σελίδα:

ΟΝΟΜΑΣΙΑ	ΚΩΔΙΚΟΣ	ΣΥΜΒΟΛΟ
Ενότητα	§	§
Παράγραφος	¶	¶
Αγγλικό ερωτηματικό	¢	?
Άνω τελεία	·	·
Διπλό μικρότερο από	«	«
Διπλό μεγαλύτερο από	»	»
Συν-Πλην	±	±
Βαθμός	°	°
Copyright	©	©
Registered	®	®
Trademark	™	™
Ευρώ	€	€
Βρετανική Λίρα	£	£
Γιεν	¥	¥

Πίνακας 4: Οι πιο βασικοί ειδικοί χαρακτήρες της HTML

Μία άλλη χρησιμότητα των ειδικών χαρακτήρων είναι η αναπαράσταση των συμβόλων <, > και &. Οι τρεις αυτοί χαρακτήρες έχουν ειδική σημασία στη γλώσσα HTML και για το λόγο αυτό δεν μπορούν να γίνουν εκμεταλλεύσιμοι απευθείας στον εκάστοτε HTML κώδικα. Έτσι, οι χαρακτήρες αυτοί αναπαρίστανται από τις ακολουθίες <, > και &, αντίστοιχα. Τέλος, αν είναι ωφέλιμη η ύπαρξη επιπλέον κενών διαστημάτων μεταξύ των λέξεων ενός κειμένου, υπάρχει η δυνατότητα εκμετάλλευσης του χαρακτήρα , που μία σημαντική χρήση του είναι η δημιουργία κενών κελιών σε πίνακες (όπως ήδη αναφέρθηκε πιο πάνω).

Συγκεφαλαιώνοντας, παραπάνω έλαβε χώρα μία αναφορά στις βασικές ετικέτες ενός HTML εγγράφου και στα κύρια χαρακτηριστικά τους. Σε καμιά περίπτωση αυτό δε σημαίνει ότι στη γλώσσα HTML δε συναντώνται επιπλέον ετικέτες και επιπρόσθετα χαρακτηριστικά πέραν των προαναφερόμενων. Έτσι, θα ήταν κρίσιμη παράλειψη να μη γίνει μνεία στην ετικέτα <div>...</div>, η οποία χρησιμοποιείται, για να μορφοποιήσει μία ομάδα από στοιχεία ενός HTML εγγράφου. Ύστερα, θα ήταν λάθος να μην πραγματοποιηθεί αναφορά στην ετικέτα <style>...</style>, που καθορίζει την εμφάνιση HTML συστατικών. Ακόμα, είναι θεμιτό να αναφερθεί η ετικέτα <META>, η οποία αποσκοπεί στη συμπερίληψη στην εκάστοτε Ιστοσελίδα πληροφοριών που δεν εμφανίζονται σε ένα πρόγραμμα περιήγησης. Η ετικέτα αυτή βρίσκεται εντός της ετικέτας <head>...</head> και συχνά χρησιμοποιείται για τη μετάβαση σε διαφορετικές Ιστοσελίδες, μετά από προκαθορισμένο χρονικό διάστημα καθώς και για τον εντοπισμό Σελίδων από μηχανές αναζήτησης στο Διαδίκτυο.

Τέλος, η γλώσσα HTML παρέχει και τη δυνατότητα εισαγωγής δεδομένων από τους χρήστες. Τα δεδομένα αυτά αποστέλλονται (πχ σε περίπτωση περιηγητή με γραφικό περιβάλλον μέσω του πατήματος ενός πλήκτρου που εμφανίζεται στην εκάστοτε Ιστοσελίδα) από το φυλλομετρητή του εκάστοτε χρήστη στον εκάστοτε εξυπηρετητή Ιστού όπου βρίσκεται αποθηκευμένη η εκάστοτε Ιστοσελίδα.

[34] [35]

2.11 Η Γλώσσα Μορφοποίησης Περιεχομένου Ιστοσελίδων, CSS

2.11.1 Εισαγωγή στη CSS

Η γλώσσα CSS (Cascading Style Sheets) (απόδοση στα Ελληνικά: Διαδοχικά Φύλλα Στυλ ή Αλληλουχία Φύλλων Στυλ) είναι μία απλή γλώσσα μορφοποίησης των στοιχείων που απαρτίζουν μία Ιστοσελίδα, η οποία βοηθάει να οριστεί με σαφήνεια και ιδιαίτερη ευελιξία ο τρόπος με τον οποίο αυτά τα στοιχεία θα εμφανίζονται στη συγκεκριμένη Ιστοσελίδα.

[37]

2.11.2 Οι Κανόνες Σύνταξης της CSS

Η γλώσσα CSS υπακούει σε συγκεκριμένους κανόνες σύνταξης, οι βασικότεροι από τους οποίους αναφέρονται παρακάτω:

- Ο CSS κανόνας (CSS rule): Αποτελείται από δύο μέρη: τον CSS επιλογέα (CSS selector), που αφορά το τι θα μορφοποιηθεί, και τις CSS ιδιότητες (CSS properties), οι οποίες αφορούν το πως αυτό θα μορφοποιηθεί. Η σύνταξη του CSS κανόνα είναι η εξής:

```
επιλογέας
{
...ιδιότητα1: τιμη1;
...ιδιότητα2: τιμη2;
...ιδιότητα3: τιμη3;
.....
}
```

- Οι CSS επιλογείς (CSS selectors): Επιλέγουν εκείνα τα στοιχεία μίας Ιστοσελίδας που πρόκειται να μορφοποιηθούν. Ουσιαστικά αποτελούν κάποιες διακριτές ετικέτες, ώστε να καταλάβει

ο εκάστοτε πλοηγός σε ποια σημεία μίας Σελίδας είναι επιθυμητό να εφαρμοστούν συγκεκριμένοι κανόνες μορφοποίησης. Υπάρχουν αρκετά περίπλοκοι επιλογείς στις προδιαγραφές της γλώσσας CSS, οι οποίοι προσφέρουν τεράστια ευελιξία, ωστόσο στη συνέχεια θα εξεταστούν οι πιο βασικοί είτε για λόγους απλούστευσης είτε επειδή αρκετοί από τους πιο περίπλοκους δεν υποστηρίζονται από όλους τους σύγχρονους περιηγητές. Οι χρησιμότεροι, λοιπόν, CSS επιλογείς είναι οι ακόλουθοι:

- ο *: Επιλέγει όλα τα στοιχεία μίας Ιστοσελίδας. Όπως είναι κατανοητό συνήθως δεν είναι και πολύ χρήσιμος ο επιλογέας αυτός από μόνος του, συνεπώς χρησιμοποιείται κυρίως σε συνδυασμό με άλλους επιλογείς.
 - ο στοιχείο: Επιλέγει εκείνα τα στοιχεία μίας Ιστοσελίδας τα οποία περιέχονται εντός της ετικέτας που ονομάζεται στοιχείο.
 - ο .όνομα_κλάσης: Επιλέγει όλα τα στοιχεία μίας Ιστοσελίδας που φέρουν την ιδιότητα class με τιμή όνομα_κλάσης. Δηλαδή όλα τα στοιχεία που ανήκουν στην κλάση που καλείται όνομα_κλάσης. Φυσικά το όνομα μίας κλάσης μπορεί να περιέχει οποιοδήποτε γράμμα του λατινικού ή άλλου αποδεκτού αλφαβήτου, αριθμούς, παύλες (-) ή κάτω παύλες (_), αλλά να ξεκινάει αποκλειστικά με γράμμα. Αξίζει να σημειωθεί ότι υπάρχει η πιθανότητα το ίδιο κάθε φορά στοιχείο να ανήκει σε περισσότερες από μία κλάσεις, διαχωρισμένες με κενά μέσα στην ιδιότητά του class. Οι κλάσεις γενικά χρησιμοποιούνται, αφότου είναι ωφέλιμη η ομαδοποίηση κάποιων HTML στοιχείων, για τα οποία δεν γίνεται να βρεθεί κάποιος άλλος επιλογέας, που να αφορά αποκλειστικά αυτά τα στοιχεία, οπότε ομαδοποιούνται σε μία συγκεκριμένη κλάση. Έτσι, είναι εφικτή στον εκάστοτε CSS κώδικα μία στοχευμένη αναφορά σε αυτά και μόνο τα στοιχεία, προκειμένου να μορφοποιηθούν, δίχως να επηρεαστεί η μορφή άλλων στοιχείων, τα οποία δεν ανήκουν στην προαναφερόμενη κλάση.
 - ο στοιχείο.όνομα_κλάσης: Αποτελεί ουσιαστικά συνδυασμό των δύο αμέσως παραπάνω επιλογέων. Επιλέγει όσα στοιχεία μίας Ιστοσελίδας περιλαμβάνονται εντός της HTML ετικέτας που ονομάζεται στοιχείο και ανήκουν στην κλάση με όνομα όνομα_κλάσης. Ο επιλογέας αυτός είναι χρήσιμος, όταν υπάρχουν πολλά διαφορετικού τύπου στοιχεία στην ίδια κάθε φορά κλάση και είναι αναγκαία η εφαρμογή διαφορετικής μορφοποίησης στα στοιχεία αυτά, με βάση τον τύπο τους.
 - ο #όνομα_id: Επιλέγει όλα τα στοιχεία μίας Ιστοσελίδας που φέρουν την ιδιότητα id με τιμή όνομα_id. Δεν πρέπει να υπάρχουν δύο (ή περισσότερα) στοιχεία στην εκάστοτε όμοια Σελίδα με το ίδιο id. Τα ids διέπονται από τους ίδιους κανόνες ονοματολογίας με τις κλάσεις. Ουσιαστικά ό,τι μπορεί να επιτευχθεί με τα ids μπορεί να επιτευχθεί και με τη χρήση κλάσεων. Εφόσον, όμως, το στοιχείο που σε κάθε περίπτωση πρόκειται να μορφοποιηθεί είναι μοναδικό, συμφέρει η εκμετάλλευση των ids.
 - ο στοιχείο[ιδιότητα=«τιμή»]: Αποτελεί ουσιαστικά μία επέκταση του επιλογέα «στοιχείο». Επιλέγει όσα στοιχεία μίας HTML Ιστοσελίδας που περιλαμβάνονται εντός της HTML ετικέτας που λέγεται στοιχείο και φέρουν την ιδιότητα που ορίζει ο τρέχων επιλογέας με τιμή εκείνη που καθορίζει και πάλι ο παρών επιλογέας.
- Οι συνδυαστικοί CSS επιλογείς (CSS combinators): Πολλές φορές στη γλώσσα CSS είναι πρακτικό να συνδυάζονται δύο ή περισσότεροι επιλογείς μεταξύ τους. Με αυτό τον τρόπο προσφέρεται στους προγραμματιστές του Διαδικτύου μεγαλύτερη ευελιξία και εξοικονόμηση χρόνου. Υπάρχουν πέντε συνδυαστικοί CSS επιλογείς, που είναι οι ακόλουθοι:
 - I. Ο CSS επιλογέας ομάδας (CSS group selector): Υλοποιείται με τη χρήση κόμματος μεταξύ δύο ή περισσότερων CSS επιλογέων και επιλέγει τα HTML στοιχεία μίας Ιστοσελίδας που πληρούν τις προϋποθέσεις καθενός από τους επιμέρους CSS επιλογείς που συγκροτούν τον τρέχων CSS επιλογέα.
 - II. Ο CSS επιλογέας «απογόνου» (CSS “descendant” selector): Υλοποιείται με τη χρήση του χαρακτήρα του κενού μεταξύ δύο ή περισσότερων CSS επιλογέων και επιλέγει τα HTML στοιχεία μίας Ιστοσελίδας που είναι «απόγονοι» ενός συγκεκριμένου στοιχείου.
 - III. Ο CSS επιλογέας «παιδιού» (CSS “child” selector): Υλοποιείται με τη χρήση του χαρακτήρα «>» μεταξύ δύο ή περισσότερων CSS επιλογέων και επιλέγει τα HTML

- στοιχεία μίας Ιστοσελίδας που είναι «άμεσα» παιδιά ενός συγκεκριμένου στοιχείου.
- IV. Ο CSS επιλογέας «προσκειμένου αδερφού» (CSS “adjacent sibling” selector): Υλοποιείται με τη χρήση του χαρακτήρα «+» μεταξύ δύο ή περισσότερων CSS επιλογέων και επιλέγει τα HTML στοιχεία μίας Ιστοσελίδας που είναι «προσκειμένα», δηλαδή τα αμέσως επόμενα, αμφιθαλή «αδέλφια» ενός συγκεκριμένου στοιχείου.
- V. Ο CSS επιλογέας «γενικού αδερφού» (CSS “general sibling” selector): Υλοποιείται με τη χρήση του χαρακτήρα «~» μεταξύ δύο ή περισσότερων CSS επιλογέων και επιλέγει τα HTML στοιχεία μίας Ιστοσελίδας που είναι «αδέλφια» με τη γενικευμένη έννοια του όρου, δηλαδή δίχως να είναι κατανάγκη διαδοχικά, ενός συγκεκριμένου στοιχείου.
- Οι ψευδο-κλάσεις (pseudo-classes) και τα ψευδο-στοιχεία (pseudo-elements): Ορισμένες φορές στη γλώσσα CSS γίνεται χρήση των λεγόμενων ψευδο-κλάσεων ή ψευδο-στοιχείων που επιτρέπουν την επιλογή στοιχείων που δεν αποτελούν γνήσια HTML στοιχεία, αλλά κομμάτια τους ή συγκεκριμένες καταστάσεις τους. Στην πράξη συνιστούν κάποιες λέξεις-κλειδιά (keywords) οι οποίες γράφονται μετά από έναν επιλογέα τύπου «στοιχείο» και αρχίζουν με άνω-κάτω τελεία (:). Οι ψευδο-κλάσεις που χρησιμοποιούνται συνηθέστερα είναι οι ακόλουθες:
 - ο a:link: Αφορά HTML στοιχεία τα οποία περιέχονται εντός του ζεύγους ετικετών <a> και αφορά τους συνδέσμους που ο εκάστοτε χρήστης δεν έχει ακόμα επισκεφθεί.
 - ο a:visited: Αφορά HTML στοιχεία τα οποία περιέχονται εντός του ζεύγους ετικετών <a> και αφορά τους συνδέσμους που ο εκάστοτε χρήστης έχει επισκεφθεί.
 - ο στοιχείο:active: Αφορά HTML στοιχεία τα οποία περιέχονται εντός της ετικέτας με την ονομασία στοιχείο, εκείνη τη στιγμή που ο εκάστοτε χρήστης έχει πατημένο κάποιο πλήκτρο του ποντικιού πάνω σε αυτά.
 - ο στοιχείο:hover: Αφορά HTML στοιχεία τα οποία περιέχονται εντός της ετικέτας με την ονομασία στοιχείο, εκείνη τη στιγμή που ο εκάστοτε χρήστης έχει το δείκτη του ποντικιού πάνω σε κάποιο από αυτά, χωρίς, όμως, να πατάει κάποιο πλήκτρο. Η τρέχουσα ψευδο-κλάση είναι από τις πιο συχνά εκμεταλλεύσιμες ψευδο-κλάσεις, η οποία βοηθάει στη δημιουργία διαφόρων όμορφων εφέ στις HTML Ιστοσελίδες, τα οποία παλιότερα ήταν εφικτά μόνο με τη χρήση JavaScript.
 - ο στοιχείο:focus: Αφορά HTML στοιχεία διαδικτυακών φορμών τα οποία περιέχονται εντός της ετικέτας με την ονομασία στοιχείο, εκείνη τη στιγμή που ο εκάστοτε χρήστης έχει εστιάσει πάνω τους.
 - ο στοιχείο:first-letter: Αφορά το πρώτο γράμμα ενός κειμένου που υπάρχει μέσα σε HTML στοιχεία τα οποία περιέχονται εντός της ετικέτας με την ονομασία στοιχείο και συντελεί στη δημιουργία αρχιγραμμάτων.
 - ο στοιχείο:first-line: Αφορά ολόκληρη την πρώτη γραμμή ενός κειμένου που υπάρχει μέσα σε HTML στοιχεία τα οποία περιέχονται εντός της ετικέτας με την ονομασία στοιχείο.
 - Τα σχόλια (comments): Μολονότι η ανάγκη για σχόλια στη γλώσσα CSS δεν είναι τόσο επιτακτική όσο σε άλλες γλώσσες μορφοποίησης ή στις γλώσσες προγραμματισμού, είναι δυνατό να εισαχθούν σχόλια στη CSS μεταξύ των χαρακτήρων /* και */.

[35] [38]

2.11.3 Οι Ιδιότητες της CSS

Τα σημαντικότερα χαρακτηριστικά (γνωρίσματα)\ιδιότητες της γλώσσας CSS είναι τα επόμενα:

- Χαρακτηριστικά γνωρίσματα που σχετίζονται με τη μορφοποίηση του κειμένου μίας Ιστοσελίδας: Τα πιο κύρια συνοψίζονται παρακάτω:
 - ο color: Αναφέρεται στο χρώμα του κειμένου μίας Ιστοσελίδας. Ωστόσο, αν δεν οριστεί χρώμα περιγράμματος για μία Σελίδα (μέσω της ιδιότητας border-color), ο εκάστοτε πλοηγός χρησιμοποιεί για το συγκεκριμένο χρώμα, το χρώμα που είναι καταχωρημένο στην ιδιότητα color. Τα χρώματα μπορούν να εισαχθούν είτε σε μορφή RGB (πχ color: rgb(255,128,30);), είτε σε δεκαεξαδική μορφή (πχ color: #ff801e;), είτε με τη μορφή

- κάποιας λέξης-κλειδί (πχ color: orange;).
- o font-size: Σχετίζεται το μέγεθος της γραμματοσειράς του κειμένου μίας Ιστοσελίδας. Οι τιμές που δέχεται είναι δυνατό να είναι εκφρασμένες σε ένα μεγάλο πλήθος μονάδων μεγέθους, από τις οποίες οι πιο ευρέως διαδεδομένες είναι τα εικονοστοιχεία (pixels) (πχ font-size: 12px;) και οι στιγμές (points) (πχ font-size: 10pt;).
 - o font-family: Επιτρέπει να οριστεί ένα πλήθος γραμματοσειρών, που θα χρησιμοποιηθούν κατά σειρά προτίμησης στο κείμενο που εμφανίζεται σε μία Ιστοσελίδα. Ουσιαστικά, μέσω αυτής της ιδιότητας, δεν καθορίζεται απλά η κύρια γραμματοσειρά ενός κειμένου μίας Σελίδας, αλλά ταυτόχρονα ορίζονται και εναλλακτικές επιλογές, ώστε αν η γραμματοσειρά που σε κάθε φάση επιλέχθηκε, δεν υπάρχει στον υπολογιστή του εκάστοτε χρήστη, να μην καταστρέψει ο εκάστοτε πλοηγός την εμφάνιση του περιεχομένου της παραπάνω Σελίδας, επιλέγοντας αυθαίρετα ο συγκεκριμένος φυλλομετρητής μία οποιαδήποτε γραμματοσειρά, αλλά να διαλέξει την επόμενη διαθέσιμη γραμματοσειρά. Αν καμία από όσες γραμματοσειρές ορίστηκαν στον κώδικα μίας Σελίδας με τη χρήση της ιδιότητας font-family δεν υπάρχει στον υπολογιστή ενός χρήστη, τότε ο εκάστοτε περιηγητής επιλέγει κάποια από τη γενική οικογένεια γραμματοσειρών (generic font-family), η οποία ορίζεται τελευταία με την κατάλληλη λέξη-κλειδί. Μερικά παραδείγματα λέξεων-κλειδιά για οικογένειες γραμματοσειρών που εκμεταλλεύονται οι διάφοροι Ιστότοποι είναι τα επόμενα:
 - serif: Γραμματοσειρές με «πατούρες», όπως είναι, για παράδειγμα, η Georgia.
 - sans-serif: Γραμματοσειρές χωρίς «πατούρες», όπως είναι, για παράδειγμα, η Trebuchet MS.
 - monospace: Γραμματοσειρές όπου το κάθε γράμμα καταλαμβάνει ίσο πλάτος με τα υπόλοιπα, όπως είναι, για παράδειγμα, η Courier New.
 - cursive: Καλλιγραφικές γραμματοσειρές, όπως είναι, για παράδειγμα, η Monotype Corsiva.
 - fantasy: «Διακοσμητικές» γραμματοσειρές.
 - o font-weight: Αφορά το «βάρος» της γραμματοσειράς του κειμένου μίας Ιστοσελίδας και στην πράξη διατίθεται, για να ορίσει, αν αυτό το κείμενο θα είναι έντονο ή όχι, μίας και οι περισσότερες γραμματοσειρές που χρησιμοποιούνται στο Διαδίκτυο διατίθενται μόνο σε δύο βάρη: κανονικό και έντονο, σε αντίθεση με πιο εξειδικευμένες γραμματοσειρές που χρησιμοποιούνται από γραφίστες, οι οποίες συχνά διατίθενται σε διάφορα βάρη. Οπότε, οι τιμές που συνήθως διατίθενται σε αυτή την ιδιότητα είναι οι normal και bold, η λειτουργία των οποίων είναι προφανής.
 - o text-decoration: Εφαρμόζει στο περιεχόμενο μίας Ιστοσελίδας διάφορα εφέ, συμπεριλαμβανόμενης και της υπογράμμισης. Οι τιμές που δέχεται είναι οι παρακάτω (και μπορούν να χρησιμοποιηθούν και συνδυαστικά, όταν είναι επιθυμητή η εφαρμογή περισσότερων του ενός εφέ στην εκάστοτε Σελίδα):
 - none: Καμία διακόσμηση.
 - underline: Υπογράμμιση.
 - overline: Γραμμή πάνω από το κείμενο μίας Ιστοσελίδας (ουσιαστικά το αντίθετο της υπογράμμισης).
 - line-through: Διαγράμμιση.
 - blink: Κάνει το κείμενο μίας Ιστοσελίδας να αναβοσβήνει. Δεν πρέπει να χρησιμοποιείται, παρά μόνο σε εξαιρετικά σπάνιες περιπτώσεις, διότι η εμπειρία έχει δείξει ότι η συγκεκριμένη τεχνική, όταν εφαρμόζεται σε μία Σελίδα, είναι ιδιαίτερα κουραστική για τον εκάστοτε αναγνώστη αυτής της Σελίδας και, επιπλέον, αφαιρεί επαγγελματικότητα από το σχεδιασμό της.
 - o align: Καθορίζει τη στοίχιση του κειμένου ενός HTML εγγράφου. Οι τιμές που παίρνει έχουν ως εξής:
 - left: Αριστερή στοίχιση.
 - center: Στοίχιση στο κέντρο.
 - right: Δεξιά στοίχιση.
 - justify: Πλήρης στοίχιση.

- o font: Ορίζει τη γραμματοσειρά που χρησιμοποιείται στην απεικόνιση των αντικειμένων μίας Ιστοσελίδας. Επιμέρους ιδιότητες που τη συνοδεύουν είναι όσες ιδιότητες σχετίζονται με τη γραμματοσειρά του κειμένου μίας Ιστοσελίδας και είτε αναφέρθηκαν παραπάνω είτε δεν αναφέρθηκαν για λόγους απλούστευσης. Εδώ, να προσεχτεί ότι, αν κάποια από τις προαναφερόμενες ιδιότητες δε συμπεριληφθεί, τότε ο εκάστοτε φυλλομετρητής θα υποθέσει την προεπιλεγμένη τιμή της.
- Χαρακτηριστικά που αφορούν το φόντο μίας Ιστοσελίδας: Τα σημαντικότερα είναι τα ακόλουθα:
 - o background-color: Χρησιμοποιείται, για να ορίσει χρώμα φόντου μίας Σελίδας. Το χρώμα αυτό μπορεί να γραφεί σε οποιαδήποτε από τις μορφές που περιγράφηκαν πιο πάνω για την ιδιότητα color.
 - o background-image: Εφαρμόζεται, ώστε να καθορίσει η εικόνα φόντου μίας Ιστοσελίδας. Η τιμή που δέχεται είναι της μορφής “url(διεύθυνση_εικόνας)”. Γενικά, όταν εισάγεται μία διεύθυνση URL ως τιμή (ή τμήμα τιμής) κάποιας CSS ιδιότητας, γράφεται πάντα σύμφωνα με την προαναφερθείσα μορφή. Φυσικά υπάρχει η ευχέρεια της εισαγωγής σχετικών URLs.
 - o background-repeat: Σχετίζεται τον τρόπο που θα επαναλαμβάνεται η εικόνα φόντου μίας Ιστοσελίδας. Οι τιμές που δέχεται είναι οι πιο κάτω:
 - no-repeat: Καμία επανάληψη.
 - repeat-x: Επανάληψη κατά τον οριζόντιο άξονα.
 - repeat-y: Επανάληψη κατά τον κατακόρυφο άξονα.
 - repeat: Επανάληψη τόσο κατά τον κατακόρυφο όσο και κατά τον οριζόντιο άξονα. Η τιμή repeat είναι και η προεπιλεγμένη τιμή της ιδιότητας background-repeat.
 - o background-position: Χρησιμοποιείται συνήθως, αφότου έχει ζητηθεί, μέσω της ιδιότητας background-repeat, να μην επαναλαμβάνεται η εικόνα φόντου μίας Ιστοσελίδας, που ορίστηκε με την ιδιότητα background-image. Επιτρέπει τον ορισμό τη θέσης που θα τοποθετηθεί η συγκεκριμένη εικόνα τόσο κάθετα όσο και οριζόντια, αν και κάποιοι περιηγητές υποστηρίζουν χωριστή δήλωση θέσης της προαναφερόμενης εικόνας στον κάθετο και τον οριζόντιο άξονα, με τη βοήθεια των ιδιοτήτων background-position-x και background-position-y, των οποίων, ωστόσο, καλύτερα να αποφεύγεται η χρήση, επειδή δεν υποστηρίζονται από όλους τους σύγχρονους φυλλομετρητές. Οι τιμές που δέχεται η ιδιότητα background-position είναι κάποιες λέξεις-κλειδιά. Επίσης, υπάρχει η δυνατότητα ορισμού της θέσης μίας εικόνας φόντου και αριθμητικά, είτε σε ποσοστό επί τοις εκατό (%), είτε σε εικονοστοιχεία (pixels), είτε σε στιγμές (points), αν και οι τιμές των εικονοστοιχείων και των στιγμών χρησιμοποιούνται πολύ σπάνια. Οι προαναφερόμενες λέξεις-κλειδιά, που χρησιμοποιούνται συνδυαστικά τις περισσότερες φορές, είναι οι ακόλουθες:
 - left: Αριστερά, σε σχέση με τον οριζόντιο άξονα.
 - center: Στο κέντρο, σε σχέση με τον οριζόντιο άξονα.
 - right: Δεξιά, σε σχέση με τον οριζόντιο άξονα.
 - top: Στην κορυφή, σε σχέση με τον κατακόρυφο άξονα.
 - center: Στο κέντρο, σε σχέση με τον κατακόρυφο άξονα
 - bottom: Στον πυθμένα, σε σχέση με τον κατακόρυφο άξονα.
 Στο σημείο αυτό, θα ήταν καλό να τονιστεί πως υπάρχει η δυνατότητα να καθοριστεί η τοποθέτηση μόνο κατά τον ένα άξονα, οπότε για τον άλλο υποτίθεται η προεπιλεγμένη τιμή, η οποία είναι συνήθως top για τον κάθετο άξονα και left για τον οριζόντιο.
 - o background: Αποτελεί επί της ουσίας μία συντόμευση, προκειμένου να οριστούν με μία μόνο ιδιότητα όλες οι παραπάνω (όπως είναι η ιδιότητα font για τα χαρακτηριστικά κειμένου μίας Σελίδας). Πρέπει να προσεχθεί ιδιαίτερα ότι, αν δεν οριστεί κάποια επιμέρους ιδιότητα, όταν χρησιμοποιείται η συγκεντρωτική ιδιότητα background, ο εκάστοτε πλοηγός θεωρεί την προεπιλεγμένη τιμή για αυτή και όχι πως δεν ορίστηκε.
- Ιδιότητες που αφορούν το περίγραμμα: Οι βασικότερες είναι οι κάτωθι:
 - o border-color: Καθορίζει το χρώμα περιγράμματος μίας Ιστοσελίδας. Το χρώμα μπορεί

να γραφεί σε οποιαδήποτε από τις μορφές που περιγράφηκαν πιο πάνω για την ιδιότητα color. Αν δεν οριστεί η τρέχουσα ιδιότητα, τότε, για το χρώμα περιγράμματος μίας Σελίδας, διατίθεται το χρώμα που ορίστηκε στην ιδιότητα color.

- ο border-width: Ρυθμίζει το πάχος του περιγράμματος μίας Ιστοσελίδας σε κάποια από τις μονάδες μέτρησης που διατίθενται στη CSS, συνήθως σε pixels.
- ο border-style: Ορίζει το στυλ του περιγράμματος μίας Ιστοσελίδας. Οι τιμές που διατίθενται συνήθως για την ιδιότητα αυτή είναι οι παρακάτω:
 - solid: Συμπαγές περίγραμμα, δηλαδή χωρίς κάποια διακόσμηση, μία ενιαία γραμμή.
 - dashed: Περίγραμμα που αποτελείται από παύλες.
 - dotted: Περίγραμμα που αποτελείται από τελείες. Αξίζει να σημειωθεί πως τα αποτελέσματα της χρήσης της τιμής αυτής διαφέρουν ελαφρώς ανά τους περιηγητές.
 - double: Παρόμοια με την τιμή solid, μόνο που δημιουργεί δύο περιγράμματα, η απόσταση και το πάχος των οποίων δεν μπορούν να ρυθμιστούν χωριστά, αλλά ρυθμίζονται αυτόματα ταυτόχρονα, ώστε το πάχος τους να ισούται με την εκάστοτε τιμή που ορίστηκε στην ιδιότητα border-width. Συνήθως χρησιμοποιείται σε συνδυασμό με πάχος περιγράμματος 3px, αποδίδοντας ένα αρκετά καλαίσθητο αποτέλεσμα.
- ο border: Συνιστά στην ουσία συντόμευση, για να οριστούν με μία μόνο ιδιότητα όλες οι παραπάνω, σχετικές με το περίγραμμα ιδιότητες (όπως θεωρείται η ιδιότητα font για τα χαρακτηριστικά κειμένου μίας Σελίδας και η ιδιότητα background για τα χαρακτηριστικά φόντου αυτής). Εδώ, πρέπει να παρατηρηθεί ότι αν δεν καθοριστεί κάποια επιμέρους ιδιότητα, όταν γίνεται χρήση της συγκεντρωτικής ιδιότητας border, ο εκάστοτε πλοηγός θεωρεί την προεπιλεγμένη τιμή για αυτή και όχι πως δεν ορίστηκε. Κάθε μία από τις ιδιότητες που αφορούν το περίγραμμα και αναφέρθηκαν πιο πάνω, αποτελείται από τέσσερις επιμέρους ιδιότητες, οι οποίες επιτρέπουν να οριστούν διαφορετικές τιμές για την ιδιότητα αυτή όσον αφορά το πάνω/δεξί/κάτω/αριστερό περίγραμμα.
- Άλλες χρήσιμες ιδιότητες: Πέρα από τις ιδιότητες που αναπτύχθηκαν παραπάνω κάποιες άλλες χρήσιμες ιδιότητες που συναντώνται στη γλώσσα CSS αναλύονται στη συνέχεια:
 - ο padding: Ορίζει το κενό που θα υπάρχει μεταξύ των ορίων ενός HTML στοιχείου και των περιεχομένων του. Είναι πολύ σημαντικό να μην παραλείπεται αυτή η ιδιότητα σε στοιχεία στα οποία έχει οριστεί κάποιο περίγραμμα, ώστε να μην κολλάνε τα περιεχόμενα τους με το περίγραμμα αυτό, κάτι που φαίνεται ιδιαίτερα άσχημο και ερασιτεχνικό. Επίσης, καλό είναι να γίνεται συχνή χρήση της τρέχουσας ιδιότητας, γιατί όσο πιο πολύ χρησιμοποιείται (μέχρι κάποιων ορίων φυσικά), τόσο περισσότερο επαγγελματική εμφάνιση παρουσιάζουν οι Ιστοσελίδες. Για να επιτευχθεί διαφορετικό padding, ανά πλευρά μίας Ιστοσελίδας, είναι εφικτό, ομοίως με την ιδιότητα border είτε να εφαρμοστούν οι ιδιότητες padding-top, padding-right, padding-bottom, padding-left, είτε να εισαχθούν και οι τέσσερις τιμές στην ιδιότητα padding με τη σειρά [top] [right] [bottom] [left] ή [top,bottom] [right,left].
 - ο margin: Πρόκειται ουσιαστικά για το αντίθετο του padding. Ορίζει τον χώρο μεταξύ των ορίων ενός HTML στοιχείου και όσων το περιβάλλουν. Είναι ιδιαίτερα σημαντικό να ορίζεται margin σε εικόνες, ώστε να έχουν απόσταση από τα περιεχόμενα τους, μίας και είναι ιδιαίτερα αντιαισθητικό να κολλάνε με το εκάστοτε κείμενο που τυχόν τις συνοδεύει σε μία Ιστοσελίδα. Φυσικά και εδώ είναι δυνατή η χρήση είτε των ιδιοτήτων margin-top, margin-right, margin-bottom, margin-left, με στόχο να καθοριστούν διαφορετικές τιμές margin ανά πλευρά, είτε να τοποθετηθούν οι τέσσερις διαφορετικές τιμές margin, είτε με τη σειρά [top], [right], [bottom], [left], είτε ως ζευγή [top,bottom] [right,left].
 - ο width, height: Καθορίζουν προφανώς το πλάτος και το ύψος ενός HTML στοιχείου, είτε σε ποσοστό επί τοις εκατό (%), το οποίο υπολογίζεται βάσει του εκάστοτε στοιχείου που το περιέχει, είτε σε κάποια μονάδα μέτρησης από αυτές που χρησιμοποιεί η γλώσσα CSS (τις περισσότερες φορές σε pixels). Σε αυτό το σημείο, ας σημειωθεί πως

υπάρχουν και οι ιδιαίτερα βολικές ιδιότητες min-width, min-height, max-width, max-height, που ορίζουν τα όρια στα οποία μπορούν να κινούνται οι διαστάσεις ενός στοιχείου, εφόσον δε είναι επιθυμητό να οριστούν συγκεκριμένες διαστάσεις.

- o float: Είναι ιδιόμορφη σημασία ιδιότητα, όσον αφορά την εμφάνιση μίας Ιστοσελίδας. Δέχεται τις τιμές right, left και none και επιτρέπει τα στοιχεία που περιβάλλουν ένα HTML στοιχείο στο οποίο εφαρμόζεται η ιδιότητα αυτή να ρέουν τριγύρω του. Μία συνηθισμένη χρήση της παρούσας ιδιότητας είναι στις εικόνες που συνοδεύουν ένα άρθρο.

[36] [39]

2.11.4 Οι Μέθοδοι Εισαγωγής της CSS

Η εισαγωγή της γλώσσας CSS στις Ιστοσελίδες επιτυγχάνεται με τρεις διαφορετικούς τρόπους, ανάλογα την εκάστοτε περίπτωση. Οι τρόποι αυτοί, κατά σειρά φθίνουσα προτεραιότητας, είναι οι ακόλουθοι:

- CSS εντός μίας γραμμής (Inline CSS): Αν επιθυμείται η μορφοποίηση ενός και μόνο HTML στοιχείου, και δεν πρόκειται να χρειαστεί το συγκεκριμένο είδος μορφοποίησης για τίποτα άλλο σε μία Ιστοσελίδα, είναι εφικτό να γραφούν «χύμα» CSS ιδιότητες, δηλαδή χωρίς το κομμάτι του CSS επιλογή και δίχως άγκιστρα, μέσα στην ετικέτα style του εκάστοτε HTML στοιχείου που πρόκειται να μορφοποιηθεί. Αξίζει να σημειωθεί ότι, αν κάποιες από τις ιδιότητες που γραφούν εντός της ετικέτας style ενός στοιχείου, έρχονται σε σύγκρουση με κανόνες της γλώσσας CSS που έχουν οριστεί για αυτό το στοιχείο και σε άλλο σημείο του κώδικα της εκάστοτε Σελίδας που το περιέχει και το αφορούν, τότε ο εκάστοτε πλοηγός, αναφορικά με το παρόν στοιχείο, θα επιλέξει να εφαρμόσει αυτά που γράφηκαν εντός της ετικέτας style.
- CSS εντός μίας Ιστοσελίδας (Internal CSS): Πολλές φορές, είναι σκόπιμο να εφαρμοστούν κάποιοι CSS κανόνες αποκλειστικά για μία Ιστοσελίδα ενός Ιστοτόπου (ο οποίος αποτελείται από ένα σύνολο επιμέρους Σελίδων) και όχι για όλον τον Ιστοχώρο ή να διαφοροποιηθούν ορισμένοι CSS κανόνες μόνο ως προς μερικές ιδιότητές τους και όχι στο σύνολό τους. Προκειμένου να επιτευχθεί κάτι τέτοιο, δίχως τη δημιουργία ξεχωριστού αρχείου CSS κάθε φορά, απαιτείται η εισαγωγή CSS κανόνων εντός της ετικέτας style, η οποία με τη σειρά της πρέπει να είναι τοποθετημένη μέσα στην ετικέτα head, ενός HTML εγγράφου και η ταυτόχρονη απόδοση στην ιδιότητα type της ετικέτας style της τιμή text/css, όπως φαίνεται στην κατώτερη σύνταξη:

```
<head>  
<style type="text/css">  
.....  
</style>  
</head>
```

- Εξωτερικό CSS αρχείο ("External CSS"): Η πιο δόκιμη λύση για την εισαγωγή της γλώσσας CSS στον HTML κώδικα ενός Ιστοτόπου, την οποία οφείλουν να εκμεταλλεύονται σε πρώτη προτεραιότητα οι προγραμματιστές του Διαδικτύου, είναι η χρήση εξωτερικών CSS αρχείων. Η λύση αυτή υλοποιείται με τη συγγραφή CSS κανόνων σε ένα εξωτερικό (έξω από τον HTML κώδικα του εκάστοτε Ιστοχώρου) αρχείου με επέκταση .css (πχ main.css) και τη σύνδεση του αρχείου αυτού με τον κύριο HTML κώδικα του εκάστοτε Ιστοτόπου, με την τοποθέτηση στην ετικέτα head κάθε ξεχωριστής Ιστοσελίδας αυτού του Τύπου, που συγκροτεί τον εκάστοτε Ιστοτόπο, του παρακάτω αποσπάσματος κώδικα:

```
<link href="externalFileName.css" type="text/css"/>
```

Όπου externalFileName το όνομα του εκάστοτε εξωτερικού αρχείου που περιέχει τους CSS κανόνες μορφοποίησης της εκάστοτε HTML Ιστοσελίδας.

Εδώ, να σημειωθεί πως η τρέχουσα μεθοδολογία είναι η πιο χρήσιμη, αφού προσφέρει τη δυνατότητα εκμετάλλευσης των ίδιων CSS κανόνων σε κάθε μία από τις επιμέρους Σελίδες που

απαρτίζουν έναν Ιστότοπο, χωρίς να χρειάζεται να γίνεται αντιγραφή-επικόλληση των κανόνων αυτών χωριστά σε καθεμία από τις προαναφερόμενες Σελίδες.

[38]

2.11.5 Η Προτεραιότητα της CSS

Πολλές φορές για το ίδιο κάθε φορά στοιχείο μίας Ιστοσελίδας μπορεί να ταιριάζουν αρκετοί CSS κανόνες μορφοποίησης, ορισμένοι από τους οποίους να περιέχουν ιδιότητες που συγκρούονται μεταξύ τους. Για να αποτρέπεται η δυσάρεστη αυτή κατάσταση, η γλώσσα CSS εισάγει κανονισμούς προτεραιότητας, οι σπουδαιότεροι από τους οποίους αναφέρονται στη συνέχεια:

- Οι CSS κανόνες μορφοποίησης που βρίσκονται μέσα στην HTML ετικέτα `<style>...</style>` έχουν πάντα μεγαλύτερη προτεραιότητα από οποιοσδήποτε άλλες (εκτός από όσες χρησιμοποιούν τη σήμανση “!important”, η οποία θα αναλυθεί πιο κάτω).
- Οι CSS κανόνες μορφοποίησης που βρίσκονται εντός της HTML ετικέτας `<style>...</style>` έχουν μεγαλύτερη προτεραιότητα από αυτούς που βρίσκονται σε ένα εξωτερικό CSS αρχείο.
- Όταν δύο ή περισσότερα εξωτερικά CSS αρχεία συνδέονται με μία Ιστοσελίδα, οι κανόνες μορφοποίησης αυτού του αρχείου που αναφέρεται τελευταίο στην προαναφερόμενη Σελίδα έχουν μεγαλύτερη προτεραιότητα από τους κανόνες των προγενεστέρων του.
- Ένας CSS κανόνας μορφοποίησης έχει μεγαλύτερη προτεραιότητα από όσους CSS κανόνες γράφηκαν πριν από αυτόν, εκτός και αν δεν συντρέχει άλλος λόγος, ώστε να είναι πιο σημαντικός κάποιος από τους προηγούμενους του κανόνες.
- Οι επιλογείς του τύπου «#όνομα_id» έχουν μεγαλύτερη προτεραιότητα από τους επιλογείς του τύπου «.όνομα_κλάσης», οι οποίοι με τη σειρά τους έχουν μεγαλύτερη προτεραιότητα από τους επιλογείς του τύπου «στοιχείο», διότι είναι οι πρώτοι είναι πιο σαφείς από τους δεύτερους και οι δεύτεροι από τους τρίτους.
- Σε κάθε περίπτωση, αν είναι επιθυμητό ένας κανόνας μορφοποίησης να έχει μεγαλύτερη προτεραιότητα από αυτή που του αντιστοιχεί, υπάρχει η δυνατότητα πριν τον τερματισμό του κανόνα αυτού με το σύμβολο του ελληνικού ερωτηματικού (;) να εισαχθεί ο χαρακτηρισμός “!important”. Ωστόσο, γενικά πρέπει να μη πολυχρησιμοποιείται αυτή η τεχνική, καθώς μπορεί να οδηγήσει σε μπερδεμένο CSS κώδικα.

[39]

2.11.6 Η Κληρονομικότητα της CSS

Όπως έχει ήδη αναφερθεί, στη γλώσσα HTML κάθε στοιχείο αποτελεί «απόγονο» (“descendant”) κάποιου άλλου στοιχείου, εκτός από το στοιχείο `<html>` που αποτελεί «πρόγονο» (“ancestor”) κάθε στοιχείου σε μία Ιστοσελίδα. Το κάθε στοιχείο μπορεί να έχει και δικούς του απογόνους, εκτός αν πρόκειται για στοιχεία που δε χρησιμοποιούν ετικέτα κλεισίματος, όπως είναι, για παράδειγμα, το ``. Έτσι, διαμορφώνεται μία ιεραρχία, η οποία αποτελεί το DOM κάθε HTML Ιστοσελίδας.

Όταν εφαρμόζεται μία CSS ιδιότητα σε ένα HTML στοιχείο, μπορεί να κληρονομηθεί και στα στοιχεία-απογόνους του ή μπορεί και να μην κληρονομηθεί, ανάλογα την εκάστοτε ιδιότητα. Όσον αφορά τις ιδιότητες που αναφέρθηκαν παραπάνω, αυτές που κληρονομούνται και στα στοιχεία απογόνους είναι οι εξής:

- color
- font-size
- font-family
- font-style
- font-weight
- text-align
- text-decoration

Φυσικά, αν οριστεί μέσω κάποιου CSS κανόνα μία διαφορετική τιμή για μία συγκεκριμένη ιδιότητα, θα εφαρμοστεί αυτή η τιμή και όχι η κληρονομημένη (inherited). Από τη στιγμή που είναι αρεστό στην ιδιότητα ενός στοιχείου να καταχωρηθεί η τιμή της αντίστοιχης ιδιότητας του «γονέα»

του, η οποία κανονικά δεν κληρονομείται, υπάρχει η ευχέρεια ως τιμή αυτής ιδιότητας για αυτό το στοιχείο να εκχωρηθεί η inherit.

[39]

2.11.7 Σύγκριση της CSS

Είναι ξεκάθαρο πλέον ότι η χρήση της γλώσσας CSS κερδίζει έδαφος στη μορφοποίηση των Ιστοσελίδων έναντι της γλώσσας HTML. Για το λόγο αυτό στην τελευταία έκδοση της HTML κάποια βασικά στοιχεία μορφοποίησης, που μπορούσε κανείς να συναντήσει σε πρωτότερες εκδόσεις της γλώσσας (πχ `<style>...</style>`), παύουν να υφίστανται και έχουν ενσωματωθεί στον κώδικα της CSS. Έτσι, τα βασικότερα πλεονεκτήματα της χρήσης της CSS έναντι της μορφοποίησης μέσω της HTML είναι τα επόμενα:

- Ο κώδικας της γλώσσας CSS είναι πολύ πιο ευέλικτος από εκείνον της γλώσσας HTML, για αυτό και η χρήση της CSS κατέστησε εφικτές μορφοποιήσεις στο περιεχόμενο Ιστοσελίδων οι οποίες ήταν αδύνατες ή πολύ δύσκολες με την κλασική HTML.
- Η χρησιμοποίηση της γλώσσας CSS έκανε ευκολότερη τη συντήρηση των Ιστοσελίδων, από τη στιγμή που η εμφάνιση ενός ολόκληρου Ιστοτόπου μπορεί να ελέγχεται από ένα και μόνο εξωτερικό CSS αρχείο. Οπότε, κάθε αλλαγή στο στυλ ενός Ιστοχώρου μπορεί να γίνεται με μία μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων στον HTML κώδικά του.
- Ένα HTML αρχείο που έχει μορφοποιηθεί με τη γλώσσα CSS αντί για τη γλώσσα HTML, διαθέτει μικρότερο μέγεθος, επειδή ο κάθε κανόνας μορφοποίησης γράφεται μόνο μία φορά και όχι σε κάθε σημείο μίας Ιστοσελίδας που εφαρμόζεται.
- Η μεταχείριση της γλώσσας CSS σε αντίθεση με τη γλώσσα HTML παρέχει καλύτερη SEO. Δηλαδή οι μηχανές αναζήτησης δεν μπερδεύονται ανάμεσα σε HTML περιεχόμενο και τη μορφοποίησή του, αλλά έχουν πρόσβαση σε αυτό σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν (indexing).
- Η εισαγωγή της CSS τεχνικής έκανε τους Ιστοχώρους γρηγορότερους, καθώς, από τη στιγμή που διατίθεται ένα εξωτερικό CSS αρχείο, ο εκάστοτε περιηγητής την πρώτη φορά που θα φορτώσει κάποια Ιστοσελίδα ενός Ιστοτόπου αποθηκεύει το αρχείο αυτό στην κρυφή του μνήμη, οπότε δεν χρειάζεται να το φορτώσει πάλι, κάθε φορά που ένας χρήστης κατεβάζει κάποια άλλη Σελίδα του προαναφερόμενου Τύπου.

[40]

2.12 Η Γλώσσα Σεναρίων, JavaScript

2.12.1 Εισαγωγή στην JavaScript

Η JavaScript είναι μία συμπαγής γλώσσα σεναρίων (scripting language), η οποία βοηθά τον εκάστοτε προγραμματιστή του Παγκοσμίου Ιστού στην κατασκευή αλληλεπιδραστικών και λειτουργικών Ιστοτόπων. Οι εφαρμογές αυτές μπορούν να εκτελούνται είτε στο πρόγραμμα περιήγησης του εκάστοτε χρήστη (client side JavaScript), είτε σε ένα διακομιστή Ιστού (server side JavaScript). Για να είναι ικανό ένα πρόγραμμα περιήγησης να εκτελέσει ένα σενάριο γραμμένο σε JavaScript, πρέπει να διαθέτει έναν ερμηνευτή της JavaScript (JavaScript interpreter). Ένα σενάριο της JavaScript ενσωματώνεται στον HTML κώδικα της εκάστοτε Ιστοσελίδας.

Μία συνηθισμένη ερώτηση, που γίνεται από νέους κυρίως προγραμματιστές, είναι ποια είναι η διαφορά μεταξύ ενός σεναρίου (script) και ενός προγράμματος (program). Η απάντηση είναι η εξής: Ένα πρόγραμμα που δημιουργείται με μία κανονική γλώσσα προγραμματισμού (πχ Pascal, Basic, C++, Java) ονομάζεται πηγαίος κώδικας (source code). Το πρόγραμμα αυτό μεταγλωττίζεται (compile) και στη συνέχεια εκτελείται σε δυαδική μορφή από τον επεξεργαστή του εκάστοτε υπολογιστή. Αντίθετα, οι εντολές που περιλαμβάνει ένα σενάριο γράφονται με έναν απλό συντάκτη κειμένου (text editor), και, αντί να περνούν κατευθείαν στον επεξεργαστή του εκάστοτε υπολογιστή για εκτέλεση, την εκτέλεσή τους αναλαμβάνει ένα άλλο πρόγραμμα, που στην περίπτωση της JavaScript είναι το εκάστοτε πρόγραμμα περιήγησης στο Διαδίκτυο. Καταλαβαίνει κανείς ότι το να γράψει και να εκτελέσει ένα σενάριο

είναι πολύ πιο εύκολη υπόθεση από το να γράψει και να εκτελέσει ένα πρόγραμμα, αν και η εκτέλεση ενός σεναρίου είναι πιο αργή από την εκτέλεση ενός προγράμματος.

[36] [41]

2.12.2 Οι Μέθοδοι Εισαγωγής της JavaScript

Σε ένα HTML έγγραφο ο JavaScript κώδικας τοποθετείται μεταξύ του ζεύγους ετικετών `<script>` `</script>`. Παλιότερες εκδόσεις της γλώσσας HTML, προκειμένου να εισάγουν JavaScript κώδικα στα έγγραφά τους, χρησιμοποιούσαν το χαρακτηριστικό γνώρισμα `type` της ετικέτας `script`, σύμφωνα με τη σύνταξη:

```
<script type="text/javascript">  
.....  
</script>
```

Πλέον, όμως, η προαναφερόμενη ιδιότητα `type` δεν είναι υποχρεωτική, εφόσον η γλώσσα JavaScript είναι η προκαθορισμένη γλώσσα σεναρίων για τη γλώσσα HTML.

Είναι δυνατό να υπάρχουν περισσότερα από ένα JavaScript σενάρια σε ένα HTML έγγραφο. Όσον αφορά, τώρα, την τοποθέτηση του JavaScript κώδικα εντός ενός HTML εγγράφου υπάρχουν οι ακόλουθες δυνατότητες:

- Ανάμεσα στο ζεύγος ετικετών `<head>` `</head>`.
- Ανάμεσα στο ζεύγος ετικετών `<body>` `</body>`.
- Ανάμεσα τόσο στο ζεύγος ετικετών `<head>` `</head>` όσο και στο ζεύγος ετικετών `<body>` `</body>`, ταυτόχρονα, αν και αυτή η συνήθεια καλύτερα να αποφεύγεται.
- Σε εξωτερικά JavaScript αρχεία.

Στο σημείο τούτο, να τονιστεί πως τα JavaScript αρχεία έχουν την επέκταση `.js`. Προκειμένου να γίνει χρήση ενός εξωτερικού JavaScript αρχείου, τοποθετείται το όνομα του εν λόγω αρχείου που περιέχει το εκάστοτε JavaScript σενάριο στην ιδιότητα `src` (`source`) της ετικέτας `<script>` σύμφωνα με τη σύνταξη:

```
<script src="externalJavaScriptFileName.js"></script>
```

Όπου `externalJavaScriptFileName` το όνομα του εκάστοτε εξωτερικού αρχείου που περιέχει τον αρμόζων σε κάθε περίπτωση JavaScript κώδικα.

Εννοείται πως είναι εφικτή η αναφορά σε ένα εξωτερικό JavaScript σενάριο είτε μεταξύ του ζεύγος ετικετών `<head>` `</head>`, είτε μεταξύ του ζεύγος ετικετών `<body>` `</body>`, είτε μεταξύ και των δύο παραπάνω ειδών ζευγών, παράλληλα, αν και η τελευταία πρακτική είναι προτιμότερο να αποφεύγεται. Και στις τρεις προηγούμενες περιπτώσεις το εκάστοτε JavaScript σενάριο συμπεριφέρεται σα να είναι τοποθετημένο εκεί ακριβώς όπου το ζεύγος ετικετών `<script>` `</script>` είναι τοποθετημένο. Προσοχή, όμως: τα εξωτερικά σενάρια δεν επιτρέπεται να περιέχουν το ζεύγος ετικετών `<script>` `</script>`!!!

Κλείνοντας, να αναφερθεί ότι η εκμετάλλευση εξωτερικών JavaScript σεναρίων είναι μία σωστή και χρήσιμη πρακτική, όταν το ίδιο απόσπασμα κώδικα διατίθεται σε πολλές διαφορετικές Ιστοσελίδες και, μάλιστα, προσφέρει αρκετά πλεονεκτήματα, τα σημαντικότερα από τα οποία είναι τα ακόλουθα:

- Διακρίνει τον εκάστοτε HTML κώδικα από τον εκάστοτε JavaScript κώδικα κάνοντας με αυτό τον τρόπο και τους δύο συγκεκριμένους κώδικες πιο εύκολα αναγνώσιμους και συντηρήσιμους.
- Επιταχύνει την φόρτωση των υφιστάμενων κάθε φορά Ιστοσελίδων.

[41] [42]

2.12.3 Οι Βασικοί Συντακτικοί Κανόνες της JavaScript

Οι βασικοί κανόνες σύνταξης που διέπουν τη γλώσσα JavaScript είναι οι ακόλουθοι:

- Η JavaScript είναι γλώσσα που κάνει διάκριση ανάμεσα στα πεζά (μικρά) και τα κεφαλαία γράμματα, δηλαδή είναι `case sensitive Language`, γεγονός που σηματοδοτεί ότι οι λέξεις της

γλώσσας (πχ μεταβλητές, εντολές) με κεφαλαία είναι διαφορετικές από τις ίδιες λέξεις με μικρά.

- Τα σύμβολα που απαντώνται στη γλώσσα JavaScript (πχ παρενθέσεις, άγκιστρα, εισαγωγικά) σε κάθε περίπτωση τοποθετούνται ανά ζεύγη.
- Κάθε μία από τις εντολές ή τις προτάσεις της γλώσσας JavaScript ακολουθείται από το σύμβολο του ελληνικού ερωτηματικού (;), όπως γίνεται, για παράδειγμα, και στη γλώσσα C++ ή την Java. Εντούτοις, η JavaScript δεν είναι και τόσο αυστηρή στο θέμα αυτό. Πιο ειδικά, αν μία εντολή ή δήλωση βρίσκεται μόνη της στην ίδια γραμμή του εκάστοτε JavaScript κώδικα, μπορεί να παραλειφθεί το ; στο τέλος αυτής της γραμμής. Αν, όμως, μία εντολή ή δήλωση ακολουθείται και από άλλη στην ίδια γραμμή του εκάστοτε JavaScript κώδικα, τότε το ; πρέπει να προστεθεί υποχρεωτικά στο τέλος της συγκεκριμένης γραμμής. Καλό είναι, πάντως, να μην παραλείπεται το συγκεκριμένο σύμβολο στο τέλος μίας εντολής ή μίας πρότασης εντός του εκάστοτε JavaScript κώδικα .
- Στα JavaScript σενάρια, πολλές φορές χρησιμοποιούνται προσδιοριστές (identifiers). Οι προσδιοριστές είναι ονόματα που χρησιμοποιούν οι προγραμματιστές, ώστε να ονομάσουν μεταβλητές, κλάσεις, μεθόδους, μεταβλητές και άλλες σχετικές οντότητες της γλώσσας JavaScript. Οι προσδιοριστές έχουν συγκεκριμένους κανόνες ονομασίας. Έτσι, το όνομα ενός προσδιοριστή στη JavaScript είναι μία απεριόριστου μήκους ακολουθία χαρακτήρων του συστήματος Unicode, η οποία, όμως, πρέπει να ξεκινά είτε από γράμμα, είτε από το χαρακτήρα του δολαρίου (\$), είτε από το χαρακτήρα της υπογράμμισης (_) (παρόλο που η χρήση του \$ ή της _ δεν προτείνεται), και να μην περιέχει κενά. Δεν πρέπει σε καμία περίπτωση να αρχίζει με αριθμό ή κάποιον άλλο μη αλφαβητικό χαρακτήρα, εκτός του \$ ή της _, ούτε να ισοδυναμεί με κάποιες από τις δεσμευμένες λέξεις της JavaScript (που θα αναφερθούν στη συνέχεια). Ύστερα από τον πρώτο χαρακτήρα, οι άλλοι μπορεί να είναι είτε γράμματα, είτε αριθμοί, είτε \$ είτε _. Επίσης, αξίζει να σημειωθεί πως το όνομα ενός προσδιοριστή είναι χρήσιμο να συγκροτείται από ολόκληρες λέξεις και όχι από κρυπτογραφημένες συντομογραφίες. Κλείνοντας, να παρατηρηθεί ότι, αν το όνομα ενός προσδιοριστή αποτελείται από μία και μόνο λέξη, τότε καλό είναι η λέξη αυτή να γράφεται με μικρά γράμματα, ενώ, αν συνίσταται από περισσότερες από μία λέξεις, τότε είναι βολικό κάθε μία από αυτές τις λέξεις, πέρα της πρώτης, να αρχίζει με κεφαλαίο γράμμα.
- Η γλώσσα JavaScript εκμεταλλεύεται μία ομάδα από λέξεις για συγκεκριμένους σκοπούς (πχ τύπων δεδομένων, ονόματα εντολών, μεθόδων). Οι λέξεις αυτές ονομάζονται δεσμευμένες (reserved words) ή λέξεις-κλειδιά (keywords) και δεν πρέπει να χρησιμοποιούνται από τους προγραμματιστές για άλλους σκοπούς. Οι δεσμευμένες λέξεις της JavaScript είναι οι επόμενες: abstract, arguments, boolean, break, byte, case, catch, char, class, const, continue, debugger, default, delete, do, double, else, enum, eval, export, extends, false, final, finally, float, for, function, goto, if, implements, import, in, instanceof, int, interface, let, long, native, new, null, package, private, protected, public, return, short, static, super, switch, synchronized, this, throw, throws, transient, true, try, typeof, var, void, volatile, while, with, yield.
- Μία λέξη ή φράση που είναι ανάγκη να τυπωθεί όπως είναι μέσα στον JavaScript κώδικα της εκάστοτε Ιστοσελίδας, τοποθετείται εντός διπλών εισαγωγικών ("").
- Σε πολλά σημεία ενός JavaScript σεναρίου επιβάλλεται να εισαχθούν σχόλια. Τα σχόλια (comments) είναι κείμενο που δεν εκτελείται από τον εκάστοτε υπολογιστή, αλλά χρειάζεται να βρίσκεται σε διάφορα σημεία ενός σεναρίου, ώστε να θυμίζει στον εκάστοτε προγραμματιστή τη λειτουργία διαφόρων τμημάτων του εκάστοτε JavaScript κώδικα, ώστε να είναι εύκολη η συντήρηση ή η τροποποίησή τους, αν χρειαστεί, τόσο από τον ίδιο όσο και από άλλους προγραμματιστές. Τα σχόλια στη γλώσσα JavaScript καταχωρούνται μετά από τους χαρακτήρες //, αν εκτείνονται σε μία σειρά, ή ανάμεσα στα σύμβολα /* και */, αν επεκτείνονται σε περισσότερες από μία σειρές.

[41] [46]

2.12.4 Οι Μεταβλητές, τα Δεδομένα και οι Τελεστές της JavaScript

2.12.4.1 Οι Μεταβλητές

Οι μεταβλητές (variables) είναι ποσότητες μνήμης στις οποίες έχει δοθεί ένα όνομα και χρησιμοποιούνται για τη φύλαξη τιμών (πχ αριθμητικών, αλφαριθμητικών, λογικών ή άλλου είδους). Η τιμή μίας μεταβλητής είναι δυνατό να αλλάξει κατά τη διάρκεια της εκτέλεσης ενός προγράμματος. Οι μεταβλητές χρησιμοποιούνται από όλες τις γλώσσες προγραμματισμού και προσδίδουν μεγάλη ευελιξία στα προγράμματα. Το όνομα μίας μεταβλητής της γλώσσας JavaScript πρέπει να υπακούει ακριβώς στους ίδιους κανόνες που υπακούνε και οι προσδιοριστές της συγκεκριμένης γλώσσας.

Μία μεταβλητή, τώρα, στη γλώσσα JavaScript συντάσσεται με τη χρήση της δεσμευμένης λέξης var, όπως παρακάτω:

```
var variableName;
```

Όπου variableName το όνομα της εκάστοτε JavaScript μεταβλητής ενός JavaScript κώδικα.

Σχετικά, δε, με την εμβέλεια των μεταβλητών της γλώσσας JavaScript σε γενικές γραμμές ισχύουν τα εξής:

- Καθολικές (Global): Ορίζονται έξω από τις συναρτήσεις (που θα παρουσιαστούν παρακάτω) του JavaScript κώδικα ενός HTML εγγράφου, και αναγνωρίζονται σε όλα τα σημεία αυτού του εγγράφου ακόμα και μέσα στις ίδιες τις συναρτήσεις. Μία καθολική μεταβλητή μπορεί να οριστεί με ή χωρίς τη χρήση της δεσμευμένης λέξης var.
- Τοπικές (Local): Ισχύουν μόνο μέσα στη συναρτήσεις στις οποίες ορίστηκαν. Όταν οι συναρτήσεις αυτές ολοκληρωθούν, όλες οι τοπικές μεταβλητές που υπήρχαν μέσα τους παύουν να υφίστανται. Αν μία από τις προαναφερόμενες συναρτήσεις ξανακληθεί, δημιουργούνται νέες τοπικές μεταβλητές. Η δήλωση μίας τοπικής μεταβλητής μέσα σε μία συνάρτηση γίνεται με τη χρήση της δεσμευμένης λέξης var. Αν δε χρησιμοποιηθεί η λέξη var στη δήλωση μίας τοπικής μεταβλητής, τότε η μεταβλητή αυτή θεωρείται καθολική. Είναι επιθυμητό να προσεχθεί πως, αν οριστεί μία τοπική μεταβλητή μέσα σε μία συνάρτηση, χρησιμοποιώντας το ίδιο όνομα με μία καθολική μεταβλητή, τότε μέσα στην εν λόγω συνάρτηση ισχύει η προαναφερόμενη τοπική μεταβλητή. Επίσης, αν μία τοπική μεταβλητή αλλάξει τιμή εντός μίας συνάρτησης, κάτι τέτοιο δεν θα επηρεάσει την τιμή μίας καθολικής μεταβλητής που τυγχάνει να φέρει το ίδιο όνομα με την προαναφερόμενη τοπική μεταβλητή.

[41] [42]

2.12.4.2 Τα Δεδομένα

Ένα από το πιο κύρια χαρακτηριστικά μίας γλώσσας προγραμματισμού είναι το σύνολο των τύπων δεδομένων που αυτή υποστηρίζει. Το σύνολο αυτό έχει να κάνει με τον τύπο των τιμών που είναι σε θέση να αναπαριστά και να διαχειρίζεται η εκάστοτε γλώσσα. Συγκεκριμένα, τώρα, η γλώσσα JavaScript επιτρέπει την εργασία με τους ακόλουθους τύπους δεδομένων

- Τα αριθμητικά δεδομένα ή οι αριθμοί (numbers): Αφορούν αριθμούς οι οποίοι λαμβάνουν χώρα σε αριθμητικές παραστάσεις ή χρησιμοποιούνται για την εκτέλεση αριθμητικών πράξεων. Τα νούμερα που εκμεταλλεύεται η γλώσσα JavaScript αποθηκεύονται όλα σύμφωνα με το πρότυπο IEEE (Institute of Electrical and Electronic Engineers) 754 ως 64-bits αριθμοί κινητής υποδιαστολής διπλής ακρίβειας (double), δίχως να διαχωρίζονται σε επιμέρους κατηγορίες, όπως συμβαίνει με τις περισσότερες γλώσσες προγραμματισμού. Μάλιστα, σύμφωνα με το συγκεκριμένο πρότυπο, ο αριθμός (ως κλάσμα) καταχωρείται στα bits 0-51, ο εκθέτης του στα bits 52-62 και το πρόσημό του στο bit 63 και για την αναπαράστασή τους γίνεται χρήση κυρίως του δεκαδικού αριθμητικού συστήματος, αλλά σπανιότερα και του οκταδικού συστήματος, καθώς, επίσης, και του δεκαεξαδικού.
- Οι λογικές τιμές (boolean): Είναι δύο ειδών: η αληθής (true) τιμή ή 1 και η ψευδής (false) τιμή ή 0. Οι λογικές τιμές χρησιμεύουν κατά την επεξεργασία λογικών προτάσεων και διατίθενται δίχως τη χρήση εισαγωγικών.
- Τα αλφαριθμητικά δεδομένα ή οι συμβολοσειρές (strings): Πρόκειται για λέξεις ή ομάδες χαρακτήρων που μπορούν να περιλαμβάνουν γράμματα και αριθμούς ή άλλα σύμβολα. Κάθε αλφαριθμητικό δεδομένο τοποθετείται ανάμεσα σε διπλά (") ή απλά (') εισαγωγικά. Αν μεταξύ των εισαγωγικών δεν υπάρχει κανένας χαρακτήρας, τότε το αλφαριθμητικό ονομάζεται κενό.

- Οι πίνακες (arrays): Δηλώνονται με ένα ζεύγος αγκυλών [] ανάμεσα στις οποίες περιλαμβάνονται τα στοιχεία του εκάστοτε πίνακα, τα οποία διαχωρίζονται μεταξύ τους με κόμμα (,).
- Τα αντικείμενα (objects): Ορίζονται με ένα ζεύγος αγκίστρων {} ανάμεσα στα οποία γράφονται οι ιδιότητες των εκάστοτε αντικειμένων με τη μορφή όνομα-τιμή και χωρισμένες μεταξύ τους με κόμμα (,).

Στο σημείο τούτο, κρίνεται απαραίτητο να τονιστεί πως και οι τέσσερις προηγούμενοι βασικοί τύποι δεδομένων αποτελούν αντικείμενα (τα οποία θα μελετηθούν παρακάτω) της γλώσσας JavaScript. Με άλλα λόγια, οι αριθμοί είναι αντικείμενα τύπου Number, οι λογικές τιμές είναι αντικείμενα τύπου Boolean, τα αλφαριθμητικά ή οι συμβολοσειρές είναι αντικείμενα τύπου String και οι πίνακες είναι αντικείμενα τύπου Array.

Επιπρόσθετα, είναι επιθυμητό να αναφερθεί ότι, εκτός από τους προαναφερόμενους τύπους δεδομένων, η γλώσσα JavaScript εκμεταλλεύεται κάποιους επιπρόσθετους τύπους δεδομένων, που είναι οι επόμενοι:

- NaN: Σημαίνει “Not a Number” («Όχι ένας Αριθμός») και επιστρέφεται, εφόσον το αποτέλεσμα μίας αριθμητικής πράξης σε ένα JavaScript κώδικα δεν ορίζεται, όπως είναι, για παράδειγμα, η διαίρεση με το μηδέν.
- undefined: Επιστρέφεται, αφότου σε ένα JavaScript κώδικα χρησιμοποιείται μία μεταβλητή που δεν υπάρχει, ή που υπάρχει, αλλά δεν της έχει καταχωρηθεί κάποια τιμή.
- Infinity και -Infinity: Επιστρέφεται, καθώς κατά την εκτέλεση πράξεων μέσα σε ένα JavaScript κώδικα κάποια τιμή ξεπεράσει το άνω ή το κάτω επιτρεπτό όριο.
- null: Δηλώνει απουσία τιμής σε ένα JavaScript κώδικα, δηλαδή τίποτα. Δεν πρέπει να συγχέεται με το μηδέν ή το κενό διάστημα.

Αξίζει στο σημείο αυτό, να παρατηρηθεί ότι όλοι τύποι δεδομένων NaN, undefined, Infinity, -Infinity αποτελούν καθολικές\εσωτερικές μεταβλητές (global\internal variables) της γλώσσας JavaScript.

[41] [42]

2.12.4.3 Οι Τελεστές

Πολλές φορές χρειάζεται να συνδυαστούν τα δεδομένα ενός JavaScript κώδικα με κάποιους τρόπους, ώστε να προκύψουν καινούργια αποτελέσματα. Για το σκοπό αυτό χρησιμοποιούνται παραστάσεις (μαθηματικές ή άλλης μορφής) και μέσα σε αυτές συνδέονται τα δεδομένα με τελεστές. Οι σημαντικότεροι τελεστές που απαντώνται στη γλώσσα JavaScript παρουσιάζονται αναλυτικά στη συνέχεια.

Πρώτον, ο μοναδιαίος τελεστής αύξησης (unary increment operator) (++) και ο μοναδιαίος τελεστής μείωσης (unary decrement operator) (--), οι οποίοι μεταβάλλουν, αυξάνοντας ή μειώνοντας, αντίστοιχα, την τιμή ενός τελεστέου κατά μία μονάδα. Οι συγκεκριμένοι τελεστές μπορούν να μπουν είτε ως πρόθεμα (prefix) είτε ως επίθεμα (postfix) σε έναν τελεστή. Ανάλογα, λοιπόν, με την τοποθέτησή τους, μπροστά ή πίσω από έναν τελεστή, η δράση τους διαφοροποιείται ως εξής:

- Αν τοποθετηθεί ο τελεστής ++ πριν από ένα τελεστέο, τότε αρχικά αυξάνεται η τιμή του συγκεκριμένου τελεστέου κατά ένα και μετά καταχωρείται η τιμή του σε έναν άλλο τελεστέο.
- Αν τοποθετηθεί ο τελεστής ++ μετά από ένα τελεστέο, τότε αρχικά καταχωρείται η τιμή του συγκεκριμένου τελεστέου σε έναν δεύτερο τελεστέο και μετά αυξάνεται η τιμή του δεύτερου τελεστέου κατά ένα.
- Αν τοποθετηθεί ο τελεστής -- πριν από ένα τελεστέο, τότε αρχικά μειώνεται η τιμή αυτού του τελεστέου κατά ένα και μετά καταχωρείται η τιμή του σε έναν άλλο τελεστέο.
- Αν τοποθετηθεί ο τελεστής -- μετά από ένα τελεστέο, τότε αρχικά καταχωρείται η τιμή του συγκεκριμένου τελεστέου σε έναν δεύτερο τελεστέο και μετά μειώνεται η τιμή του δεύτερου τελεστέου κατά ένα.

Σε αυτό το σημείο, να γίνει μνεία πως οι μοναδιαίοι τελεστές αύξησης ή μείωσης δρουν αποκλειστικά σε τελεστέους που παριστάνουν αριθμητικά δεδομένα.

Δεύτερον, οι αριθμητικοί τελεστές (arithmetic operators), που χρησιμοποιούνται για τις αριθμητικές πράξεις και περιγράφονται στον κάτωθι Πίνακα:

Τελεστής	Ονομασία	Σημασία
----------	----------	---------

+	Συν (Plus)	Προσθέτει δύο τελεστέους.
-	Μείον (Minus)	Αφαιρεί δύο τελεστέους.
*	Επί (For)	Πολλαπλασιάζει δύο τελεστέους.
/	Διά (By)	Διαιρεί δύο τελεστέους.
%	Ακέραιο Διά (Modulo)	Υπολογίζει το ακέραιο ηλίκο της διαίρεσης δύο τελεστέων (ακεραίων αριθμών).

Πίνακας 5: Οι αριθμητικοί τελεστές της JavaScript

Από πλευράς προτεραιότητας πράξεων, οι τελεστές *, /, % είναι ισοδύναμοι μεταξύ τους και έχουν υψηλότερη προτεραιότητα από τους τελεστές + και -, οι οποίοι είναι, επίσης, ισάξιοι μεταξύ τους. Αν σε μία παράσταση απαντώνται αριθμητικοί τελεστές με την ίδια προτεραιότητα, τότε οι πράξεις εκτελούνται από τα αριστερά προς τα δεξιά με τη σειρά που σημειώνονται. Επιπλέον, αν υπάρχουν παρενθέσεις, τότε γίνονται πρώτα οι πράξεις μέσα σε αυτές. Ακόμα, κατά την κωδικοποίηση σε JavaScript (τόσο αριθμητικών όσο και αλγεβρικών) παραστάσεων πιθανά άγκιστρα ή αγκύλες μεταφράζονται σε παρενθέσεις (οι παρενθέσεις, φυσικά, παραμένουν παρενθέσεις).

Κάπου εδώ, πρέπει να τονιστεί το γεγονός ότι οι αριθμητικοί τελεστές εφαρμόζονται σε τελεστέους που αναπαριστούν αριθμητικά δεδομένα. Ωστόσο, ο τελεστής + δρα είτε οι τελεστέοι είναι αριθμοί, είτε αλφαριθμητικά (οπότε και τα συνενώνει μεταξύ τους), είτε συνδυασμοί αριθμών και αλφαριθμητικών (οπότε και συνενώνει τους αριθμούς με τα αλφαριθμητικά).

Τρίτον, οι τελεστές εκχώρησης (assignment operators), οι οποίοι εξετάζονται στο μετέπειτα Πίνακα:

Τελεστής	Ονομασία	Σημασία
=	Απλή Καταχώρηση (Simple Assignment)	Καταχωρεί την τιμή του δεξιού τελεστέου μίας ισότητας στον αριστερό τελεστέο αυτής της ισότητας.
+=	Συν ΚΑΙ Καταχώρηση (Plus AND Assignment)	Προσθέτει το δεξιό τελεστέο μίας ισότητας στον αριστερό τελεστέο αυτής της ισότητας και καταχωρεί το αποτέλεσμα στον αριστερό τελεστέο της εν λόγω ισότητας.
-=	Μείον ΚΑΙ Καταχώρηση (Minus AND Assignment)	Αφαιρεί τον δεξιό τελεστέο μίας ισότητας από τον αριστερό τελεστέο αυτής της ισότητας και καταχωρεί το αποτέλεσμα στον αριστερό τελεστέο της εν λόγω ισότητας.
*=	Επί ΚΑΙ Καταχώρηση (For AND Assignment)	Πολλαπλασιάζει το δεξιό τελεστέο μίας ισότητας με τον αριστερό τελεστέο αυτής της ισότητας και καταχωρεί το αποτέλεσμα στον αριστερό τελεστέο της εν λόγω ισότητας.
/=	Διά ΚΑΙ Καταχώρηση (By AND Assignment)	Εκτελεί τη διαίρεση του αριστερού τελεστέου μίας ισότητας με το δεξιό τελεστέο αυτής της ισότητας και καταχωρεί το αποτέλεσμα της στον αριστερό τελεστέο της εν λόγω ισότητας.
%=	Ακέραιο Διά ΚΑΙ Καταχώρηση (Modulo AND Assignment)	Εκτελεί την ακέραια διαίρεση του αριστερού τελεστέου (ακέραιος διαιρετέος) μίας ισότητας με το δεξιό τελεστέο (ακέραιος διαιρέτης) αυτής της ισότητας και καταχωρεί το αποτέλεσμα (ακέραιο υπόλοιπο) στον αριστερό τελεστέο της εν λόγω ισότητας.

Πίνακας 6: Οι τελεστές εκχώρησης της JavaScript

Στο σημείο τούτο, είναι αξιόλογο να αναφερθεί ότι υπάρχει η δυνατότητα να γίνει εκχώρηση μίας τιμής σε πολλές μεταβλητές ταυτόχρονα.

Τέταρτον, οι τελεστές σύγκρισης, που χρησιμοποιούνται για συγκρίσεις τελεστέων και παρουσιάζονται στον επόμενο Πίνακα:

Τελεστής	Ονομασία	Σημασία
==	Ίσο (Equal)	Ελέγχει αν οι τιμές δύο τελεστών είναι ίσες, αν ναι, η εν λόγω συνθήκη γίνεται true, αν όχι, false.
!=	Όχι Ίσο (Not Equal)	Ελέγχει αν οι τιμές δύο τελεστών είναι διαφορετικές, αν ναι, η εν λόγω συνθήκη γίνεται true, αν όχι, false.
>	Μεγαλύτερο (Greater than)	Ελέγχει αν η τιμή του αριστερού τελεστού μίας ανισότητας είναι μεγαλύτερη από την τιμή του δεξιού τελεστού αυτής της ανισότητας, αν ναι, η εν λόγω συνθήκη γίνεται true, αν όχι, false.
<	Μικρότερο (Less than)	Ελέγχει αν η τιμή του αριστερού τελεστού μίας ανισότητας είναι μικρότερη από την τιμή του δεξιού τελεστού αυτής της ανισότητας, αν ναι, η εν λόγω συνθήκη γίνεται true, αν όχι, false.
>=	Μεγαλύτερο ή Ίσο (Greater than or Equal)	Ελέγχει αν η τιμή του αριστερού τελεστού μίας ανισοισότητας είναι μεγαλύτερη ή ίση από την τιμή του δεξιού τελεστού αυτής της ανισοισότητας, αν ναι, η εν λόγω συνθήκη γίνεται true, αν όχι, false.
<=	Μικρότερο ή Ίσο (Less than or Equal)	Ελέγχει αν η τιμή του αριστερού τελεστού μίας ανισοισότητας είναι μικρότερη ή ίση από την τιμή του δεξιού τελεστού, αυτής της ανισοισότητας, αν ναι, η συνθήκη γίνεται true, αν όχι, false.

Πίνακας 7: Οι τελεστές σύγκρισης της JavaScript

Όπως είναι φανερό, οι τελεστές σύγκρισης επιστρέφουν λογικές τιμές (true ή false). Επίσης, δεν πρέπει να συγχέεται ο τελεστής ισότητας (==) με τον τελεστή εκχώρησης (=), που παρουσιάστηκε αμέσως παραπάνω.

Πέμπτον, οι λογικοί τελεστές (logical operators), οι οποίοι συνδέουν δύο ή περισσότερες προτάσεις και δημιουργούν μία σύνθετη λογική πρόταση. Υπενθυμίζεται πως μία λογική πρόταση μπορεί να είναι αληθής (true) ή 1 είτε ψευδής (false) ή 0. Οι λογικοί τελεστές της γλώσσας JavaScript προβάλλονται στον παρακάτω Πίνακα:

Τελεστής	Ονομασία	Σημασία
&&	Λογικό ΚΑΙ (Logical AND)	Αν δύο λογικές προτάσεις είναι true, τότε μία σύνθετη λογική πρόταση που αποτελείται από αυτές τις προτάσεις γίνεται true, αλλιώς false.
	Λογικό Ή (Logical OR)	Αν τουλάχιστον μία από δύο λογικές προτάσεις είναι true, τότε μία σύνθετη λογική πρόταση που αποτελείται από αυτές τις προτάσεις γίνεται true, αλλιώς false.

!	Λογικό ΟΧΙ (Logical NOT)	Αντιστρέφει τη λογική τιμή μίας λογικής πρότασης. Έτσι, αν η τιμή της είναι true, την κάνει false, ενώ, αν είναι false, την κάνει true.
^	Λογικό Αποκλειστικό Ή (Logical XOR)	Αν και μόνο αν η μία από δύο λογικές προτάσεις είναι true, τότε μία σύνθετη λογική πρόταση που αποτελείται από αυτές τις προτάσεις γίνεται true, αλλιώς false.

Πίνακας 8: Οι λογικοί τελεστές της JavaScript

Έκτον, ο τελεστής υπόθεσης (conditional operator), που συχνά αναφέρεται και ως τριαδικός τελεστής (ternary operator), δέχεται τρία ορίσματα και συντάσσεται ως εξής:

logicalExpression? statement1: statement2

Όπου:

- *logicalExpression*: Η εκάστοτε λογική πρόταση ενός JavaScript κώδικα που εξετάζεται αν είναι true ή false.
- *statement1*: Η εκάστοτε εντολή ενός JavaScript κώδικα που εκτελείται αν η *logicalExpression* είναι true.
- *statement2*: Η εκάστοτε εντολή ενός JavaScript κώδικα που εκτελείται αν η *logicalExpression* είναι false.

Έβδομον, οι τελεστές δυαδικών ψηφίων (bitwise operators), οι οποίοι εκτελούν λογικές πράξεις μεταξύ ακεραίων αριθμών, και όχι μεταξύ λογικών παραστάσεων, όπως πράττουν οι λογικοί τελεστές. Οι τελεστές δυαδικών ψηφίων που συναντώνται στη γλώσσα JavaScript παρατίθενται στον επόμενο Πίνακα:

Τελεστής	Ονομασία	Σημασία
&	Ψηφίο προς Ψηφίο ΚΑΙ (Bitwise AND)	Εκτελεί την ψηφίο προς ψηφίο πράξη του Λογικού Και μεταξύ δύο ακεραίων τελεστών.
	Ψηφίο προς Ψηφίο Ή (Bitwise OR)	Εκτελεί την ψηφίο προς ψηφίο πράξη του Λογικού Η μεταξύ δύο ακεραίων τελεστών.
^	Ψηφίο προς Ψηφίο Αποκλειστικό Ή (Bitwise XOR)	Εκτελεί την ψηφίο προς ψηφίο πράξη του Λογικού Αποκλειστικού Λογικού Ή μεταξύ δύο ακεραίων τελεστών.
~	Ψηφίο προς Ψηφίο ΟΧΙΪΣΥΜΠΛΗΡΩΜΑ (Bitwise NOT\ Complement)	Υπολογίζει το συμπλήρωμα ενός ακεραίου τελεστού, αντιστρέφοντας τα ψηφία του εν λόγω τελεστού ένα προς ένα (τα 0 τα μετατρέπει σε 1 και τα 1 σε 0).
<<	Ψηφίο προς Ψηφίο Αριστερή Ολίσθηση (Bitwise Shift Left Operator)	Μετακινεί όλα τα ψηφία του εκάστοτε ακεραίου τελεστού που είναι τοποθετημένος στα αριστερά του τελεστή <<, προς τα αριστερά, τόσες θέσεις όσες υποδεικνύει ο εκάστοτε ακεραίος τελεστής που είναι τοποθετημένος στα δεξιά του τελεστή <<. Τα νέα ψηφία συμπληρώνονται με μηδέν. Μετακίνηση ενός ψηφίου του προαναφερόμενου αριστερού τελεστού προς τα αριστερά κατά μία θέση είναι ισοδύναμη με πολλαπλασιασμό αυτού του τελεστού επί δύο, δύο ψηφίων ισοδυναμεί με πολλαπλασιασμό επί 4, τριών ψηφίων ισοδυναμεί με πολλαπλασιασμό επί 8 και ούτω καθεξής.

>>	Ψηφίο προς Ψηφίο Δεξιά Ολί-σθηση με Πρόσημο (Bitwise Shift Right with Sign Operator)	Μετακινεί όλα τα ψηφία του εκάστοτε ακεραίου τελεστέου που είναι τοποθετημένος στα αριστερά του τελεστή >> προς τα δεξιά, τόσες θέσεις όσες υποδεικνύει ο εκάστοτε ακεραίος τελεστής που είναι τοποθετημένος στα δεξιά του τελεστή >>. Τα ψηφία που συμπληρώνονται στα αριστερά του προαναφερόμενου αριστερού τελεστέου, προκειμένου αυτός να μετακινηθεί προς τα δεξιά, εξαρτώνται από το πρόσημο του εν λόγω τελεστή, πριν την προαναφερόμενη μετακίνηση. Αν το συγκεκριμένο πρόσημο ήταν θετικό, ο παραπάνω αριστερός τελεστέος έχει μηδενικά στα υψηλά ψηφία, μετά την παραπάνω μετακίνηση, ενώ, αν ήταν αρνητικό, έχει άσσους στα υψηλά ψηφία. Μετακίνηση ενός ψηφίου του προηγούμενου αριστερού τελεστέου προς τα δεξιά κατά μία θέση είναι ισοδύναμη με διαίρεση αυτού του τελεστέου διά δύο, δύο ψηφίων ισοδυναμεί με διαίρεση διά 4, τριών ψηφίων ισοδυναμεί με διαίρεση διά 8 και ούτω καθεξής (εξαιρώντας σε καθεμία από τις προγενέστερες διαιρέσεις το υπόλοιπο που προκύπτει).
>>>	Ψηφίο προς Ψηφίο Δεξιά Ολί-σθηση με Μηδενικά (Bitwise Shift Right with Zero Operator)	Ο συγκεκριμένος τελεστής έχει την ίδια δράση με τον τελεστή >>, με τη διαφορά ότι τα ψηφία που συμπληρώνονται από τα αριστερά, για να πραγματοποιηθεί η μετακίνηση προς τα δεξιά, είναι πάντα ίσα με μηδέν.

Πίνακας 9: Οι τελεστές δυαδικών ψηφίων της JavaScript

[36] [41] [42] [43] [46]

2.12.5 Οι Δομές Προγραμματισμού της JavaScript

Παραπάνω αποκτήθηκε μία συνάφεια με τα βασικά χαρακτηριστικά της γλώσσας JavaScript. Στην καθημερινή ζωή, όμως, μία Γλώσσα Προγραμματισμού (Programming Language) χρησιμοποιείται για τη λύση προβλημάτων. Η λύση ενός προβλήματος δε γίνεται με δύο-τρεις εντολές ή με ένα πολύ απλό σενάριο, αλλά απαιτεί κάποιες πιο σύνθετες ενέργειες.

Εφόσον, λοιπόν, απαιτείται η επίλυση ενός προβλήματος, καταρχήν αναλύεται, καταγράφοντας τα στοιχεία που δίνονται καθώς και αυτά που ζητούνται σε κάθε περίπτωση. Στη συνέχεια δημιουργείται ο αλγόριθμος του εν λόγω προβλήματος (problem algorithm), δηλαδή τα διαδοχικά βήματα και οι ενέργειες που πρέπει να πραγματοποιηθούν, προκειμένου να επιλυθεί το συγκεκριμένο πρόβλημα. Η διατύπωση του παραπάνω αλγορίθμου γίνεται είτε με τη βοήθεια ειδικών σχημάτων (λογικό διάγραμμα) (logical diagram) είτε με τη βοήθεια κατάλληλων χαρακτηριστικών λέξεων (ψευδογλώσσα) (pseudocode). Μόλις γίνει η διατύπωση του προαναφερόμενου αλγορίθμου με λογικό διάγραμμα ή με ψευδογλώσσα, εφαρμόζονται μία σειρά από εντολές (statements) και εκφράσεις-προτάσεις-παραστάσεις (expressions) μίας γλώσσας προγραμματισμού ή σεναρίων, ώστε να μετατρέψουν το συγκεκριμένο αλγόριθμο σε πηγαίο κώδικα (source code) αυτής της γλώσσας. Αφού γίνει αυτό, στη συνέχεια ο πηγαίος κώδικας μεταφράζεται σε εκτελέσιμο πρόγραμμα από τον εκάστοτε ηλεκτρονικό υπολογιστή, δηλαδή σε ακολουθίες από 0 και 1, τα οποία είναι τα μόνα σύμβολα που καταλαβαίνει ο κάθε υπολογιστής. Η μετατροπή αυτή γίνεται με τη βοήθεια του μεταγλωττιστή (compiler) που διαθέτουν οι γλώσσες προγραμματισμού ή, αν πρόκειται για μία γλώσσα σεναρίων, όπως είναι, για παράδειγμα, η JavaScript, μέσω του εκάστοτε ερμηνευτή-προγράμματος περιήγησης (browser) (πχ Opera). Στην όλη πορεία που σε κάθε περίπτωση ακολουθείται ενδέχεται να γίνουν κάποια συντακτικά ή λογικά λάθη, τα οποία πρέπει να εντοπιστούν και να διορθωθούν με μία διαδικασία που ονομάζεται εντοπισμός σφαλμάτων (debugging).

Από τα λίγα που αναφέρθηκαν πιο πάνω γίνεται κατανοητό ότι ουσιαστικό ρόλο στην επίλυση ενός προβλήματος παίζει η κατάλληλη σχεδίαση και διατύπωση ενός αλγορίθμου. Αν η σχεδίαση αυτού του αλγορίθμου δε γίνει σωστά, δεν πρόκειται να δοθεί λύση στο συγκεκριμένο πρόβλημα. Οι βασικές δομές ελέγχου (control structures) μίας γλώσσας προγραμματισμού ή σεναρίων που χρησιμοποιούνται για την κατάστρωση ενός αλγορίθμου είναι οι εξής: η δομή ακολουθίας (sequence structure), η δομή επιλογής (choice structure) και η δομή επανάληψης (repeat structure).

Αρχικά, η δομή ακολουθίας είναι η πιο απλή αλγοριθμική δομή. Αποτελείται από μία ακολουθία εντολών οι οποίες εκτελούνται η μία μετά την άλλη με τη σειρά που εμφανίζονται.

Ύστερα, η δομή επιλογής είναι εκείνη η οποία δίνει νοημοσύνη στον υπολογιστή. Σε πολλές περιπτώσεις πρέπει να αποφασιστεί αν θα ακολουθηθεί η μία ή η άλλη πορεία, ανάλογα με την τιμή μίας λογικής συνθήκης (logical condition), δηλαδή μίας έκφρασης-πρότασης-παράστασης που επιστρέφει μία λογική τιμή (true ή false). Αυτό μπορεί να διατυπωθεί στον εκάστοτε αλγόριθμο με τη βοήθεια της δομής επιλογής. Η δομή επιλογής έχει διάφορες παραλλαγές, οι οποίες θα εξεταστούν στη συνέχεια.

Η πρώτη και απλούστερη μορφή από τις δομές επιλογής είναι η if, η οποία δίνει αποκλειστικά τη δυνατότητα εξέτασης μίας λογικής συνθήκης και συντάσσεται ως εξής:

```
if (condition)
{
    statements;
}
```

Όπου:

- condition: Η εκάστοτε συνθήκη της δομής if ενός JavaScript σεναρίου που εξετάζεται και η οποία, αν είναι true, εκτελούνται οι statements, ενώ, αν είναι false, δεν εκτελείται τίποτα και το προαναφερόμενο σενάριο συνεχίζει την ακολουθιακή δομή του. Τις περισσότερες φορές μέσα στην παραπάνω συνθήκη υπάρχουν τελεστές σύγκρισης.
- statements: Η εκάστοτε ομάδα εντολών της δομής if ενός JavaScript σεναρίου που εκτελείται, αν η condition επιστρέψει την τιμή true. Εδώ, να σημειωθεί πως η συγκεκριμένη ομάδα μπορεί να περιέχει αποκλειστικά μία ή ακόμα και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.

Η δεύτερη και περισσότερο σύνθετη μορφή από τις δομές επιλογής είναι η if...else, η οποία δίνει τόσο τη δυνατότητα εξέτασης μίας λογικής συνθήκης όσο και τη δυνατότητα μίας εναλλακτικής επιλογής σε περίπτωση που αυτή η συνθήκη δεν ικανοποιείται και συντάσσεται ως εξής:

```
if (condition)
{
    statements1;
}
else
{
    statements2;
}
```

Όπου:

- condition: Η εκάστοτε συνθήκη της δομής if...else ενός JavaScript σεναρίου που εξετάζεται και η οποία, αν είναι true, το προαναφερόμενο σενάριο εκτελεί τη statements1, ενώ, αν είναι false, εκτελεί τη statements2. Τις περισσότερες φορές μέσα στην παραπάνω συνθήκη υπάρχουν τελεστές σύγκρισης.
- statements1: Η εκάστοτε πρώτη ομάδα εντολών της δομής if...else ενός JavaScript σεναρίου που εκτελείται, αν η condition επιστρέψει την τιμή true. Εδώ, να σημειωθεί ότι η συγκεκριμένη ομάδα μπορεί να περιέχει αποκλειστικά μία ή ακόμα και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.

- *statements2*: Η εκάστοτε δεύτερη ομάδα εντολών της δομής η *if...else* ενός JavaScript σεναρίου που εκτελείται, αν η *condition* επιστρέψει την τιμή *false*. Εδώ, να σημειωθεί ότι η συγκεκριμένη ομάδα μπορεί να περιέχει αποκλειστικά μία ή, ακόμα, και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.

Η τρίτη και πολυπλοκότερη μορφή από τις δομές επιλογής είναι η *if...else if...else*, η οποία δίνει τόσο τη δυνατότητα εξέτασης πολλαπλών συνθηκών όσο και τη δυνατότητα μίας εναλλακτικής επιλογής σε περίπτωση που καμία από αυτές τις λογικές συνθήκες δεν ικανοποιείται και συντάσσεται ως εξής:

```

if (condition1)
{
    statements1;
}
else if (condition2)
{
    ...statements2;
}
else if (condition3)
{
    ...statements3;
}
.....
else if (conditionN)
{
    ...statementsN;
}
else
{
    ...statements;
}

```

Όπου:

- *condition1, condition2, condition3, ... , conditionN*: Οι εκάστοτε συνθήκες 1,2,3, ..., N, αντίστοιχα, της δομής *if...else if...else* ενός JavaScript σεναρίου που εξετάζονται με τη σειρά, έως ότου κάποια να βρεθεί *true*, οπότε και το προαναφερόμενο σενάριο εκτελεί μία από τις *statements1, statements2, statements3, ... , statementsN*, η οποία συνοδεύει την αντίστοιχη συνθήκη που θα επιστρέψει την τιμή *true*. Τις περισσότερες φορές μέσα στις παραπάνω συνθήκες υπάρχουν τελεστές σύγκρισης.
- *statements1, statements2, statements3, ... , statementsN*: Οι εκάστοτε ομάδες εντολών 1,2,3, ..., N, αντίστοιχα, της δομής *if...else if...else* ενός JavaScript σεναρίου από τις οποίες εκτελείται εκείνη που συνοδεύει την πρώτη από τις *condition1, condition2, condition3, ... , conditionN*, που θα επιστρέψει την τιμή *true*. Εδώ, να σημειωθεί πως η ομάδα εντολών που εκτελείται κάθε φορά μπορεί να περιέχει αποκλειστικά μία ή ακόμα και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.
- *statements*: Η εκάστοτε ομάδα εντολών της δομής *if...else if...else* ενός JavaScript σεναρίου που εκτελείται, αν καμία εκ των *condition1, condition2, condition3, ... , conditionN* *condition* δεν επιστρέψει την τιμή *true*. Εδώ, να σημειωθεί ότι η συγκεκριμένη ομάδα εντολών μπορεί να περιέχει αποκλειστικά μία ή ακόμα και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.

Στο σημείο τούτο, να τονιστεί ότι, αν σε μία από τις προαναφερθείσες δομές επιλογής, μία *if* ή μία *else if* ή μία *else* δε συνοδεύεται από καμία εντολή, τότε το εκάστοτε JavaScript σενάριο δεν κάνει

κάτι, αλλά συνεχίζει κανονικά την εκτέλεσή του, και ουσιαστικά η χρήση της κενής if ή της else if ή της else δεν έχει νόημα.

Όταν, τώρα, πρέπει να ελεγχθεί η τιμή μίας μεταβλητής ή μίας έκφρασης, η οποία μπορεί να πάρει διάφορες τιμές, και στη συνέχεια να ακολουθηθεί κάποια πορεία, αν και είναι εφικτό, όπως αναφέρθηκε ακριβώς πιο πάνω, να εκτελεστεί η διαδικασία αυτή με μία δομή if ... else if ... else, υπάρχει η ευχέρεια να γίνει χρήση μίας τέταρτης μορφής της δομής πολλαπλής επιλογής, της switch, η οποία διευκολύνει σε τέτοιες περιπτώσεις και συντάσσεται ως εξής:

```
switch ( )
{
  case (value1);
  ...{
  .....statements1;
  ...}
  break;
  case (value2);
  ...{
  statements2;
  ...}
  break;
  case (value3);
  ...{
  statements3;
  ...}
  break;
  .....
  case (valueN);
  ...{
  statementsN;
  ...}
  break;
  default:
  ...{
  statements;
  ...}
  break;
}
```

Όπου:

- expression: Η εκάστοτε μεταβλητή ή έκφραση της δομής switch ενός JavaScript σεναρίου η οποία εξετάζεται ως προς την τιμή της. Εδώ να σημειωθεί πως η συγκεκριμένη μεταβλητή ή έκφραση μπορεί να πάρει πληθώρα διαφορετικών τιμών.
- value1, value2, value3, ... , valueN: Οι εκάστοτε δυνατές τιμές ενός JavaScript σεναρίου που μπορεί να πάρει η expression.
- statements1, statements2, statements3, ... , statementsN: Οι εκάστοτε ομάδες εντολών 1,2,3, ... , N, αντίστοιχα, της δομής switch ενός JavaScript σεναρίου από τις οποίες εκτελείται εκείνη που συνοδεύει την πρώτη από τις value1, values2, value3, ... , valueN, που αποτελεί την τιμή της expression. Εδώ, να σημειωθεί ότι η κάθε μία από τις προαναφερθείσες ομάδες εντολών μπορεί να περιέχει αποκλειστικά μία ή, ακόμα, και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.
- statements: Η εκάστοτε ομάδα εντολών της δομής switch ενός JavaScript σεναρίου που εκτελείται προκαθορισμένα, δηλαδή, αν η τιμή της expression δε γίνει ίση με καμία από τις value1, value2, value3, ... , valueN. Εδώ, να σημειωθεί ότι η κάθε μία από τις προαναφερθείσες ομάδες

εντολών μπορεί να περιέχει αποκλειστικά μία ή, ακόμα, και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.

Στο σημείο αυτό πρέπει να δοθεί έμφαση στο γεγονός ότι στη δομή switch ενός JavaScript σεναρίου η χρησιμοποιούμενη έκφραση ή μεταβλητή πρέπει να είναι κάποιος ακέραιος αριθμός ή κάποιος χαρακτήρας. Επίσης, σε κάθε εντολή case της δομής switch ενός JavaScript σεναρίου πρέπει να τοποθετείται η εντολή υπερπήδησης break, με την οποία πραγματοποιείται έξοδος από τη δομή switch, εφόσον εκτελεστεί η συγκεκριμένη case, διότι, αν δε γίνει κάτι τέτοιο, τότε το προαναφερόμενο σενάριο θα συνεχίσει τη ροή του, εκτελώντας και τις εντολές της επόμενης case, ενέργεια που πιθανότατα να οδηγήσει σε λογικό λάθος. Στο σημείο τούτο, να σχολιαστεί πως, αν στη δομή switch ενός JavaScript σεναρίου επιλεγεί μία εντολή case, η οποία δεν περιέχει καμία εντολή, αλλά περιέχει την εντολή break, τότε το παραπάνω σενάριο δεν θα εκτελέσει κάποια ενέργεια και θα βγει από τη δομή switch βρίσκεται, συνεχίζοντας την ακολουθιακή δομή του, όμως, αν επιλεγεί μία case που δεν περιέχει ούτε εντολές ούτε break, τότε στην προκειμένη αυτή το πρωτότερο σενάριο θα εκτελέσει τις εντολές της αμέσως επόμενης case που θα συναντήσει και ουσιαστικά οι δύο αυτές case θα ισοδυναμούν με δύο εκφράσεις μίας λογικής συνθήκης, που συνδέονται μεταξύ τους με το λογικό τελεστή ||. Επιπλέον, στη δομή switch ενός JavaScript σεναρίου η χρήση της εντολής break μετά την εντολή default είναι προαιρετική, από τη στιγμή που η τελευταία έπεται όλων των εντολών case, ωστόσο, αν η default παρεμβάλλεται μεταξύ των case, τότε είναι επιτακτική η χρήση της break.

Πολλές φορές, ωστόσο, οι ανάγκες ενός προβλήματος επιβάλλουν την επανάληψη μίας ομάδας εντολών. Η δομή επανάληψης είναι μία προγραμματιστική δομή η οποία επιτρέπει την επανάληψη μίας σειράς από εντολές. Η γλώσσα JavaScript διαθέτει τέσσερις μορφές σχετικά με τη δομή επανάληψης: Τη while, την do...while, τη for και τη for...in, η οποία απαντάται αποκλειστικά και μόνο στην JavaScript.

Ο βρόγχος while είναι η βασική δομής επανάληψης της γλώσσας JavaScript και χρησιμοποιείται, όταν ο αριθμός των επαναλήψεων ενός JavaScript σεναρίου δεν είναι προκαθορισμένος, αλλά εξαρτάται από μία λογική συνθήκη. Αν αυτή η συνθήκη είναι true, τότε εκτελείται μία ομάδα εντολών (ή αποκλειστικά μία ή ακόμα και καμία εντολή) που βρίσκεται εντός του βρόγχου while. Αν η προαναφερόμενη συνθήκη γίνει false, τότε ο βρόγχος while σταματά να εκτελείται και η ροή του προαναφερόμενου σεναρίου συνεχίζεται με τις επόμενες εντολές του. Εννοείται πως, αν η παραπάνω συνθήκη, που καθορίζει την εκτέλεση του βρόγχου βρεθεί false στην αρχή, τότε ο βρόγχος while δε θα εκτελεστεί καμία φορά. Η while, λοιπόν, συντάσσεται ως εξής:

```
while (condition)
{
    statements;
}
```

Όπου:

- condition: Η εκάστοτε συνθήκη του βρόγχου while ενός JavaScript σεναρίου που εξετάζεται και η οποία, αν είναι true, το προαναφερόμενο σενάριο εκτελεί τη statements.
- statements: Η εκάστοτε ομάδα εντολών του βρόγχου while ενός JavaScript σεναρίου που εκτελείται, αν η condition επιστρέψει την τιμή true. Εδώ, να σημειωθεί ότι η statements μπορεί να περιέχει αποκλειστικά μία ή ακόμα και καμία εντολή, οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.

Παρόμοιος με το βρόγχο while είναι και ο βρόγχος do...while, με τη διαφορά ότι η εκάστοτε λογική συνθήκη δεν ελέγχεται στην αρχή του βρόγχου do...while, αλλά στο τέλος. Αυτό εξασφαλίζει την εκτέλεση του βρόγχου do...while τουλάχιστον μία φορά ακόμα και αν η προαναφερόμενη συνθήκη είναι false. Η do...while, λοιπόν, συντάσσεται ως εξής:

```
do
{
    statements;
```

```
}  
while (condition);
```

Όπου:

- **condition:** Η εκάστοτε συνθήκη του βρόγχου `do...while` ενός JavaScript σεναρίου που εξετάζεται και η οποία, αν είναι `true`, το προαναφερόμενο σενάριο εκτελεί τη `statements`.
- **statements:** Η εκάστοτε ομάδα εντολών του βρόγχου `do...while` ενός JavaScript σεναρίου που εκτελείται, αν η `condition` επιστρέψει την τιμή `true`. Εδώ, να σημειωθεί ότι η `statements` μπορεί να περιέχει αποκλειστικά μία ή ακόμα και καμία εντολή, οπότε το ζεύγος των αγκίστρων `{ }` είναι προαιρετικό.

Με το βρόγχο `for`, τώρα, μπορεί να πραγματοποιηθεί ένας προκαθορισμένος αριθμός επαναλήψεων μίας εντολής ή μίας ομάδας εντολών, σε αντίθεση με τους βρόγχους `while` και `do...while`. Ο βρόχος `for` περιέχει μία λογική συνθήκη, η οποία περιλαμβάνει συνήθως ένα μετρητή. Ο μετρητής αυτός μπορεί να μεταβάλλεται προς τα πάνω ή προς τα κάτω. Η εκτέλεση των εντολών που υπάρχουν μέσα στο βρόχο `for`, γίνεται, εφόσον ο προαναφερόμενος μετρητής επαληθεύει την παραπάνω συνθήκη. Όταν η τιμή του συγκεκριμένου μετρητή δεν επαληθεύει πλέον την προηγούμενη συνθήκη, τότε σταματούν να επαναλαμβάνονται οι εντολές που βρίσκονται μέσα στο βρόχο `for`. Η σύνταξη, λοιπόν, του βρόγχου `for` είναι η ακόλουθη:

```
for (initialization; condition; iterationStep)  
{  
...statements;  
}
```

Όπου:

- **initialization:** Η εκάστοτε έκφραση του βρόγχου `for` ενός JavaScript σεναρίου που αποδίδει αρχική τιμή στον εκάστοτε μετρητή του συγκεκριμένου βρόγχου. Εκτελείται μία φορά στην αρχή κάθε βρόγχου `for`.
- **condition:** Η εκάστοτε συνθήκη του βρόγχου `for` ενός JavaScript σεναρίου η οποία ελέγχεται πριν από κάθε επανάληψη, συμπεριλαμβανομένης και της πρώτης. Αν είναι `true`, το προαναφερόμενο σενάριο προχωρά στην εκτέλεση της `statements`, αν είναι `false`, ο εν λόγω βρόχος τερματίζεται.
- **iterationStep:** Η εκάστοτε έκφραση του βρόγχου `for` ενός JavaScript σεναρίου η οποία καθορίζει το βήμα της εκάστοτε επανάληψης.
- **statements:** Η εκάστοτε ομάδα εντολών του βρόγχου `for` ενός JavaScript σεναρίου που εκτελείται, αν η `condition` επιστρέψει την τιμή `true`. Εδώ, να σημειωθεί πως η `statements` μπορεί να περιέχει αποκλειστικά μία ή ακόμα και καμία εντολή, οπότε το ζεύγος των αγκίστρων `{ }` είναι προαιρετικό.

Κάπου εδώ, να επισημανθεί πως, αν σε μία από τις προαναφερθείσες δομές επανάληψης, μία `while`, ή μία `do`, ή μία `for` δεν ακολουθείται από καμία εντολή, τότε το προαναφερόμενο σενάριο δεν επιτελεί κάποια λειτουργία, αλλά εισάγεται σε αυτό το κενό σημείο του μία καθυστέρηση, η οποία σε κάποιες περιπτώσεις είναι επιθυμητή, προκειμένου να εξυπηρετήσει έναν ιδιαίτερο σκοπό του εν λόγω σεναρίου.

Οι βρόχοι της γλώσσας JavaScript, που μελετήθηκαν μέχρι τώρα, είναι απλοί. Υπάρχει η δυνατότητα, όμως, να τοποθετηθεί ένας βρόχος μέσα σε έναν άλλον και να προκύψουν έτσι ένθετοι βρόχοι (`nested loops`), αρκεί να προσεχθούν τα εξής:

- Ο εκάστοτε εσωτερικός βρόχος πρέπει να περιέχεται εξ ολοκλήρου στον εκάστοτε εξωτερικό που τον περιέχει.
- Δεν υπάρχει η πιθανότητα εισχώρησης στο εσωτερικό ενός βρόγχου, αν δεν πραγματοποιηθεί πέρασμα από την πρώτη εντολή του.
- Δε γίνεται να εκμεταλλευτεί η ίδια μεταβλητή ως απαριθμητής σε δύο ή περισσότερους βρόχους, όταν ο ένας εμπεριέχεται μέσα στον άλλον.

Τέλος, να τονιστεί ότι πολλές φορές είναι αναγκαίο ένας προγραμματιστής της γλώσσας JavaScript να είναι ικανός να μεταβάλλει τη ροή τη δομή ενός σεναρίου της τρέχουσας γλώσσας. Για να το καταφέρει αυτό έχει τους παρακάτω τρόπους:

- Με χρήση της εντολής break: Χρησιμοποιείται με τις δομές επανάληψης while, do ... while και for καθώς και με την εντολή διακλάδωσης switch ενός JavaScript σεναρίου, όπως έχει ήδη αναφερθεί. Μόλις η ροή ενός σεναρίου συναντήσει μέσα σε μία δομή του την εντολή break, εξέρχεται της εν λόγω δομής και συνεχίζει με την πρώτη εντολή που υπάρχει μετά από αυτή. Η εντολή break ενός σεναρίου εκμεταλλεύεται για την έξοδο από ένα βρόχο, προτού συμπληρωθούν όλες οι ανακυκλώσεις του ή για την έξοδο από μία δομή επιλογής του προαναφερόμενου σεναρίου, εφόσον ικανοποιηθεί μία από τις συνθήκες της και εκτελεστεί ο αντίστοιχος κώδικας της.
- Με χρήση της εντολής continue: Διατίθεται με τις δομές επανάληψης while, do ... while και for ενός JavaScript σεναρίου. Μόλις η ροή ενός από τους βρόγχους επανάληψης ενός σεναρίου συναντήσει την εντολή continue, η εκτέλεση των εντολών, που αντιστοιχούν στην τρέχουσα σε κάθε περίπτωση τιμή του μετρητή ενός από τους προαναφερόμενους βρόγχους, σταματάει και το συγκεκριμένο σενάριο συνεχίζει με την επανάληψη που αντιστοιχεί στην επόμενη τιμή του προηγούμενου μετρητή.

Μάλιστα, για να επιτευχθεί μεγαλύτερη ακρίβεια στον έλεγχο ενός βρόγχου επανάληψης ενός JavaScript σεναρίου ο εκάστοτε προγραμματιστής μπορεί να χρησιμοποιεί μία ετικέτα συνδυαστικά με τις εντολές break ή continue. Μία ετικέτα είναι ένας προσδιοριστής ακολουθούμενος από μία άνω-κάτω τελεία (:).

Υπάρχει, ωστόσο, μία δομή επανάληψης, η οποία απαντάται εξολοκλήρου στη γλώσσα JavaScript, όπως εξάλλου τονίστηκε παραπάνω. Η δομή αυτή είναι ο βρόγχος επανάληψης for...in βρόγχος και χρησιμεύει για την κυκλική κίνηση διαμέσου των ιδιοτήτων ενός αντικειμένου της JavaScript. Η σύνταξη, λοιπόν, του βρόγχου for...in είναι η ακόλουθη:

```
for (propertyVariable in object)
{
  statements
}
```

Όπου:

- propertyVariable: Το όνομα της εκάστοτε μεταβλητής στην οποία καταχωρείται σε κάθε επανάληψη του βρόγχου for...in ενός JavaScript σεναρίου μία από τις ιδιότητες του υπό μελέτη σε κάθε περίπτωση JavaScript αντικειμένου. Ο συγκεκριμένος βρόγχος επαναλαμβάνεται μέχρι να εξαντληθούν όλες οι ιδιότητες του προαναφερόμενου αντικειμένου.
- object: Το εκάστοτε αντικείμενο ενός JavaScript σεναρίου του οποίου οι ιδιότητες καταχωρούνται μία προς μία στην propertyVariable, ύστερα από κάθε επανάληψη του βρόγχου for...in του προαναφερόμενου σεναρίου.
- statements: Η εκάστοτε ομάδα εντολών του βρόγχου for...in ενός JavaScript σεναρίου που εκτελείται, προκειμένου να καταχωρηθούν όλες οι ιδιότητες του object. Εδώ, να σημειωθεί ότι η συγκεκριμένη ομάδα μπορεί να περιέχει αποκλειστικά μία εντολή (όταν το object διαθέτει μόνο μία ιδιότητα), οπότε το ζεύγος των αγκίστρων {} είναι προαιρετικό.

[41] [42] [43] [46]

2.12.6 Οι Συναρτήσεις της JavaScript

Συνάρτηση (function), γενικά στις γλώσσες σεναρίων, ονομάζεται ένα τμήμα κώδικα το οποίο δύναται να συνίσταται από μία ομάδα από εντολές ή ακόμα και μία εντολή με ξεχωριστό όνομα που εκτελεί μία συγκεκριμένη εργασία. Μία συνάρτηση καταρχήν ορίζεται και κατόπιν καλείται. Η κλήση μίας συνάρτησης γίνεται με το όνομά της και έχει ως αποτέλεσμα την εκτέλεση του κώδικα που περιέχει κάθε φορά. Η χρήση συναρτήσεων διευκολύνει τη δημιουργία αρθρωτών και όχι μονολιθικών προγραμμάτων. Επιπλέον, ο κώδικας μίας συνάρτησης γράφεται μία φορά κατά τον ορισμό της και μπορεί να κληθεί από διάφορα σημεία ενός σεναρίου. Ο ορισμός των συναρτήσεων στην JavaScript γίνεται εντός

του εκάστοτε JavaScript κώδικα, δηλαδή εντός των ετικετών `<script>` και `</script>`, και έχει την ακόλουθη σύνταξη:

```
function name (parameters)
{
...statements;
}
```

Όπου:

- **name:** Το όνομα της εκάστοτε JavaScript συνάρτησης ενός JavaScript σεναρίου που είναι μοναδικό και αποδίδεται σύμφωνα με τους κανόνες που διέπουν τα ονόματα των προσδιοριστών της τρέχουσας γλώσσας και οι οποίοι αναφέρθηκαν παραπάνω.
- **parameters:** Οι παράμετροι ή αλλιώς τα ορίσματα της εκάστοτε JavaScript συνάρτησης ενός JavaScript σεναρίου, δηλαδή οι τιμές που περνούν σε αυτή τη συνάρτηση κατά την κλήση της από κατάλληλα σημεία του κώδικα του προαναφερόμενου σεναρίου και οι οποίες θα επεξεργαστούν στο εσωτερικό της. Αν υπάρχουν περισσότερες από μία τέτοιες παράμετροι, τότε χωρίζονται με κόμμα (,) μεταξύ τους. Υπάρχει η πιθανότητα, ωστόσο, να ορίζονται συναρτήσεις με μία ή ακόμα και καμία παράμετρο, οπότε στην τελευταία περίπτωση γράφεται μόνο το ζεύγος των παρενθέσεων (), χωρίς να περικλείεται τίποτα ανάμεσά τους.
- **statements:** Το σώμα της εκάστοτε JavaScript συνάρτησης ενός JavaScript σεναρίου, με άλλα λόγια μία ή περισσότερες μεταβλητές και μία ή περισσότερες εντολές που χρησιμοποιεί η προαναφερόμενη συνάρτηση και οι οποίες επιτελούν σε κάθε περίπτωση την επιθυμητή εργασία. Μπορεί, όμως, το σώμα μίας συνάρτησης να αποτελείται αποκλειστικά από μία ή ακόμα και καμία εντολή, δίχως, ωστόσο, σε ουδεμία περίπτωση να παρακάμπτεται το ζεύγος των αγκιστρών {} που το καθορίζουν.

Επιπρόσθετα, μία JavaScript συνάρτηση ενός JavaScript σεναρίου μπορεί να επιστρέψει ή να μην επιστρέψει κάποια τιμή μετά την εκτέλεση του κώδικά της. Αν επιστρέψει κάποια τιμή, αυτό γίνεται με την εντολή `return`, η οποία συντάσσεται ως εξής:

return variable;

Όπου *variable* το όνομα της εκάστοτε μεταβλητής, όπου θα εκχωρηθεί η τιμή που θα επιστρέψει η εκάστοτε JavaScript συνάρτηση ενός JavaScript σεναρίου.

Για να εκτελεστεί, τώρα, μία JavaScript συνάρτηση, πρέπει να κληθεί στο κατάλληλο σημείο ενός JavaScript σεναρίου. Η κλήση της συνάρτησης γίνεται με το όνομά της ακολουθούμενο από τις παρενθέσεις (). Ο χώρος μέσα στις παρενθέσεις είτε είναι κενός (αν δεν υπάρχουν ορίσματα), είτε περιέχει ένα ή περισσότερα ορίσματα, τα οποία χωρίζονται μεταξύ τους με κόμμα (,), όπως εξάλλου τονίστηκε και προηγουμένως.

Επιπρόσθετα, αξίζει να αναφερθεί πως, αν χρειαστεί, μία συνάρτηση στη JavaScript μπορεί να καλέσει μία άλλη συνάρτηση εντός ενός JavaScript σεναρίου. Για αυτό το σκοπό η προαναφερόμενη δεύτερη συνάρτηση τοποθετείται μέσα στο σώμα της προαναφερόμενης πρώτης. Απαραίτητη προϋπόθεση είναι ο ορισμός της εν λόγω δεύτερης συνάρτησης να έχει προηγηθεί του ορισμού της εν λόγω πρώτης. Ακόμα, μία συνάρτηση μπορεί να καλέσει τον ίδιο της τον εαυτό, οπότε και ονομάζεται αναδρομική συνάρτηση (*recursive function*). Η αναδρομή είναι τεχνική που την προσφέρουν όλες οι σύγχρονες γλώσσες προγραμματισμού και είναι χρήσιμη σε πολλές περιπτώσεις.

Κλείνοντας, να σημειωθεί ότι η JavaScript διαθέτει ένα αριθμό από καθολικές (*global*), ή, αλλιώς εσωτερικές (*built-in*) συναρτήσεις, από τις οποίες οι πιο σημαντικές φαίνονται στον ακόλουθο Πίνακα:

Συνάρτηση	Λειτουργία	Σύνταξη	Παράμετρος (-οι)	Τιμή Επιστροφής
-----------	------------	---------	------------------	-----------------

decodeURI()	Αποκωδικοποιεί έναν URI.	decodeURI(uri)	uri: Υποχρεωτική. Ένας URI που πρόκειται να αποκωδικοποιηθεί.	Ένα αντικείμενο String, που αναπαριστά έναν αποκωδικοποιημένο URI.
decodeURIComponent()	Αποκωδικοποιεί το συστατικό ενός URI.	decodeURIComponent(uri)	uri: Υποχρεωτική. Ένας URI που ένα συστατικό του πρόκειται να αποκωδικοποιηθεί.	Ένα αντικείμενο String, που αναπαριστά ένα αποκωδικοποιημένο συστατικό ενός URI.
encodeURIComponent()	Κωδικοποιεί έναν URI.	encodeURIComponent(uri)	uri: Υποχρεωτική. Ένας URI που κάθε φορά πρόκειται να κωδικοποιηθεί.	Ένα αντικείμενο String, που αναπαριστά έναν κωδικοποιημένο URI.
encodeURIComponent()	Κωδικοποιεί το συστατικό ενός URI.	encodeURIComponent(uri)	uri: Υποχρεωτική. Ένας URI που ένα συστατικό του πρόκειται να κωδικοποιηθεί.	Ένα αντικείμενο String, που αναπαριστά ένα κωδικοποιημένο συστατικό ενός URI..
eval()	Αποτιμά μία μεταβλητή, ή μία έκφραση, ή εκτελεί μία ομάδα από εντολές	eval(string)	string: Υποχρεωτική. Μία μεταβλητή, ή μία έκφραση, ή μία ομάδα εντολών που πρόκειται να αποτιμηθεί.	Όλοι οι δυνατοί τύποι δεδομένων.
isFinite()	Εξετάζει αν μία τιμή είναι πεπερασμένη, δηλαδή έγκυρος αριθμός.	isFinite(value)	value: Υποχρεωτική. Μία τιμή που πρόκειται να εξεταστεί αν εί-	Ένα αντικείμενο Boolean, που έχει την τιμή false, αν η παράμετρος value είναι +infi-

			ναι έγκυρος αριθμός.	nity, ή -infinity, ή NaN, αλλιώς έχει την τιμή true.
isNaN()	Εξετάζει αν μία τιμή δεν είναι έγκυρος αριθμός.	isNaN(value)	value: Υποχρεωτική. Η τιμή που κάθε φορά πρόκειται να εξεταστεί αν δεν είναι έγκυρος αριθμός.	Ένα αντικείμενο Boolean, που έχει την τιμή true, αν η παράμετρος value είναι NaN, αλλιώς έχει την τιμή false.
Number()	Μετατρέπει την τιμή ενός αντικειμένου σε αριθμό.	Number(object)	object: Προαιρετική. Το αντικείμενο που κάθε φορά πρόκειται να μετατραπεί σε αριθμό.	Ένα αντικείμενο Number, στο οποίο μετατρέπεται η παράμετρος object. Αν η παράμετρος object δεν μπορεί να μετατραπεί σε έναν έγκυρο αριθμό, η τιμή επιστροφής της συνάρτησης είναι Nan, ενώ, αν η παράμετρος object παραλείπεται, 0.
parseFloat()	Μετατρέπει ένα αλφαριθμητικό σε αριθμό κινητής υποδιαστολής μονής ακρίβειας.	parseFloat(string)	string: Υποχρεωτική. Το αλφαριθμητικό που κάθε φορά πρόκειται να μετατραπεί σε αριθμό κινητής υποδιαστολής	Ένα αντικείμενο Number, στο οποίο μετατρέπεται ο πρώτος χαρακτήρας της παραμέτρου string. Αν η παραπάνω μετατροπή δεν

			μονής ακρίβειας.	είναι εφικτή, η τιμή επιστροφής της συνάρτησης είναι NaN.
parseInt()	Μετατρέπει ένα αλφαριθμητικό σε ακέραιο αριθμό.	parseInt(string,radix)	i) string: Υποχρεωτική. Το αλφαριθμητικό που κάθε φορά πρόκειται να μετατραπεί σε ακέραιο αριθμό. ii) radix: Υποχρεωτική. Ένας αριθμός (από το 2 ως το 36), που κάθε φορά αναπαριστά το αριθμητικό σύστημα που χρησιμοποιείται από τη συνάρτηση	Ένα αντικείμενο Number, στο οποίο μετατρέπεται ο πρώτος χαρακτήρας της παραμέτρου string. Αν η παραπάνω μετατροπή δεν είναι εφικτή, η τιμή επιστροφής της συνάρτησης είναι NaN.
String()	Μετατρέπει την τιμή ενός αντικείμενου σε αλφαριθμητικό.	String(object)	object: Υποχρεωτική. Ένα αντικείμενο που πρόκειται να μετατραπεί σε αλφαριθμητικό.	Ένα αντικείμενο String, στο οποίο μετατρέπεται το παράμετρος object.

Πίνακας 10: Οι πιο βασικές καθολικές συναρτήσεις της JavaScript

[41] [42] [43]

2.12.7 Τα Συμβάντα και οι Χειριστές Συμβάντων της JavaScript

Στα σύγχρονα προγράμματα, η επικοινωνία μεταξύ ενός χρήστη και ενός υπολογιστή γίνεται μέσω μίας Διασύνδεσης Γραφικών Χρήστη (Graphical User Interface-GUI). Η εκάστοτε GUI περιλαμβάνει διάφορα συστατικά, όπως είναι, για παράδειγμα, οι εικόνες, τα κουμπιά, τα πλαίσια κειμένου, οι γραμμές εργαλείων, τα μενού επιλογών, οι λίστες. Με τη βοήθεια τέτοιου είδους συστατικών ο εκάστοτε χρήστης μπορεί να εκτελεί τις εργασίες του και να λαμβάνει ή να μεταβιβάζει πληροφορίες. Ένα

τέτοιο περιβάλλον επικοινωνίας συναντάμε στις Ιστοσελίδες του Παγκόσμιου Ιστού, όπου μία Σελίδα είναι εφικτό να περιλαμβάνει όλα ή κάποια από τα αντικείμενα μίας GUI.

Με τον όρο συμβάντα ή γεγονότα (events) αναφερόμαστε σε ενέργειες τις οποίες πραγματοποιεί ο εκάστοτε χρήστης σε μία GUI, όπως είναι, για παράδειγμα, το άνοιγμα ή το κλείσιμο ενός παράθυρου, η τοποθέτηση του δείκτη του ποντικιού πάνω σε μία εικόνα, το πάτημα ενός κουμπιού, η επιλογή ενός στοιχείου μίας λίστας. Μόλις λάβει χώρα ένα τέτοιο συμβάν, εκτελείται ένας ειδικός κώδικας, ο οποίος πραγματοποιεί κάποιες λειτουργίες που είναι συνδεδεμένες με το προαναφερόμενο συμβάν. Το τμήμα αυτό του κώδικα ονομάζεται χειριστής συμβάντος (event handler).

Αυτό το είδος προγραμματισμού υποστηρίζεται από όλες τις μοντέρνες γλώσσες και ονομάζεται προγραμματισμός οδηγούμενος από συμβάντα (event driven programming). Η JavaScript ανήκει στις γλώσσες που υποστηρίζουν αυτού του είδους τον προγραμματισμό.

Οι χειριστές συμβάντων της γλώσσας JavaScript μπορούν να τοποθετηθούν σε διάφορες θέσεις ενός εγγράφου HTML. Η γενική σύνταξη για την τοποθέτηση ενός χειριστή έχει δύο δυνατές μορφές, που είναι οι παρακάτω:

- Είτε με τη χρήση διπλών εισαγωγικών ("")

`<HTML_Tag attributes event handler= "JavaScript Code">`

- Είτε με τη χρήση απλών εισαγωγικών ('):

`<HTML_Tag attributes event handler= 'JavaScript Code'>`

Όπου:

- HTML_Tag: Η εκάστοτε HTML ετικέτα ενός HTML εγγράφου μέσα στην οποία τοποθετείται ο εκάστοτε χειριστής συμβάντος.
- attributes: Οι εκάστοτε ιδιότητες που προσδιορίζουν περαιτέρω την HTML_Tag. Οι ιδιότητες αυτές μπορεί να περιορίζονται σε μία ή και να παραλείπονται, εφόσον είναι επιθυμητό να αποδοθεί ένα ή και να μην αποδοθεί κανένα επιπλέον χαρακτηριστικό στην HTML_Tag.
- event handler: Ο εκάστοτε χειριστής συμβάντος που εκτελείται μόλις προκύψει ένα συμβάν σε ένα HTML έγγραφο.
- JavaScript Code: Ο κατάλληλος σε κάθε περίπτωση JavaScript κώδικας που συνοδεύει τον event handler.

Τα σημαντικότερα συμβάντα που υποστηρίζει η γλώσσα JavaScript και οι αντίστοιχοι χειριστές φαίνονται στον επόμενο Πίνακα:

Συμβάν	Χειριστής Συμβάντος	Προκαλείται...
Abort	onAbort	Όταν ο εκάστοτε χρήστης διακόπτει τη φόρτωση μίας Ιστοσελίδας.
Blur	onBlur	Όταν ο εκάστοτε χρήστης εγκαταλείπει το εκάστοτε τρέχον αντικείμενο (πχ ένα πλαίσιο κειμένου) και επιλέγει ένα άλλο σε μία Ιστοσελίδα.
Change	onChange	Όταν συμβαίνει μία αλλαγή σε ένα αντικείμενο μίας Ιστοσελίδας (πχ όταν πληκτρολογείται κείμενο σε ένα πεδίο κειμένου).
Click	onClick	Όταν ο εκάστοτε χρήστης πατάει με το ποντίκι σε ένα αντικείμενο μίας Ιστοσελίδας (πχ σε ένα κουμπί).
Dragdrop	onDragdrop	Όταν ο εκάστοτε χρήστης χρησιμοποιεί τη λειτουργία μεταφοράς και απόθεσης για ένα αντικείμενο μίας Ιστοσελίδας (πχ για μία εικόνα).
Error	onError	Όταν συμβαίνει κάποιο σφάλμα κατά τη φόρτωση ενός εγγράφου, ή μίας εικόνας, ή οποιουδήποτε άλλου αντικείμενου μίας Ιστοσελίδας.
Focus	onFocus	Όταν ο εκάστοτε χρήστης εστιάζει σε ένα αντικείμενο μίας Ιστοσελίδας (πχ σε μία εικόνα).

Keydown	onKeydown	Όταν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατάει ένα πλήκτρο του πληκτρολογίου.
Keypress	onKeypress	Όταν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατάει και ελευθερώνει ένα πλήκτρο του πληκτρολογίου.
Keyup	onKeyup	Όταν ο εκάστοτε χρήστης μίας Ιστοσελίδας ελευθερώνει ένα πλήκτρο του πληκτρολογίου.
Load	on Load	Όταν έχει ολοκληρωθεί η φόρτωση μίας Ιστοσελίδας από το εκάστοτε πρόγραμμα περιήγησης.
MouseDown	onMouseDown	Όταν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατάει ένα πλήκτρο του ποντικιού.
Mouseout	onMouseout	Όταν ο εκάστοτε χρήστης απομακρύνει το δείκτη του ποντικιού από ένα αντικείμενο μίας Ιστοσελίδας(πχ από μία εικόνα).
Mouseover	onMouseover	Όταν ο εκάστοτε χρήστης κινεί το δείκτη του ποντικιού πάνω από ένα αντικείμενο (πχ εικόνα, κουμπί, σύνδεσμο υπερκειμένου) μίας Ιστοσελίδας.
Mouseup	onMouseup	Όταν ο εκάστοτε χρήστης μίας Ιστοσελίδας ελευθερώνει ένα πλήκτρο του ποντικιού.
Reset	onReset	Όταν ο εκάστοτε χρήστης σβήνει τα περιεχόμενα των πεδίων μίας φόρμας μίας Ιστοσελίδας.
Resize	onResize	Όταν ο εκάστοτε χρήστης αλλάζει το μέγεθος ενός παραθύρου μίας Ιστοσελίδας.
Select	onSelect	Όταν ο εκάστοτε χρήστης επιλέγει το περιεχόμενο ενός αντικειμένου μίας Ιστοσελίδας (πχ ενός πεδίου κειμένου μίας φόρμας).
Submit	onSubmit	Όταν ο εκάστοτε χρήστης υποβάλλει τα περιεχόμενα μίας φόρμας μίας Ιστοσελίδας.
Unload	onUnload	Όταν ο εκάστοτε χρήστης εγκαταλείπει την εκάστοτε τρέχουσα Ιστοσελίδα, για να επισκεφθεί μία άλλη.

Πίνακας 11: Τα πιο βασικά συμβάντα και οι αντίστοιχοι χειριστές συμβάντων της JavaScript.

[41] [42] [44] [46]

2.12.8 Τα Αντικείμενα της JavaScript

2.12.8.1 Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό

Είναι προφανές ότι η γλώσσα JavaScript υποστηρίζει αντικειμενοστραφή προγραμματισμό. Η αλήθεια είναι πως η JavaScript περιέχει αντικειμενοστραφή στοιχεία, χωρίς να φθάνει, όμως, στο επίπεδο των γλωσσών Java ή C++.

Η τεχνολογία του αντικειμενοστραφούς προγραμματισμού (Object Oriented Programming-OOP) είναι μία σχετικά καινούργια τεχνολογία δημιουργίας προγραμμάτων, η οποία ξεφεύγει από τον παραδοσιακό τρόπο προγραμματισμού που ήταν διαθέσιμος παλαιότερα. Οι γλώσσες προγραμματισμού που υποστηρίζουν πλήρως τις αρχές του αντικειμενοστραφούς προγραμματισμού χαρακτηρίζονται ως γλώσσες προσανατολισμένες στα αντικείμενα (object oriented), ενώ οι γλώσσες εκείνες που υποστηρίζουν μόνο μερικές από τις αρχές αυτές χαρακτηρίζονται ως γλώσσες βασισμένες στα αντικείμενα (object based). Και στις δύο περιπτώσεις χρησιμοποιούνται νέες έννοιες και όροι στους οποίους θα ακολουθήσει λεπτομερής αναφορά. Πιο συγκεκριμένα, θα μελετηθούν οι ακόλουθες έννοιες:

- Κλάσεις.
- Αντικείμενα.
- Ιδιότητες.
- Μέθοδοι.

Η κεντρική έννοια του αντικειμενοστραφούς προγραμματισμού είναι η κλάση (class). Μία κλάση αποτελεί μία γενική φόρμα κατασκευής αντικειμένων που διαθέτουν ανάλογες ιδιότητες και μεθόδους. Με άλλα λόγια, η κλάση αποτελεί μία «συνταγή» δημιουργίας αντικειμένων. Από την καθημερινή ζωή μπορεί να φανταστεί κανείς την κλάση αυτοκίνητο ως την έννοια που περιλαμβάνει διάφορες ιδιότητες, όπως είναι, για παράδειγμα, ο κατασκευαστής, το μοντέλο, το έτος κατασκευής, η ιπποδύναμη, ο αριθμός θυρών, καθώς και ποικίλες μεθόδους, όπως είναι, για παράδειγμα το ξεκίνημα, το σταμάτημα, η επιτάχυνση, η επιβράδυνση, η στροφή αριστερά, η στροφή δεξιά, η οπισθοπορεία. Αντικείμενα αυτής της κλάσης είναι εφικτό, για παράδειγμα, ένα VW Passat μοντέλο 2000, 106 HP, ή ένα Toyota Avensis μοντέλο 1999, 98 HP.

Οι πλήρως αντικειμενοστραφείς γλώσσες, όπως η C++ και η Java, εκτός από μία πληθώρα έτοιμων κλάσεων που διαθέτουν, επιτρέπουν στον εκάστοτε προγραμματιστή να φτιάξει και τις δικές του κλάσεις με τη χρήση της δεσμευμένης λέξης class. Η γλώσσα JavaScript δε διαθέτει τέτοια δεσμευμένη λέξη, αλλά επιτρέπει στον προγραμματιστή αφενός μεν να δουλεύει με αντικείμενα και αφετέρου να δημιουργεί νέα με τη βοήθεια συναρτήσεων, όπως θα φανεί στη συνέχεια.

Στην καθημερινή ορολογία ως αντικείμενο χαρακτηρίζεται οποιοδήποτε πράγμα βρίσκεται στο γύρω κόσμο, όπως είναι, για παράδειγμα, ένα αυτοκίνητο, ένα βιβλίο, μία τηλεόραση, ένα χαρτονόμισμα. Κάτι ανάλογο συμβαίνει και στον προγραμματισμό: ως αντικείμενο (object) νοείται μία δομή δεδομένων η οποία διαθέτει ιδιότητες και μεθόδους.

Τα αντικείμενα που μπορεί να χρησιμοποιήσει η γλώσσα JavaScript είτε βρίσκονται στο εκάστοτε πρόγραμμα περιήγησης, όπως είναι, για παράδειγμα το έγγραφο (document) μίας Ιστοσελίδας που φορτώνεται από ένα πρόγραμμα περιήγησης, το παράθυρο (window) ενός περιηγητή, ένα πλαίσιο (frame) μίας Ιστοσελίδας, το κυρίως μέρος ενός εγγράφου HTML (body), είτε αποτελούν στοιχεία της ίδιας της γλώσσας, όπως είναι, για παράδειγμα, ο πίνακας (Array), η ημερομηνία (Date), τα αντικείμενα Math και String.

Οι ιδιότητες (properties) ενός αντικειμένου περιγράφουν ένα αντικείμενο με τρόπο ανάλογο με εκείνον που ένα επίθετο προσδιορίζει ένα ουσιαστικό (πχ κόκκινο αυτοκίνητο). Για παράδειγμα, η ιδιότητα closed του αντικειμένου window προσδιορίζει, αν ένα παράθυρο είναι κλειστό ή ανοικτό. Για να πραγματοποιηθεί αναφορά σε μία ιδιότητα ενός αντικειμένου γράφεται πρώτα αυτό το αντικείμενο, έπειτα τοποθετείται μία τελεία και τέλος γράφεται η ιδιότητα του συγκεκριμένου αντικειμένου. Συνεπώς η προαναφερθείσα ιδιότητα ακολουθεί τη σύνταξη:

```
window.closed;
```

Οι μέθοδοι (methods) ενός αντικειμένου καθορίζουν τον τρόπο με τον οποίο αυτό συμπεριφέρεται ή ενεργεί, όπως ακριβώς τα ρήματα καθορίζουν τη συμπεριφορά ενός ουσιαστικού (πχ το αυτοκίνητο κινείται, στρίβει, σταματά). Για παράδειγμα το αντικείμενο Window της γλώσσας JavaScript διαθέτει τη μέθοδο open(), η οποία ανοίγει ένα παράθυρο μίας Ιστοσελίδας. Ας προσεχθεί ότι κάθε μέθοδος ακολουθείται από ένα ζεύγος παρενθέσεων μέσα στις οποίες τοποθετούνται οι παράμετροι ή τα ορίσματα της συγκεκριμένης μεθόδου. Ο τρόπος που γίνεται χρήση μίας μεθόδου ενός αντικειμένου είναι παρόμοιος με τον τρόπο που χρησιμοποιείται μία ιδιότητα αυτού του αντικειμένου. Γράφεται, δηλαδή, αρχικά το εν λόγω αντικείμενο, κατόπιν τοποθετείται μία τελεία και τέλος γράφεται η μέθοδος του παρόντος αντικειμένου. Ωστόσο, μπορεί να γράφεται απλώς μόνο η μέθοδος ενός αντικειμένου, αν αυτή αναφέρεται στο εκάστοτε παρόν αντικείμενο. Οπότε η μέθοδος που αναφέρθηκε πιο πάνω συντάσσεται ως εξής:

```
window.open();
```

ή, αν πραγματοποιείται αναφορά στο εκάστοτε τρέχον αντικείμενο:

```
open();
```

[46]

2.12.8.2 Τα Αντικείμενα

Τα αντικείμενα της γλώσσας JavaScript, όπως εξάλλου προαναφέρθηκε, είτε είναι ενσωματωμένα στη γλώσσα όπως είναι, για παράδειγμα, τα αντικείμενα Array, String, είτε δημιουργούνται από

το εκάστοτε πρόγραμμα περιήγησης, όπως είναι, για παράδειγμα, τα αντικείμενα Window, Document. Κάθε είδους αντικείμενο στη JavaScript έχει τρία κύρια χαρακτηριστικά όσον αφορά τον τρόπο ορισμού και αναφοράς του, τα οποία είναι τα ακόλουθα:

- Η δεσμευμένη λέξη `this`: Χρησιμοποιείται, για να δηλώσει το εκάστοτε παρόν αντικείμενο, κατά τους ορισμούς αντικειμένων ή συναρτήσεων ή μέσα σε χειριστές συμβάντων ενός JavaScript σεναρίου.
- Η δεσμευμένη λέξη `with`: Προσδιορίζει αντικείμενα ενός JavaScript σεναρίου, ειδικά, όταν αυτά είναι ένθετα μέσα σε άλλα αντικείμενα και ακολουθείται από μία ομάδα δηλώσεων που περικλείεται σε άγκιστρα, δηλαδή έχει τη σύνταξη:

```
with (object)  
{  
statements  
}
```

Όπου:

- `object`: Το εκάστοτε αντικείμενο ενός JavaScript σεναρίου που είναι επιθυμητό να προσδιοριστεί με τη δεσμευμένη λέξη `with`.
 - `statements`: Οι εκάστοτε δηλώσεις ενός JavaScript σεναρίου με τις οποίες γίνονται εκχωρήσεις που αποδίδονται στο `object`. Οι δηλώσεις μπορεί να είναι αποκλειστικά μόνο μία και τότε το ζεύγος των αγκιστρών {} είναι προαιρετικό.
- Η συνάρτηση-κατασκευαστής (function-constructor): Αφού οριστεί μία JavaScript συνάρτηση, τότε μπορεί να εκμεταλλευτεί η λέξη `new` για να δημιουργηθούν τα αντικείμενά της. Το πρότυπο ECMAScript περιλαμβάνει τους παρακάτω βασικότερους κατασκευαστές αντικειμένων (constructors) της γλώσσας JavaScript:
 - `Array`: Επιτρέπει τη δημιουργία πινάκων στη γλώσσα JavaScript.
 - `Boolean`: Δημιουργεί μία λογική τιμή στη γλώσσα JavaScript.
 - `Date`: Επιτρέπει το χειρισμό ημερομηνιών στη γλώσσα JavaScript.
 - `Function`: Δημιουργεί μία νέα JavaScript συνάρτηση.
 - `Global`: Περιέχει όλες τις καθολικές μεταβλητές, όπως είναι, για παράδειγμα, οι `Infinity`, `Nan` και όλες τις ενσωματωμένες συναρτήσεις της γλώσσας JavaScript, όπως είναι, για παράδειγμα, οι `eval`, `isNaN`.
 - `Math`: Περιέχει βασικές μαθηματικές συναρτήσεις και σταθερές της γλώσσας JavaScript.
 - `Number`: Αντιπροσωπεύει τα αριθμητικά δεδομένα της γλώσσας JavaScript.
 - `Object`: Αντικείμενο του οποίου τις ιδιότητες και τις μεθόδους κληρονομούν όλα τα αντικείμενα της γλώσσας JavaScript.
 - `String`: Επιτρέπει το χειρισμό αλφαριθμητικών στη γλώσσα JavaScript.

Στη συνέχεια θα εξεταστούν δύο από τα σημαντικότερα ενσωματωμένα αντικείμενα της γλώσσας JavaScript, που περιλαμβάνονται στο πρότυπο ECMAScript, το `Array` και το `String`.

[46]

2.12.8.3 Το Αντικείμενο `Array`

Ένας πίνακας (`array`) στις γλώσσες σεναρίων είναι μία ομάδα από θέσεις μνήμης οι οποίες έχουν κοινό όνομα και διακρίνονται η μία από την άλλη με τη χρήση ενός μοναδικού αριθμού, ο οποίος ονομάζεται δείκτης (`subscript\index`). Στις θέσεις ενός τέτοιου πίνακα αποθηκεύονται ομοειδή δεδομένα, αν και αυτό δεν είναι δεσμευτικό στη γλώσσα JavaScript. Η JavaScript είναι γλώσσα που παρέχει ελευθερίες όσον αφορά τους τύπους δεδομένων (`loosely typed`) και επιτρέπει την αποθήκευση διαφορετικών τύπων δεδομένων στον ίδιο πίνακα.

Κάθε θέση ενός πίνακα στη γλώσσα JavaScript προσδιορίζεται από το όνομα του πίνακα και από ένα δείκτη ο οποίος τοποθετείται μέσα σε αγκύλες []. Ο δείκτης του πρώτου στοιχείου ενός πίνακα είναι πάντα το 0. Έτσι, η N-οστή (όπου N οποιοσδήποτε φυσικός αριθμός) θέση ενός πίνακα με το όνομα, για παράδειγμα, `Array` συμβολίζεται `Array[N-1]`.

Για να δημιουργηθεί ένας νέος πίνακας, που στην ουσία είναι ένα νέο αντικείμενο Array στη γλώσσα JavaScript, και να δεσμευτούν οι κατάλληλες θέσεις μνήμης, χρησιμοποιείται η δεσμευμένη λέξη new σύμφωνα με την κατώτερη σύνταξη:

```
var arrayName;  
arrayName=new Array(N);
```

Όπου:

- arrayName: Το όνομα που είναι επιθυμητό να έχει ο καινούργιος πίνακας ο οποίος ορίζεται σε κάθε περίπτωση.
- N: Ο συνολικός αριθμός των θέσεων του πίνακα arrayName.

Η παραπάνω σύνταξη έχει και πιο σύντομη μορφή, η οποία είναι η ακόλουθη:

```
var arrayName=new Array(N);
```

Αφού οριστεί ένας πίνακας, το επόμενο βήμα είναι να του εκχωρηθούν τα δεδομένα που θα περιέχει. Αυτό γίνεται συνήθως με ξεχωριστές εντολές εκχώρησης, οι οποίες έχουν σύνταξη όμοια με την πιο κάτω:

```
arrayName[0]=a0;  
arrayName[1]=a1;  
arrayName[2]=a2;  
arrayName[3]=a3;  
.....  
arrayName[N]=an;
```

Όπου a₀, a₁, a₂, a₃, ..., a_n οι τιμές που καταχωρούνται στις θέσεις 0,1,2,3, ..., N, αντίστοιχα, του πίνακα με όνομα arrayName.

Υπάρχει βέβαια η δυνατότητα οι τιμές που είναι επιθυμητο να κρατάει ένας πίνακας να του αποδίδονται κατά τον ορισμό του, σύμφωνα με την ακόλουθη σύνταξη:

```
var arrayName=new Array(a0, a1, a2, a3, ..., an);
```

Η πιο σημαντική, τώρα, ιδιότητα του αντικείμενου Array της γλώσσας JavaScript είναι η ιδιότητα length. Με την εν λόγω ιδιότητα επιστρέφεται ή καθορίζεται το μήκος του εκάστοτε πίνακα, δηλαδή το πλήθος των στοιχείων που αυτός περιέχει. Έτσι, προκειμένου να επιστραφεί το μήκος του πίνακα με όνομα arrayName, θα ακολουθηθεί η εξής σύνταξη:

```
arrayName.length;
```

Ενώ, για να καθοριστεί το μήκος του ανωτέρου πίνακα θα γραφεί η επόμενη σύνταξη:

```
arrayName.length=number;
```

Όπου number ένας θετικός ακέραιος αριθμός που καθορίζει το μήκος ενός πίνακα στη γλώσσα JavaScript

Επιπρόσθετα, το αντικείμενο Array της γλώσσας JavaScript διαθέτει σημαντικές μεθόδους με τις οποίες γίνεται χειρισμός των δεδομένων των πινάκων που απαντώνται στον JavaScript κώδικα μίας Ιστοσελίδας. Οι πιο σημαντικές μέθοδοι του εν λόγω αντικείμενου παρουσιάζονται στον επόμενο Πίνακα:

Μέθοδος	Λειτουργία	Σύνταξη	Παράμετρος (-οι)	Τιμή Επιστροφής
concat()	Συνενώνει τα δεδομένα δύο ή περισσότερων πινάκων σε ένα νέο πίνακα.	array.concat(array1,array2,array3,...,arrayN)	array1,array2,array3,...,arrayN: Υποχρεωτικές. Οι πίνακες 1,2,3,...,N, αντίστοιχα, που πρόκειται να συνενωθούν.	Ένα αντικείμενο Array, που αναπαριστά τους πίνακες που συνενώθηκαν.
join()	Συνενώνει όλα τα στοιχεία ενός πίνακα και σχηματίζει ένα αλφαριθμητικό.	array.join(separator)	separator: Προαιρετική. Ένα διαχωριστικό, που πρόκειται να χρησιμοποιηθεί τη μεταξύ των στοιχείων ενός πίνακα, όταν τα εν λόγω στοιχεία συνενωθούν, ώστε να σχηματίσουν ένα αλφαριθμητικό. Αν η παράμετρος separator παραλείπεται, τότε τα στοιχεία του προαναφερόμενου πίνακα διαχωρίζονται μεταξύ τους με κόμμα (,).	Ένα αντικείμενο String, που αναπαριστά ένα αλφαριθμητικό, το οποίο προέκυψε από τη συνένωση των στοιχείων ενός πίνακα, τα οποία διαχωρίστηκαν μεταξύ τους με την παράμετρο separator.
pop()	Αφαιρεί το τελευταίο στοιχείο ενός πίνακα	array.pop()	Καμία.	Οτιδήποτε αναπαριστά το τελευταίο στοιχείο ενός πίνακα, που αφαιρέθηκε.
push()	Προσθέτει ένα ή περισσότερα στοιχεία στο τέλος ενός πίνακα.	array.push(item1,item2,item3,...,itemN)	item1,item2,item3,...,itemN: Υποχρεωτικές. Τα στοιχεία 1,2,3,...,N, αντίστοιχα, που πρόκειται να προστεθούν στο τέλος ενός πίνακα.	Ένα αντικείμενο Number, που αναπαριστά το νέο μέγεθος ενός πίνακα το οποίο προέκυψε έπειτα από την προσθήκη ενός ή περισσότερων στοιχείων στο τέλος του εν λόγω πίνακα.
reverse()	Αντιστρέφει τα στοιχεία ενός πίνακα.	array.reverse()	Καμία.	Ένα αντικείμενο Array, που αναπαριστά έναν πίνακα, ο οποίος αντιστράφηκε.
shift()	Αφαιρεί το πρώτο στοιχείο ενός πίνακα.	array.shift()	Καμία.	Οτιδήποτε αναπαριστά το πρώτο στοιχείο ενός πίνακα, που αφαιρέθηκε.
slice()	Επιστρέφει ένα επιλεγμένο μέρος ενός πίνακα.	array.slice(start, end)	i) start: Υποχρεωτική. Ένας θετικός ακέραιος αριθμός, που καθορίζει μία θέση ε-	Ένα αντικείμενο Array, που περιέχει τα στοιχεία ε-

			<p>νός πίνακα, από την οποία πρόκειται να ξεκινήσει επιλογή στοιχείων του εν λόγω πίνακα.</p> <p>ii) end: Υποχρεωτική. Ένας ακέραιος αρνητικός αριθμός, που καθορίζει μία θέση ενός πίνακα, στην οποία πρόκειται να σταματήσει η επιλογή στοιχείων του εν λόγω πίνακα, η οποία ξεκίνησε από τη θέση αυτού του πίνακα, η οποία καθορίστηκε με την παράμετρο start.</p> <p>(Μπορεί να γίνει χρήση αρνητικών ακεραίων αριθμών, ώστε να ξεκινά επιλογή στοιχείων από το τέλος ενός πίνακα.)</p>	<p>νός πίνακα που επιλέχθηκαν.</p>
splice()	<p>Προσθέτει ή διαγράφει στοιχεία σε ή από έναν πίνακα.</p>	<p>array.splice(index,howmany,item1,item2,item3,...,itemN)</p>	<p>i) index: Υποχρεωτική. Ένας θετικός ακέραιος αριθμός που καθορίζει τη θέση ενός πίνακα, από την οποία πρόκειται να ξεκινήσει προσθήκη ή διαγραφή στοιχείων του εν λόγω πίνακα.</p> <p>(Μπορεί να γίνει χρήση ακεραίων αρνητικών αριθμών, ώστε να ξεκινά προσθήκη ή διαγραφή στοιχείων από το τέλος ενός πίνακα.)</p> <p>ii) howmany: Υποχρεωτική. Ένας μη αρνητικός ακέραιος αριθμός, που καθορίζει το πλήθος των στοιχείων ενός πίνακα, τα οποία πρόκειται να διαγραφούν. Αν ο εν λόγω αριθμός είναι ίσος με 0, τότε δε θα διαγραφεί κανένα στοιχείο από τον παρπάνω πίνακα.</p>	<p>Ένα αντικείμενο Array, που περιέχει τα στοιχεία ενός πίνακα που τυχόν διαγράφηκαν</p>

			ii) item1,item2,item3, ...,itemN: Προαιρετικές. Τα νέα στοιχεία 1,2,3,...,N, αντίστοιχα, που πρόκειται να προστεθούν σε έναν πίνακα.	
sort()	Ταξινομεί τα στοιχεία ενός πίνακα. Η σειρά ταξινόμησης μπορεί να είναι είτε αλφαβητική είτε αριθμητική, και είτε αύξουσα είτε φθίνουσα. Προεπιλεγμένα η μέθοδος sort() ταξινομεί τα στοιχεία ενός πίνακα σε αλφαβητική και αύξουσα σειρά θεωρώντας πάντα ως αλφαριθμητικά (ακόμα και αν είναι αριθμοί, οπότε σε αυτή την περίπτωση, επειδή προκύπτει λανθασμένη ταξινόμηση, κρίνεται επιτακτική μία εναλλακτική μέθοδος ταξινόμησης).	array.sort(compareFunction)	compareFunction: Προαιρετική. Μία συνάρτηση, που πρόκειται να ταξινομήσει με έναν εναλλακτικό τρόπο τα στοιχεία ενός πίνακα, όταν αυτά είναι αριθμοί. Έτσι, η συνάρτηση αυτή δέχεται ως παραμέτρους τις παραμέτρους της μεθόδου sort() και υπολογίζει ανά δύο τη διαφορά των συγκεκριμένων παραμέτρων, επιστρέφοντας για κάθε έναν από τους προαναφερομένους υπολογισμό μία αρνητική, ή μία μηδενική, ή μία θετική τιμή με βάση την οποία γίνεται διαδοχικά η ταξινόμηση όλων των στοιχείων του παραπάνω πίνακα.	Ένα αντικείμενο Array, που περιέχει τα στοιχεία ενός πίνακα, τα οποία ταξινομήθηκαν.
unshift()	Προσθέτει ένα ή περισσότερα στοιχεία στην αρχή ενός πίνακα.	array.unshift(item1,item2,item3,...,itemN)	item1,item2,item3,..., itemN: Υποχρεωτικές. Τα στοιχεία 1, 2,3,...,N, αντίστοιχα, που πρόκειται να προστεθούν στην αρχή ενός πίνακα.	Ένα αντικείμενο Number, που αναπαριστά το νέο μέγεθος ενός πίνακα, το οποίο προέκυψε έπειτα από την προσθήκη ενός ή περισσότερων στοιχείων στην αρχή του εν λόγω πίνακα.

Πίνακας 12: Οι πιο βασικές μέθοδοι του αντικειμένου Array της JavaScript

[36] [41][46]

2.12.8.4 Το Αντικείμενο String

Το αντικείμενο String της γλώσσας JavaScript διαθέτει μία σειρά ιδιοτήτων και μεθόδων που παρέχουν την ευχέρεια στον εκάστοτε προγραμματιστή Ιστοσελίδων να δημιουργεί και να χειρίζεται αλφαριθμητικά δεδομένα.

Για τη δημιουργία ενός νέου αλφαριθμητικού αντικειμένου στη γλώσσα JavaScript διατίθεται η δεσμευμένη λέξη `new` σύμφωνα με την ακόλουθη σύνταξη:

```
var stringName;  
stringName=new String("str");
```

Όπου:

- `stringName`: Το όνομα που είναι επιθυμητό να έχει το νέο αλφαριθμητικό που ορίζεται σε κάθε περίπτωση.
- `str`: Η τιμή που καταχωρείται στο νέο αλφαριθμητικό που ορίζεται σε κάθε περίπτωση.

Η παραπάνω σύνταξη έχει και πιο σύντομη μορφή, η οποία είναι η ακόλουθη:

```
var stringName=new String("str");
```

Συνήθως, όμως, δημιουργείται ένα νέο αλφαριθμητικό εκχωρώντας του απευθείας την τιμή, σύμφωνα με τη σύνταξη:

```
var stringName="str";
```

Η πιο σημαντική, τώρα, ιδιότητα του αντικειμένου `String` της γλώσσας JavaScript είναι η ιδιότητα `length`. Η συγκεκριμένη ιδιότητα επιστρέφει έναν αριθμό που αναπαριστά το μήκος ενός αλφαριθμητικού, δηλαδή το σύνολο των χαρακτήρων που απαρτίζουν τον εν λόγω αλφαριθμητικό. Συνεπώς, προκειμένου να υπολογιστεί το μήκος του αλφαριθμητικού με όνομα `stringName`, θα γραφεί η μετέπειτα σύνταξη:

```
stringName.length;
```

Στο σημείο τούτο, να επισημανθεί πως το μήκος ενός κενού αλφαριθμητικού, δηλαδή ενός αλφαριθμητικού που δεν περιέχει κανένα χαρακτήρα ισούται με μηδέν, ενώ το μήκος ενός αλφαριθμητικού που ταυτίζεται με το κενό διάστημα ισούται με ένα, διότι αποτελείται από ένα χαρακτήρα, αυτόν του κενού!!!

Επιπλέον, το αντικείμενο `String` της γλώσσας JavaScript διαθέτει χρήσιμες μεθόδους με τις οποίες πραγματοποιείται ο χειρισμός των αλφαριθμητικών\συμβολοσειρών που εμφανίζονται στον JavaScript κώδικα μίας Ιστοσελίδας. Οι βασικότερες από αυτές τις μεθόδους αποτυπώνονται στον παρακάτω Πίνακα:

Μέθοδος	Λειτουργία	Σύνταξη	Παράμετρος (-οι)	Τιμή Επιστροφής
<code>charAt()</code>	Επιστρέφει ένα χαρακτήρα ενός αλφαριθμητικού, ο οποίος αντιστοιχεί σε ένα συγκεκριμένο δείκτη μέσα στο εν λόγω αλφαριθμητικό.	<code>string.charAt(index)</code>	<code>index</code> : Υποχρεωτική. Ένας ακέραιος αριθμός, που αναπαριστά ένα συγκεκριμένο δείκτη μέσα σε ένα αλφαριθμητικό, ο οποίος αντιστοιχεί σε ένα χαρακτήρα του εν λόγω αλφαριθμητικού, ο οποίος πρόκειται να επιστραφεί.	Ένα αντικείμενο <code>String</code> , που αναπαριστά ένα χαρακτήρα ενός αλφαριθμητικού, ο οποίος αντιστοιχεί στην παράμετρο <code>index</code> . Αν η παράμετρος <code>index</code> δεν αντιστοιχεί σε κάποιο χαρακτήρα ενός αλφαριθμητικού, τότε η τιμή επιστροφής είναι ο

				χαρακτήρας του κενού.
charCodeAt()	Επιστρέφει την Unicode τιμή ενός χαρακτήρα ενός αλφαριθμητικού, ο οποίος αντιστοιχεί σε ένα συγκεκριμένο δείκτη μέσα στο εν λόγω αλφαριθμητικό.	string.charCodeAt(index)	index: Υποχρεωτική. Ένας ακέραιος αριθμός, που αναπαριστά ένα συγκεκριμένο δείκτη μέσα σε ένα αλφαριθμητικό, ο οποίος αντιστοιχεί σε ένα χαρακτήρα του εν λόγω αλφαριθμητικού, που η Unicode τιμή του πρόκειται να επιστραφεί.	Ένα αντικείμενο Number, που αναπαριστά την Unicode τιμή ενός χαρακτήρα ενός αλφαριθμητικού, ο οποίος αντιστοιχεί στην παράμετρο index. Αν η παράμετρος index δεν αντιστοιχεί σε κάποιον χαρακτήρα ενός αλφαριθμητικού ή είναι αρνητική, τότε η τιμή επιστροφής είναι NaN.
concat()	Συνενώνει δύο ή περισσότερα αλφαριθμητικά.	string.concat(string1, string2, string3, ..., stringN)	string1, string2, string3, ..., stringN: Υποχρεωτικές. Τα αλφαριθμητικά που πρόκειται να συνενωθούν.	Ένα αντικείμενο String, που αναπαριστά τα αλφαριθμητικά τα οποία συνενώθηκαν.
fromCharCode()	Μετατρέπει Unicode τιμές σε χαρακτήρες.	string.fromCharCode(value1, value2, value3, ..., valueN)	value1, value2, value3, ..., valueN: Υποχρεωτικές. Οι Unicode τιμές 1, 2, 3, ..., N, αντίστοιχα, που πρόκειται να μετατραπούν σε χαρακτήρες.	Ένα αντικείμενο String, που αναπαριστά τους χαρακτήρες στους οποίους μετατράπηκαν οι Unicode τιμές.
indexOf()	Επιστρέφει τη θέση μέσα σε ένα αλφαριθμητικό, στην οποία βρίσκεται για πρώτη φορά ένας συγκεκριμένος χαρακτήρας (ή μία συγκεκριμένη ακολουθία χαρακτήρων).	string.indexOf(search, start)	i) search: Υποχρεωτική. Ένας συγκεκριμένος χαρακτήρας (ή μία συγκεκριμένη ακολουθία χαρακτήρων), που πρόκειται να αναζητηθεί μέσα σε ένα αλφαριθμητικό. ii) start: Προαιρετική. Η θέση μέσα σε ένα αλφαριθμητικό, από την οποία πρόκειται να ξεκινήσει η αναζήτηση της παραμέτρου search. Αν η παράμετρος start παραλείπεται, τότε η προαναφερόμενη	Ένα αντικείμενο Number, που αναπαριστά τη θέση μέσα σε ένα αλφαριθμητικό, στην οποία βρέθηκε για πρώτη φορά η παράμετρος search ή -1, αν δε βρέθηκε ποτέ.

			θέση είναι προεπιλεγμένα η θέση του πρώτου χαρακτήρα του παραπάνω αλφαριθμητικού.	
lastIndexOf ()	Επιστρέφει τη θέση μέσα σε ένα αλφαριθμητικό, στην οποία βρίσκεται για τελευταία φορά ένας συγκεκριμένος χαρακτήρας (ή μία συγκεκριμένη ακολουθία χαρακτήρων).	string.lastIndexOf (search,index)	i) search: Υποχρεωτική. Ένας συγκεκριμένος χαρακτήρας (ή μία συγκεκριμένη ακολουθία χαρακτήρων), που πρόκειται να αναζητηθεί μέσα σε ένα αλφαριθμητικό. ii) start: Προαιρετική. Η θέση μέσα σε ένα αλφαριθμητικό, από την οποία πρόκειται να ξεκινήσει η αναζήτηση της παραμέτρου search (ψάχνοντας από το τέλος προς την αρχή ενός αλφαριθμητικού). Αν η παράμετρος start παραλείπεται, τότε η προαναφερόμενη θέση είναι προεπιλεγμένα η θέση του τελευταίου χαρακτήρα του παραπάνω αλφαριθμητικού.	Ένα αντικείμενο Number, που αναπαριστά τη θέση μέσα σε ένα αλφαριθμητικό, στην οποία βρέθηκε για τελευταία φορά η παράμετρος search ή -1, αν δε βρέθηκε ποτέ.
split()	Διαχωρίζει ένα αλφαριθμητικό σε έναν πίνακα από υποαλφαριθμητικά του εν λόγω αλφαριθμητικού και επιστρέφει τον πίνακα αυτόν.	string.split(separator, limit)	i) separator: Προαιρετική. Ένα διαχωριστικό που καθορίζει το χαρακτήρα ή την κανονική έκφραση που πρόκειται χρησιμοποιηθεί για το διαχωρισμό ενός αλφαριθμητικού. Αν η παράμετρος separator παραλείπεται, τότε ολόκληρο το προαναφερόμενο αλφαριθμητικό επιστρέφεται (ως ένα πίνακας που περιέχει ένα και μοναδικό αντικείμενο, το εν λόγω αλφαριθμητικό).	Ένα αντικείμενο τύπου Array, που περιέχει το αλφαριθμητικό ή τα υποαλφαριθμητικά στα οποία αυτό χωρίζεται.

			ii) limit: Προαιρετική. Ένας ακέραιος αριθμός που καθορίζει τα υποαριθμητικά οποία διαιρείται το αριθμητικό. Τα στοιχεία μετά το συγκεκριμένο αριθμό των διαιρέσεων δεν επιστρέφονται.	
substr()	Επιστρέφει ένα τμήμα ενός αλφαριθμητικού που καθορίζεται από τα ορίσματα αρχής και μήκους που δέχεται.	string.substr(start, length)	i) start: Υποχρεωτική: Η θέση του αλφαριθμητικού που αποτελεί το σημείο εκκίνησης του υποαριθμητικού που επιστρέφεται. ii) length: Προαιρετική. Ο αριθμός των χαρακτήρων που επιστρέφονται. Αν παραλείπεται, επιστρέφεται το υπόλοιπο μέρος του αλφαριθμητικού μετά το σημείο εκκίνησης.	Ένα αντικείμενο τύπου String, που περιέχει το μέρος του αλφαριθμητικού που επιστρέφεται. Αν η τιμή της length είναι μηδέν ή αρνητική, επιστρέφεται ένας κενός χαρακτήρας.
substring()	Επιστρέφει ένα τμήμα ενός αλφαριθμητικού που καθορίζεται από τα ορίσματα αρχής και τέλους που δέχεται.	string.substring(start, end)	i) start: Υποχρεωτική: Η θέση του αλφαριθμητικού που αποτελεί το σημείο εκκίνησης του υποαριθμητικού που επιστρέφεται. ii) Προαιρετική: Η θέση του αλφαριθμητικού που αποτελεί το σημείο τερματισμού του υποαριθμητικού που επιστρέφεται. Ο χαρακτήρας που αντιστοιχεί στη θέση τερματισμού δε συμπεριλαμβάνεται στο υποαριθμητικό που επιστρέφεται.	Ένα αντικείμενο τύπου String, που περιέχει το μέρος του αλφαριθμητικού που επιστρέφεται.

toLowerCase()	Μετατρέπει τα γράμματα ενός αλφαριθμητικού (τα οποία πιθανόν περιέχει και είναι κεφαλαία) σε πεζά (μικρά).	string.toLowerCase()	Καμία.	Ένα αντικείμενο String, που αναπαριστά ένα αλφαριθμητικό, του οποίου όλα τα γράμματα (τα οποία πιθανόν περιέχει και πριν της εφαρμογή της μεθόδου toLowerCase() ήταν κεφαλαία) μετατράπηκαν σε πεζά.
toString()	Επιστρέφει την τιμή ενός αλφαριθμητικού.	string.toString()	Καμία.	Ένα αντικείμενο τύπου String, που αναπαριστά την τιμή ενός αλφαριθμητικού.
toUpperCase()	Μετατρέπει τα γράμματα ενός αλφαριθμητικού (τα οποία πιθανόν περιέχει και είναι πεζά (μικρά)) σε κεφαλαία.	string.toUpperCase()	Καμία.	Ένα αντικείμενο String, που αναπαριστά ένα αλφαριθμητικό, του οποίου όλα τα γράμματα (τα οποία πιθανόν περιέχει και πριν της εφαρμογή της μεθόδου toUpperCase() ήταν πεζά) μετατράπηκαν σε κεφαλαία.
valueOf()	Επιστρέφει την πρωταρχική τιμή ενός αλφαριθμητικού.	string.valueOf()	Καμία.	Ένα αντικείμενο String, που αναπαριστά την πρωταρχική τιμή ενός αλφαριθμητικού.

Πίνακας 13: Οι πιο βασικές μέθοδοι του αντικειμένου String της JavaScript

[36] [41] [42] [46]

2.12.9 Τα Παράθυρα Διαλόγου της JavaScript

Η γλώσσα JavaScript υποστηρίζει τρεις βασικούς τύπους παραθύρων διαλόγου (dialog boxes), καθένας από τους οποίους αναλύεται διεξοδικά στη συνέχεια:

Πρώτον, ένα παράθυρο διαλόγου προειδοποίησης (warning dialog box) χρησιμοποιείται συχνά, για να ειδοποιήσει ένα χρήστη σε περίπτωση που αυτός προβεί σε οποιαδήποτε μη έγκυρη ενέργεια, κατά την προσπάθειά του εισόδου δεδομένων σε μία Ιστοσελίδα (πχ εισαγωγή άκυρων χαρακτήρων σε ένα πεδίο κειμένου).

Κάτι τέτοιο επιτυγχάνεται με τη χρήση της μεθόδου alert του αντικειμένου Window της γλώσσας JavaScript, η οποία συντάσσεται ως εξής:

```
alert(message);
```

Όπου `message` το εκάστοτε προαιρετικό όρισμα τύπου `String` της μεθόδου `alert` του αντικειμένου `Window` της γλώσσας `JavaScript` που καθορίζει την κατάλληλη σε κάθε περίπτωση προειδοποίηση, η οποία θα εμφανιστεί στο εκάστοτε παράθυρο διαλόγου προειδοποίησης μίας Ιστοσελίδας.

Παρόλα αυτά, ένα παράθυρο διαλόγου προειδοποίησης μπορεί να χρησιμοποιηθεί, ώστε να παρουσιάσει φιλικά μηνύματα απλής ειδοποίησης. Τα συγκεκριμένα παράθυρα δίνουν αποκλειστικά την επιλογή ενός κουμπιού “OK”, προκειμένου ο εκάστοτε χρήστης να προχωρήσει παρακάτω.

Δεύτερον, ένα παράθυρο διαλόγου επιβεβαίωσης (`confirmation dialog box`) διατίθεται αρκετές φορές, ώστε να δώσει ένας χρήστης τη συγκατάθεσή του, προκειμένου να εκτελεστεί σε μία Ιστοσελίδα μία συγκεκριμένη ενέργεια. Αυτό το παράθυρο διαλόγου επιβεβαίωσης έχει δύο κουμπιά, που είναι τα εξής:

- “OK”: Το πατάει ο εκάστοτε χρήστης, αν συμφωνεί να δώσει την συγκατάθεσή του, προκειμένου να εκτελεστεί σε μία Ιστοσελίδα μία συγκεκριμένη ενέργεια.
- “Cancel”: Το πατάει ο εκάστοτε χρήστης, αν διαφωνεί να δώσει την συγκατάθεσή του, προκειμένου να εκτελεστεί σε μία Ιστοσελίδα μία συγκεκριμένη ενέργεια.

Τα παραπάνω υλοποιούνται με τη βοήθεια της μεθόδου `confirm` του αντικειμένου `Window` της γλώσσας `JavaScript`, η οποία συντάσσεται ως εξής:

```
confirm(message);
```

Όπου `message` το εκάστοτε προαιρετικό όρισμα τύπου `String` της μεθόδου `confirm` του αντικειμένου `Window` της γλώσσας `JavaScript` που καθορίζει την πρόταση σε κάθε περίπτωση επιβεβαίωσης, που θα εμφανιστεί στο εκάστοτε παράθυρο διαλόγου επιβεβαίωσης μίας Ιστοσελίδας.

Υπεισέρχοντας στις τεχνικές λεπτομέρειες της μεθόδου `confirm` να αναφερθεί ότι επιστρέφει μία λογική τιμή, ανάλογα, αν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατήσει το κουμπί “OK” ή “Cancel” ενός παραθύρου διαλόγου επιβεβαίωσης. Για την εν λόγω τιμή επιστροφής ισχύουν τα μετέπειτα:

- Αν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατήσει το κουμπί “OK” ενός παραθύρου διαλόγου επιβεβαίωσης, τότε η λογική τιμή της τιμής επιστροφής της μεθόδου `confirm` γίνεται `true`.
- Αν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατήσει το κουμπί “Cancel” ενός παραθύρου διαλόγου επιβεβαίωσης, τότε η λογική τιμή της τιμής επιστροφής της μεθόδου `confirm` γίνεται `false`.

Για την πλήρη ανάπτυξη ενός παραθύρου διαλόγου επιβεβαίωσης πρέπει, πλην μεθόδου `confirm`, να γίνεται χρήση και της μεθόδου `alert` του αντικειμένου `Window` της γλώσσας `JavaScript`, ώστε, όταν κρίνεται αναγκαίο, να παρουσιάζεται στην εκάστοτε Ιστοσελίδα το αρμόζων μήνυμα προειδοποίησης.

Τρίτον, ένα παράθυρο διαλόγου προτροπής (`prompt dialog box`) είναι πολύ χρήσιμο, ιδιαίτερα όταν θέλουμε ένας χρήστης να αλληλεπιδρά με το περιεχόμενο μίας Ιστοσελίδας, με τη χρήση ενός αναδυόμενου πεδίου κειμένου στο οποίο αυτός ο χρήστης θα εισάγει δεδομένα. Το συγκεκριμένο παράθυρο διαλόγου προτροπής έχει δύο κουμπιά:

- “OK”: Το πατάει ο εκάστοτε χρήστης, για να επαληθεύσει την εισαγωγή δεδομένων σε ένα παράθυρο διαλόγου προτροπής μίας Ιστοσελίδας.
- “Cancel”: Το πατάει ο εκάστοτε χρήστης, προκειμένου να ακυρώσει την εισαγωγή δεδομένων σε ένα παράθυρο διαλόγου προτροπής μίας Ιστοσελίδας.

Τα ανώτερα πραγματοποιούνται με τη μέθοδο `prompt`, η οποία συντάσσεται ως εξής:

```
prompt(text,defaultText);
```

Όπου:

- **text:** Το εκάστοτε απαραίτητο όρισμα της μεθόδου `prompt` του αντικειμένου `Window` της γλώσσας JavaScript, που δηλώνει τη χρήσιμη σε κάθε περίπτωση προτροπή προς τον εκάστοτε χρήστη που θα εμφανιστεί στο εκάστοτε παράθυρο διαλόγου προτροπής μίας Ιστοσελίδας.
- **defaultText:** Το εκάστοτε προαιρετικό όρισμα τύπου `String` της μεθόδου `prompt` του αντικειμένου `Window` της γλώσσας JavaScript που σχετίζεται με το σε κάθε περίπτωση προκαθορισμένο κείμενο που θα εισάγει ο εκάστοτε χρήστης στο πεδίο κειμένου του εκάστοτε παραθύρου διαλόγου προτροπής μίας Ιστοσελίδας

Μπαίνοντας στις τεχνικές λεπτομέρειες τις μεθόδου `prompt` να αναφερθεί πως έχει μία τιμή επιστροφής, τύπου `String`, ανάλογα, αν εκάστοτε χρήστης μίας Ιστοσελίδας πατήσει το κουμπί “OK” ή “Cancel” ενός παραθύρου διαλόγου προτροπής: Για την προαναφερόμενη τιμή επιστροφής ισχύουν τα παρακάτω:

- Αν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατήσει το κουμπί “OK” ενός παραθύρου διαλόγου προτροπής, τότε η λογική τιμή της τιμής επιστροφής της μεθόδου `prompt` ταυτίζεται με το κείμενο που εισήγαγε αυτός ο χρήστης.
- Αν ο εκάστοτε χρήστης μίας Ιστοσελίδας πατήσει το κουμπί “Cancel” ενός παραθύρου διαλόγου προτροπής, τότε η λογική τιμή της τιμής επιστροφής της μεθόδου `prompt` γίνεται `null`.

Για την πλήρη υλοποίηση ενός παραθύρου διαλόγου προτροπής απαιτείται η μέθοδος `prompt` να συνδυάζεται με τη μέθοδο `alert` του αντικειμένου `Window` της γλώσσας JavaScript, ώστε, όποτε είναι ανάγκη να εμφανίζεται στην εκάστοτε Ιστοσελίδα το σχετικό μήνυμα προειδοποίησης.

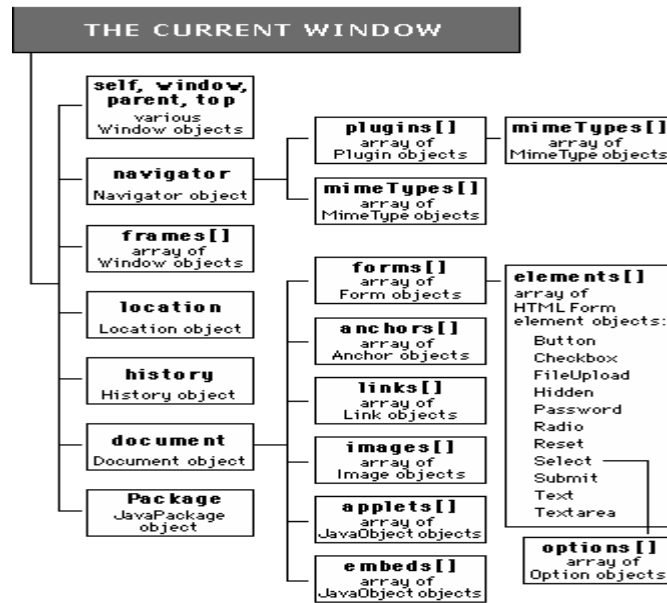
[41] [43]

2.12.10 Η JavaScript στα Προγράμματα Περιήγησης

2.12.10.1 Εισαγωγή

Για να κατανοηθεί ο τρόπος δράσης ενός κώδικα JavaScript σε ένα πρόγραμμα περιήγησης, πρέπει να δοθεί έμφαση σε τρία βασικά θέματα:

- I. Στην έννοια του αντικειμένου `Window`: Το αντικείμενο `Window` ενός HTML εγγράφου λειτουργεί ως υποδοχέας, άλλων αντικειμένων καθώς και ως χώρος εκτέλεσης του κώδικα JavaScript στον υπολογιστή του χρήστη. Ο κύριος σκοπός ενός προγράμματος περιήγησης στο Διαδίκτυο είναι να εμφανίζει τα HTML έγγραφα σε ένα παράθυρο. Για το λόγο αυτόν το αντικείμενο `Window` ενός HTML εγγράφου έχει τη μεγαλύτερη σπουδαιότητα, αφού περικλείει το αντικείμενο `Document` το οποίο αντιστοιχεί στο προαναφερόμενο HTML έγγραφο, και επειδή σε αυτό εκτελείται ο κώδικας JavaScript.
- II. Στο Μοντέλο Αντικειμένου Εγγράφου (`Document Object Model-DOM`): Τα αντικείμενα ενός HTML εγγράφου που δημιουργούνται σε ένα πρόγραμμα περιήγησης συνιστούν ένα ιεραρχικό μοντέλο αντικειμένων, η βάση του οποίου είναι το αντικείμενο `Window`. Αυτή η ιεραρχία των αντικειμένων φαίνεται στην παρακάτω Εικόνα:



Εικόνα 17: Η ιεραρχία των αντικειμένων στο Μοντέλο Αντικειμένου Εγγράφου

Να τονιστεί, σε αυτό το σημείο, πως σύμφωνα με την παραπάνω ιεραρχία τα αντικείμενα ενός HTML εγγράφου μπορούν να προσπελαστούν ως ιδιότητες άλλων αντικειμένων. Έτσι, για παράδειγμα, για να γίνει αναφορά στην πρώτη φόρμα ενός HTML εγγράφου που παρουσιάζεται στο εκάστοτε τρέχον παράθυρο ενός πλοηγού ακολουθείται η σύνταξη:

`window.document.forms[0]`

III. Στο μοντέλο του προγραμματισμού ο οποίος καθοδηγείται από συμβάντα (event driven programming): Παραπάνω εξετάστηκαν τα συμβάντα (events) που υποστηρίζει η γλώσσα JavaScript καθώς και οι αντίστοιχους χειριστές συμβάντων (event handlers). Στις παλιές διαδικαστικές γλώσσες προγραμματισμού (procedural languages), η πορεία ενός προγράμματος ακολουθούσε μία προκαθορισμένη διαδρομή και η ίδια η εφαρμογή έλεγχε τη ροή του προγράμματος καθώς και τα τμήματα του κώδικα που είχε σκοπό να εκτελέσει. Στον προγραμματισμό που καθοδηγείται από συμβάντα η ροή ενός προγράμματος καθορίζεται από τα συμβάντα, και ο κώδικας, ο οποίος κάθε φορά εκτελείται, εξαρτάται από το συμβάν που επιλέχθηκε από τον εκάστοτε χρήστη. Έτσι, μία εφαρμογή τέτοιου τύπου περιέχει διάφορους μηχανισμούς ελέγχου (πχ κουμπιά, πλαίσια, μενού, λίστες), οι οποίοι με τη σειρά τους περιέχουν συμβάντα (πχ πάτημα, διπλόπατημα), που μπορεί να ενεργοποιήσει ο εκάστοτε χρήστης κατά βούληση. Όταν ενεργοποιείται ένα συμβάν, το εκάστοτε πρόγραμμα περιήγησης καλεί μία ειδική συνάρτηση που χειρίζεται το συμβάν αυτό (το χειριστή συμβάντος) και η οποία εκτελεί μία καθορισμένη εργασία. Με αυτό το είδος προγραμματισμού αποφεύγεται η δημιουργία μονολιθικών εφαρμογών, αλλά κατασκευάζονται εφαρμογές που αποτελούνται από πολλά τμήματα κώδικα.

[41]

2.12.10.2 Το Αντικείμενο Window

Για κάθε παράθυρο ενός προγράμματος πλοήγησης Παγκόσμιο Ιστό που εμφανίζεται στην οθόνη του εκάστοτε χρήστη, όταν ο αυτός ο χρήστης επισκέπτεται ένα Ιστότοπο δημιουργείται ένα αντικείμενο Window της γλώσσας JavaScript. Οι χρησιμότερες ιδιότητες του εν λόγω αντικειμένου καταγράφονται στον επόμενο Πίνακα:

Ιδιότητα	Περιγραφή
closed	Παίρνει λογικές τιμές και ελέγχει, αν ένα παράθυρο μίας Ιστοσελίδας είναι κλειστό ή όχι ,επιστρέφοντας την τιμή true ή false, αντίστοιχα.

defaultStatus	Καθορίζει ή επιστρέφει το κείμενο που εμφανίζεται εξ ορισμού σε μία γραμμή κατάστασης ενός παραθύρου μίας Ιστοσελίδας.
frames	Επιστρέφει έναν πίνακα αντικειμένων που περιέχει τα πλαίσια τα οποία τυχόν υπάρχουν σε ένα παράθυρο μίας Ιστοσελίδας. Χρησιμοποιείται συχνά για την προσπέλαση των ιδιοτήτων των προαναφερόμενων πλαισίων.
history	Επιστρέφει το αντικείμενο History, το οποίο κρατά το ιστορικό των περιηγήσεων ενός προγράμματος περιήγησης.
length	Επιστρέφει το πλήθος των πλαισίων που τυχόν υπάρχουν σε ένα παράθυρο μίας Ιστοσελίδας.
location	Επιστρέφει τη URL του παραθύρου μίας Ιστοσελίδας. Η τοποθέτηση μίας νέας URL στην ιδιότητα location έχει αποτέλεσμα το άνοιγμα ενός νέου παραθύρου.
name	Επιστρέφει το όνομα ενός παραθύρου μίας Ιστοσελίδας και επιτρέπει την απόδοση ενός νέου ονόματος σε αυτό το παράθυρο.
opener	Επιστρέφει μία αναφορά προς εκείνο το παράθυρο από το οποίο άνοιξε το εκάστοτε τρέχον παράθυρο μίας Ιστοσελίδας.
parent	Αν ένα παράθυρο μίας Ιστοσελίδας είναι πλαίσιο, τότε η συγκεκριμένη ιδιότητα επιστρέφει μία αναφορά προς το «γονικό» πλαίσιο που περιέχει αυτό το παράθυρο.
self	Αν ένα παράθυρο μίας Ιστοσελίδας αποτελείται από πλαίσια, τότε η συγκεκριμένη ιδιότητα επιστρέφει μία αναφορά προς το εκάστοτε τρέχον πλαίσιο.
status	Επιστρέφει το κείμενο που τοποθετείται στη γραμμή κατάστασης ενός παραθύρου μίας Ιστοσελίδας. Υπερισχύει της ιδιότητας defaultStatus.
top	Αν ένα παράθυρο μίας Ιστοσελίδας αποτελείται από πλαίσια, τότε η συγκεκριμένη ιδιότητα επιστρέφει μία αναφορά προς το εκάστοτε πλαίσιο που τοποθετείται πάνω από όλα τα υπόλοιπα πλαίσια.

Πίνακας 14: Οι πιο βασικές ιδιότητες του αντικειμένου Window της JavaScript

Σχετικά, τώρα, με τις μεθόδους του αντικειμένου Window της γλώσσας JavaScript, οι δύο σημαντικότερες μέθοδοί του, οι οποίες υλοποιούν ίσως την πιο κύρια δυνατότητα της συγκεκριμένης γλώσσας, δηλαδή το δυναμικό άνοιγμα και κλείσιμο παραθύρων ενός περιηγητή που απεικονίζει το περιεχόμενο διαφόρων Ιστοτόπων, είναι οι μέθοδοι open() και close().

Ωστόσο, ένα νέο παράθυρο μίας Ιστοσελίδας είναι εφικτό να ανοίγει και με απλό HTML κώδικα, όπως είναι, για παράδειγμα, η δήλωση: ``. Το μειονέκτημα, όμως, σε αυτή την περίπτωση είναι ότι το νέο παράθυρο μίας Ιστοσελίδας, που ανοίγει με το προηγούμενο απόσπασμα HTML κώδικα, είναι του ίδιου μεγέθους με το «γονικό» του παράθυρο και τοποθετείται πάντα μπροστά από αυτό. Κάτι τέτοιο πολλές φορές είναι ανεπιθύμητο. Η γλώσσα JavaScript παρέχει πολύ μεγαλύτερη ευελιξία στη δημιουργία και στο χειρισμό παραθύρων Ιστοχώρων.

Ας εξεταστεί καταρχήν η μέθοδος open() του αντικειμένου Window της γλώσσας JavaScript. Με τη βοήθεια της μεθόδου αυτής υπάρχει η δυνατότητα να ανοιχθεί ένα νέο παράθυρο ενός φυλλομετρητή μέσα στο οποίο φορτώνεται η Ιστοσελίδα που καθορίζεται από την εκάστοτε URL, που αποτελεί κάθε φορά το πρώτο όρισμα της εν λόγω μεθόδου και έχει τα χαρακτηριστικά που ορίζονται από το εκάστοτε τρίτο όρισμα της συγκεκριμένης μεθόδου. Η σύνταξη, λοιπόν, της μεθόδου open() έχει ως εξής:

```
window.open(URL,name,specs,replace);
```

Όπου:

- URL: Προαιρετική. Καθορίζει την ηλεκτρονική διεύθυνση μίας Ιστοσελίδας που επιθυμείται να ανοίγει μέσα στο εκάστοτε νέο παράθυρο ενός προγράμματος περιήγησης. Αν το όρισμα αυτό παραλείπεται ή είναι κενό, ο κατάλληλος σε κάθε περίπτωση φυλλομετρητής φορτώνει στη θέση του εκάστοτε καινούργιου παραθύρου ένα κενό έγγραφο, χωρίς να φορτώνει κάποιο έγγραφο Ιστοσελίδας.

- **name:** Καθορίζει το όνομα του εκάστοτε νέου παραθύρου ενός πλοηγού μέσα στο οποίο φορτώνεται η Ιστοσελίδα που καθορίζεται από τη URL. Η παράμετρος αυτή μπορεί να πάρει τις ακόλουθες τιμές:
 - **blank:** Είναι η προεπιλεγμένη τιμή. Η URL φορτώνεται σε ένα νέο κενό παράθυρο ενός περιηγητή.
 - **parent:** Η URL φορτώνεται στο εκάστοτε τρέχον «γονικό» παράθυρο ενός περιηγητή.
 - **self:** Η URL φορτώνεται αντικαθιστώντας το εκάστοτε τρέχον παράθυρο ενός περιηγητή.
 - **top:** Η URL φορτώνεται αντικαθιστώντας οποιοδήποτε παράθυρο ενός περιηγητή.
 - **name:** Το όνομα του εκάστοτε νέου παραθύρου ενός περιηγητή μέσα στο οποίο φορτώνεται η URL. Σε αυτό το σημείο να προσεχθεί ιδιαίτερα ότι το όνομα δεν πρέπει ταυτίζεται με τον τίτλο ενός παραθύρου ενός περιηγητή που απεικονίζει το περιεχόμενο μίας Ιστοσελίδας.
- **specs:** Προαιρετική. Τα χαρακτηριστικά του εκάστοτε παραθύρου ενός πλοηγού μέσα στο οποίο φορτώνεται η Ιστοσελίδα που καθορίζεται από τη URL Η παράμετρος αυτή μπορεί να πάρει τις ακόλουθες τιμές:
 - **channelmode=yes|no|1|0:** Καθορίζει αν το εκάστοτε νέο παράθυρο ενός φυλλομετρητή στο οποίο φορτώνεται η URL θα εμφανίζεται ή όχι σε θεατρική κατάσταση λειτουργίας. Η προεπιλεγμένη τιμή είναι όχι. Αφορά μόνο το πρόγραμμα περιήγησης IE.
 - **fullscreen=yes|no|1|0:** Καθορίζει αν το εκάστοτε τρέχον παράθυρο θα εμφανίζεται ή όχι σε πλήρη οθόνη. Να παρατηρηθεί, εδώ, ότι ένα παράθυρο σε πλήρη οθόνη πρέπει, επίσης, να είναι υποχρεωτικά σε θεατρική κατάσταση λειτουργίας. Η προεπιλεγμένη τιμή είναι όχι. Αφορά μόνο το πρόγραμμα περιήγησης IE.
 - **height=pixels:** Καθορίζει το ύψος του εκάστοτε τρέχοντος παραθύρου σε pixels. Η ελάχιστη τιμή είναι 100.
 - **left=pixels:** Καθορίζει την αριστερή θέση του εκάστοτε παρόντος παραθύρου σε pixels. Αρνητικές τιμές δεν επιτρέπονται.
 - **location=yes|no|1|0:** Καθορίζει την εμφάνιση ή όχι πλαισίου για την εισαγωγή της διεύθυνσης Ιστοσελίδων. Αφορά μόνο το πρόγραμμα περιήγησης Opera.
 - **menubar=yes|no|1|0:** Καθορίζει την εμφάνιση ή όχι γραμμής μενού στο εκάστοτε παρόν παράθυρο.
 - **resizable=yes|no|1|0:** Καθορίζει αν ο εκάστοτε χρήστης είναι σε θέση ή όχι να αλλάζει τις διαστάσεις του παρόντος παραθύρου. Αφορά μόνο το πρόγραμμα περιήγησης IE.
 - **scrollbars=yes|no|1|0:** Καθορίζει την εμφάνιση ή όχι ράβδων κύλισης στο εκάστοτε τρέχον παράθυρο. Αφορά μόνο τα προγράμματα περιήγησης IE, Firefox και Opera.
 - **status=yes|no|1|0:** Καθορίζει την εμφάνιση ή όχι γραμμής κατάστασης στο εκάστοτε παρόν παράθυρο.
 - **titlebar=yes|no|1|0:** Καθορίζει την εμφάνιση ή όχι γραμμής τίτλου στο εκάστοτε τρέχον παράθυρο. Παραλείπεται, εκτός αν η εφαρμογή κλήσης είναι μία HTML εφαρμογή ή ένα έμπιστο πλαίσιο διαλόγου.
 - **toolbar=yes|no|1|0:** Καθορίζει την εμφάνιση ή όχι γραμμής εργαλείων στο εκάστοτε παρόν παράθυρο.
 - **top=pixels:** Καθορίζει τη θέση κορυφής του εκάστοτε τρέχοντος παραθύρου. Αρνητικές τιμές δεν επιτρέπονται.
 - **width=pixels:** Καθορίζει το πλάτους του εκάστοτε παρόντος παραθύρου σε pixels. Η ελάχιστη τιμή είναι 100.
- **replace:** Προαιρετική. Καθορίζει αν η URL της εκάστοτε τρέχουσας Σελίδας φορτώνεται στο εκάστοτε τρέχον παράθυρο δημιουργώντας νέα ή αντικαθιστώντας την υπάρχουσα λίστα ιστορικού. Η παράμετρος αυτή μπορεί να πάρει τις ακόλουθες τιμές:
 - **true:** Η URL αντικαθιστά την υπάρχουσα λίστα ιστορικού.
 - **false:** Το URL δημιουργεί νέα λίστα ιστορικού.

Η τιμή επιστροφής, τώρα, της μεθόδου `open()` είναι μία αναφορά στο νέο παράθυρο μίας Ιστοσελίδας που ανοίγει σε κάθε περίπτωση ή `null`, αν η κλήση αποτύχει.

Στον αντίποδα της μεθόδου open(), η γλώσσα JavaScript χρησιμοποιεί τη μέθοδο close() του αντικείμενο Window, για να κλείνει δυναμικά το παράθυρο που άνοιξε με τη μέθοδο open(). Η σύνταξη, τώρα, της μεθόδου close() έχει ως εξής:

window.close();

Όπως είναι φανερό, η προαναφερόμενη μέθοδος δε δέχεται κανένα όρισμα ούτε επιστρέφει κάποια τιμή.

Επίσης, το αντικείμενο Window της γλώσσας JavaScript, πέρα από τις μεθόδους open() και close() και τις alert(), confirm(), prompt() (που παρουσιάστηκαν διεξοδικά παραπάνω), υποστηρίζει μία σειρά από επιπρόσθετες ενδιαφέρουσες μεθόδους, οι πιο κύριες από τις οποίες αναφέρονται στον παρακάτω Πίνακα:

Μέθοδος	Λειτουργία	Σύνταξη	Παράμετρος (-οι)	Τιμή Επιστροφής
blur()	Ακυρώνει την εστίαση σε ένα παράθυρο μίας Ιστοσελίδας.	window.blur()	Καμία.	Καμία.
clearInterval()	Μηδενίζει το χρονόμετρο που ορίστηκε με τη μέθοδο setInterval().	clearInterval(idOfSetInterval)	idOfSetInterval: Υποχρεωτική Η id τιμή ενός χρονόμετρο που επιστράφηκε από τη μέθοδο setInterval().	Καμία.
clearTimeout()	Μηδενίζει το χρονόμετρο που ορίστηκε με τη μέθοδο setTimeout().	clearTimeout(idOfSetTimeout)	idOfSetTimeout: Υποχρεωτική Η id τιμή του ενός χρονόμετρο που επιστράφηκε από τη μέθοδο setTimeout().	Καμία.
focus()	Καθαρίζει την εστίαση στο εκάστοτε τρέχον παράθυρο μίας Ιστοσελίδας.	window.focus()	Καμία.	Καμία.
moveBy()	Μετακινεί ένα παράθυρο μίας Ιστοσελίδας κατά ένα ορισμένο αριθμό από pixels, σε σχέση με την εκάστοτε τρέχουσα θέση του.	window.moveBy(x,y)	i) x: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που ισούται με το σύνολο των pixels που πρόκειται ένα παράθυρο μίας Ιστοσελίδας να μετακινηθεί οριζόντια σε σχέση την εκάστοτε τρέχουσα θέση του. ii) y: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που ισούται με το σύνολο των pixels	Καμία.

			που πρόκειται ένα παράθυρο μίας Ιστοσελίδας να μετακινηθεί κατακόρυφα σε σχέση την εκάστοτε τρέχουσα θέση του.	
moveTo()	Μετακινεί την πάνω αριστερή γωνία ενός παραθύρου μίας Ιστοσελίδας σε ένα συγκεκριμένο σημείο.	window.moveTo(x,y)	i) x: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που καθορίζει σε pixels την τετμημένη του σημείου που πρόκειται να μετακινηθεί η πάνω αριστερή γωνία ενός παραθύρου μίας Ιστοσελίδας. ii) y: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που καθορίζει σε pixels την τεταγμένη του σημείου που πρόκειται να μετακινηθεί η πάνω αριστερή γωνία ενός παραθύρου μίας Ιστοσελίδας.	Καμία.
print()	Τυπώνει το περιεχόμενο ενός παραθύρου μίας Ιστοσελίδας.	window.print()	Καμία.	Καμία.
resizeBy()	Αλλάζει τις διαστάσεις ενός παραθύρου μίας Ιστοσελίδας κατά ένα ορισμένο αριθμό από pixels σε σχέση με το εκάστοτε τρέχον μέγεθός του.	resizeBy(width,height)	i) width: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που καθορίζει κατά πόσα pixels πρόκειται να αλλάξει το πλάτος ενός παραθύρου μίας Ιστοσελίδας σε σχέση με το εκάστοτε τρέχον πλάτος του. ii) height: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που καθορίζει κατά πόσα pixels πρόκειται να αλλάξει το ύψος ενός παραθύρου μίας Ιστοσελίδας σε σχέση με το εκάστοτε τρέχον μήκος του.	Καμία.
resizeTo()	Αλλάζει τις διαστάσεις ενός παραθύρου μίας Ιστοσελίδας, δίνοντάς τους	window.resizeTo(width,height)	i) width: Υποχρεωτική. Ένας θετικός αριθμός που καθορίζει το πλάτος σε pixels που πρόκειται να έχει ένα παράθυρο	Καμία.

	συγκεκριμένα μεγέθη.		μίας Ιστοσελίδας. ii) height: Υποχρεωτική. Ένας θετικός αριθμός που καθορίζει το ύψος σε pixels που πρόκειται να έχει ένα παράθυρο μίας Ιστοσελίδας	
scrollBy()	Μετακινεί την ορατή περιοχή ενός παραθύρου μίας Ιστοσελίδας κατά ένα ορισμένο αριθμό από pixels σε σχέση με την εκάστοτε τρέχουσα θέση του.	window.scrollBy(x,y)	i) x: Υποχρεωτική. Ένας αριθμός που καθορίζει κατά πόσα pixels πρόκειται να μετακινηθεί οριζόντια η ορατή περιοχή ενός παραθύρου μίας Ιστοσελίδας. Αν ο προαναφερόμενος αριθμός είναι θετικός, τότε η εν λόγω μετακίνηση γίνεται προς τα δεξιά, ενώ, αν είναι αρνητικός προς τα αριστερά. ii) y: Υποχρεωτική. Ένας αριθμός που καθορίζει κατά πόσα pixels πρόκειται να μετακινηθεί κατακόρυφα η ορατή περιοχή ενός παραθύρου μίας Ιστοσελίδας. Αν ο προαναφερόμενος αριθμός είναι θετικός, τότε η εν λόγω μετακίνηση γίνεται προς τα πάνω, ενώ, αν είναι αρνητικός προς τα κάτω.	Καμία.
scrollTo()	Μετακινεί την ορατή περιοχή ενός παραθύρου μίας Ιστοσελίδας σε ένα συγκεκριμένο σημείο.	window.scrollTo(x,y)	ii) x: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που καθορίζει σε pixels την τετμημένη του σημείου που πρόκειται να μετακινηθεί η ορατή περιοχή ενός παραθύρου μίας Ιστοσελίδας. ii) y: Υποχρεωτική. Ένας θετικός ή αρνητικός αριθμός που καθορίζει σε pixels την τεταγμένη του σημείου που πρόκειται να μετακινηθεί η ορατή περιοχή ενός παραθύρου μίας Ιστοσελίδας.	Καμία.

setInterval()	Εκτελεί μία συνάρτηση (ή υπολογίζει μία έκφραση) ανά τακτά χρονικά διαστήματα (σε χιλιοστά του δευτερολέπτου).	setInterval (function, milliseconds,param1, param2,param3,..., paramN)	<p>i) function: Υποχρεωτική. Μία συνάρτηση (ή μία έκφραση) που ανά τακτά χρονικά διαστήματα (σε χιλιοστά του δευτερολέπτου) πρόκειται να εκτελείται (ή να υπολογίζεται).</p> <p>ii) milliseconds: Υποχρεωτική. Τα ενδιάμεσα χρονικά διαστήματα (σε χιλιοστά του δευτερολέπτου) ανά τα οποία πρόκειται να εκτελείται (ή να υπολογίζεται) η παράμετρος function.</p> <p>iii) param1,param2,param3,...,paramN: Προαιρετικές. Επιπρόσθετες παράμετροι που μπορεί να δέχεται η παράμετρος function (στην περίπτωση που είναι συνάρτηση και όχι έκφραση).</p>	Καμία.
setTimeout()	Εκτελεί μία συνάρτηση (ή υπολογίζει μία έκφραση) μία φορά, ύστερα από ένα συγκεκριμένο χρονικό διάστημα(σε χιλιοστά του δευτερολέπτου).	setTimeout (function, milliseconds,param1, param2,param3,..., paramN)	<p>i) function: Υποχρεωτική. Μία συνάρτηση (ή μία έκφραση) που κάθε φορά πρόκειται να εκτελεστεί (ή να υπολογιστεί) ύστερα από ένα συγκεκριμένο χρονικό διάστημα(σε χιλιοστά του δευτερολέπτου)..</p> <p>ii) milliseconds: Υποχρεωτική. Τα χρονικό διάστημα (σε χιλιοστά του δευτερολέπτου) ύστερα από το οποίο πρόκειται να εκτελεστεί ή να υπολογιστεί η παράμετρος function.</p> <p>iii) param1,param2,param3,...,paramN: Προαιρετικές. Επιπρόσθετες παράμετροι που μπορεί να δέχεται η παράμετρος function (στην περίπτωση που είναι συνάρτηση και όχι έκφραση).</p>	Καμία.

stop()		Διακόπτει τη φόρτωση του παραθύρου μίας Ιστοσελίδας.	Καμία.	Καμία.
--------	--	--	--------	--------

Πίνακας 15: Οι πιο βασικές μέθοδοι του αντικειμένου Window της JavaScript

[36] [41] [42] [43]

2.12.10.3 Τα Πλαίσια

Παρόλο την ιδιαίζουσα σημασία που έχει το αντικείμενο Window ενός HTML εγγράφου στο Διαδίκτυο, εντούτοις οι περισσότερες εφαρμογές στο Διαδίκτυο παρουσιάζονται μέσα από πλαίσια (frames) και όχι μέσα από παράθυρα. Τα πλαίσια έχουν ιδιότητες παρεμφερείς με αυτές των παραθύρων. Έτσι, ένα νέο πλαίσιο μίας Ιστοσελίδας, που ανοίγει μέσα από ένα προγενέστερο, μπορεί να αναφερθεί σε αυτό με την ιδιότητα opener. Επιπλέον, η ιδιότητα frames αναφέρεται στον πίνακα των πλαισίων που υπάρχουν σε ένα παράθυρο μίας Ιστοσελίδας. Αν ένα παράθυρο μίας Ιστοσελίδας δεν περιέχει πλαίσια, ο πίνακας frames[] αυτής της Σελίδας είναι κενός. Αν μία Ιστοσελίδα έχει πλαίσια, για να προσπελάσουμε το πρώτο από αυτά μπορούμε να δώσουμε την εντολή:

```
top.frames[0];
```

Η ιδιότητα top αναφέρεται στο κύριο παράθυρο μέσα στο οποίο βρίσκεται το πλαίσιο μίας Ιστοσελίδας. Από το πρώτο πλαίσιο μπορούμε να προσπελάσουμε το δεύτερο με την εντολή:

```
top.frames[1];
```

και ούτω καθεξής.

Ένας άλλος τρόπος για να προσπελαστεί ένα πλαίσιο μίας Ιστοσελίδας είναι να χρησιμοποιηθεί το όνομα το οποίο του έχει αποδοθεί μέσω της ιδιότητας name σε μία ετικέτα <frame> κάπου μέσα στον κώδικα ενός HTML εγγράφου. Έτσι, για να δημιουργηθούν, για παράδειγμα, δύο πλαίσια στο κύριο παράθυρο μίας Ιστοσελίδας με ονόματα leftFrame και rightFrame, τα οποία θα καλύπτουν χώρο οθόνης 30% και 70%, αντίστοιχα, ακολουθείται η σύνταξη:

```
<frameset cols="30%,70%">
<frame src="frame1.html" name="leftFrame">
<frame src="frame2.html" name="rightFrame">
</frameset>
```

Στην προκειμένη περίπτωση, για την προσπέλαση του αριστερού πλαισίου δύναται να γίνει χρήση του όρου top.leftFrame, ενώ για την προσπέλαση του δεξιού πλαισίου του όρου top.rightFrame.

Στο σημείο αυτό, αξίζει να γίνει μία σύντομη αναφορά σε μία τεχνική αλλαγής του περιεχομένου πολλών πλαισίων. Η τεχνική αυτή χρησιμοποιεί το ψευδοπρωτόκολλο JavaScript. Σύμφωνα με αυτό είναι δυνατή η εκτέλεση κώδικα μέσα από ένα σύνδεσμο URL. Αυτό γίνεται με τη χρήση ενός ειδικού προσδιοριστή, του προσδιοριστή javascript: , που ονομάζεται προσδιοριστής ψευδοπρωτοκόλλου, επειδή δεν πρόκειται για κάποιο πραγματικό πρωτόκολλο, όπως είναι, για παράδειγμα το http ή το ftp, αλλά λειτουργεί με τον ίδιο τρόπο. Με τη βοήθεια του ειδικού αυτού πρωτοκόλλου, όταν πατιέται ένας υπερσύνδεσμος, αντί να φορτώνεται μία νέα Ιστοσελίδα στο παράθυρο του εκάστοτε προγράμματος περιήγησης, εκτελείται κατάλληλος JavaScript κώδικας. Ο κώδικας αυτός μπορεί να είναι, για παράδειγμα, μία ή περισσότερες εντολές ή μία συνάρτηση. Αν οι εντολές είναι περισσότερες από μία, τότε χωρίζονται μεταξύ τους με το σύμβολο του ελληνικού ερωτηματικού (;). Σε κάποιες περιπτώσεις ο προσδιοριστής javascript: συνοδεύεται από τον τελεστή void. Ο τελεστής void χρησιμοποιείται στις περιπτώσεις που είναι ωφέλιμο να πραγματώνεται ένας υπολογισμός στα πλαίσια μίας διαδικτυακής εφαρμογής, χωρίς να επιστρέφεται κάποια τιμή.

[41]

2.12.10.4 Το Αντικείμενο Navigator

Στα πρώτα βήματα του Παγκόσμιου Ιστού τα πράγματα ήταν απλά. Όλοι οι Ιστοχώροι ήταν στατικοί, κατασκευασμένοι με γλώσσα HTML και ως πρόγραμμα περιήγησης ήταν διαθέσιμο το περιήγηση Mosaic. Το πρόγραμμα αυτό εκτοπίστηκε στη συνέχεια από το Netscape Navigator (έκδοση 1.0, το Δεκέμβριο του 1994), που κυριάρχησε για αρκετά χρόνια στις προτιμήσεις των χρηστών ως βασικό πρόγραμμα περιήγησης στο Διαδίκτυο. Η εταιρεία Microsoft, αν και αρχικά δεν είχε δώσει μεγάλη σημασία στις εφαρμογές του Διαδικτύου, τον Αύγουστο του 1995 κυκλοφόρησε το δικό της πρόγραμμα περιήγησης, το οποίο και ονόμασε Internet Explorer. Με την πάροδο των ετών το Διαδίκτυο διαδραματίζει όλο και σημαντικότερο ρολό στην καθημερινότητα των ανθρώπων, με αποτέλεσμα ολοένα και περισσότερες εταιρίες να στρέφουν το ενδιαφέρον τους προς την υλοποίηση προγραμμάτων περιήγησης στο Διαδίκτυο. Έτσι, σήμερα, εκτός από τη Microsoft με τον Internet Explorer, υπάρχει και άλλες κολοσσιαίες εταιρίες, που τα προγράμματα πλοήγησής τους στο Διαδίκτυο, όχι μονό συναγωνίζονται, αλλά σε κάποιες περιπτώσεις υπερτερούν του Internet Explorer. Παραδείγματα τέτοιων εταιριών και των αντίστοιχων φυλλομετρητών τους είναι η Mozilla με το Firefox, η Google με το Chrome, η Opera με τον ομώνυμο περιηγητή της, η Apple με το Safari.

Επειδή, ωστόσο, εντοπίζονται λειτουργικές διαφορές όχι μόνο μεταξύ των διαφορετικών προγραμμάτων πλοήγησης, αλλά και μεταξύ των εκδόσεων του ίδιου περιηγητή, σε πολλές περιπτώσεις χρειάζεται να γίνεται ανίχνευση του εκάστοτε πλοηγού καθώς και της έκδοσής του. Για το σκοπό αυτόν υπάρχει το αντικείμενο Navigator, το οποίο δουλεύει με τον ίδιο τρόπο σε όλες τις εκδόσεις όλων των διαθέσιμων προγραμμάτων φυλλομέτρησης. Οι πιο σημαντικές ιδιότητες του αντικειμένου αυτού παρουσιάζονται στον επόμενο Πίνακα:

Ιδιότητα	Περιγραφή
appName	Επιστρέφει το επίσημο όνομα του εκάστοτε προγράμματος περιήγησης.
appVersion	Επιστρέφει τον αριθμό έκδοσης του εκάστοτε προγράμματος περιήγησης και κάποιες πληροφορίες σχετικά με αυτό.
cookieEnabled	Επιστρέφει μία λογική τιμή καθορίζοντας έτσι, αν τα cookies είναι ενεργοποιημένα ή όχι στο εκάστοτε πρόγραμμα περιήγησης. Αν είναι, η τιμή αυτή είναι true, αν δεν είναι, false.
language	Επιστρέφει τη γλώσσα που χρησιμοποιεί το εκάστοτε πρόγραμμα περιήγησης.
onLine	Επιστρέφει μία λογική τιμή καθορίζοντας έτσι, αν το εκάστοτε πρόγραμμα περιήγησης είναι σε σύνδεση ή όχι. Αν είναι, η τιμή αυτή είναι true, αν όχι, false.
platform	Επιστρέφει την πλατφόρμα υλικού στην οποία εκτελείται η εφαρμογή του εκάστοτε προγράμματος περιήγησης.

Πίνακας 16: Οι πιο βασικές ιδιότητες του αντικειμένου Navigator της JavaScript

Επίσης, η πιο σημαντική μέθοδος του αντικειμένου Navigator είναι η `javaEnabled()`, η οποία επιστρέφει μία λογική τιμή καθορίζοντας έτσι, αν η γλώσσα Java είναι ενεργοποιημένη ή όχι στο εκάστοτε πρόγραμμα περιήγησης. Αν είναι, η τιμή αυτή είναι true, αν όχι, false. Η σύνταξη της συγκεκριμένης μεθόδου έχει ως εξής:

```
navigator.javaEnabled();
```

Όπως είναι φανερό η εν λόγω μέθοδος δε δέχεται κανένα όρισμα, ενώ η τιμή επιστροφής της είναι ένα αντικείμενο τύπου Boolean, το οποίο καθορίζει αν η γλώσσα Java είναι ενεργοποιημένη ή όχι στο εκάστοτε πρόγραμμα φυλλομέτρησης. Αν είναι, η τιμή αυτή είναι true, αν όχι, false.

[41] [46]

2.12.10.5 Το Αντικείμενο Screen

Το αντικείμενο Screen παρέχει πληροφορίες για τις ιδιότητες της οθόνης ενός χρήστη-επισκέπτη ενός Ιστοτόπου στον Παγκόσμιο Ιστό. Οι σημαντικότερες ιδιότητες του αντικείμενου screen φαίνονται στον παρακάτω Πίνακα:

Ιδιότητα	Περιγραφή
colorDepth	Επιστρέφει σε bits το βάθος των χρωμάτων των εικόνων που εμφανίζονται στην οθόνη ενός επισκέπτη μίας Ιστοσελίδας.
height	Επιστρέφει σε pixels το ύψος της οθόνης ενός επισκέπτη μίας Ιστοσελίδας.
pixelDepth	Επιστρέφει σε bits ανά pixel την ανάλυση του χρώματος της οθόνης ενός επισκέπτη μίας Ιστοσελίδας.
width	Επιστρέφει σε pixels το πλάτος της οθόνης ενός επισκέπτη μίας Ιστοσελίδας.

Πίνακας 17: Οι πιο βασικές ιδιότητες του αντικειμένου Screen της JavaScript

[41]

2.12.11 Τα Έγγραφα, οι Εικόνες και η JavaScript

2.12.11.1 Το Αντικείμενο Document

Το αντικείμενο Document έπεται του Window στην ιεραρχία του DOM ενός HTML Εγγράφου. Ένα αντικείμενο Document δημιουργείται, όταν ένα έγγραφο ενός Ιστοτόπου φορτώνεται στο παράθυρο ενός προγράμματος περιήγησης στο Διαδίκτυο. Το αντικείμενο Document αναφέρεται σε ένα HTML έγγραφο και σε ό,τι αυτό περιέχει. Όλα τα στοιχεία που υπάρχουν στο εν λόγω έγγραφο, όπως είναι, για παράδειγμα, οι επικεφαλίδες, οι παράγραφοι, οι εικόνες, οι συνδέσεις, οι φόρμες αντιστοιχούν σε ξεχωριστά αντικείμενα τα οποία βρίσκονται και αυτά στην ιεραρχία του DOM. Το αντικείμενο Document προσφέρει μία σειρά από ιδιότητες και μεθόδους, οι οποίες παρέχουν πληροφορίες για το εκάστοτε HTML έγγραφο ή επιτρέπουν τον καθορισμό της εμφάνισής του.

Οι κυριότερες ιδιότητες του αντικειμένου Document της γλώσσας JavaScript παρουσιάζονται στον Πίνακα που έπεται:

Ιδιότητα	Περιγραφή
anchors	Επιστρέφει έναν πίνακα αντικειμένων που περιέχει τους συνδέσμους οι οποίοι υπάρχουν σε μία Ιστοσελίδα.
applets	Επιστρέφει έναν πίνακα αντικειμένων που αντιστοιχούν στις μικροεφαρμογές οι οποίες υπάρχουν σε μία Ιστοσελίδα.
body	Ρυθμίζει ή επιστρέφει το κύριο σώμα μίας Ιστοσελίδας.
cookie	Επιστρέφει όλα τα cookies μίας Ιστοσελίδας.
domain	Επιστρέφει το DNS όνομα του διακομιστή Ιστού μίας Ιστοσελίδας.
embeds	Επιστρέφει έναν πίνακα του οποίου κάθε στοιχείο αντιστοιχεί σε ένα ζευγάρι ετικετών ενσωμάτωσης που χρησιμοποιούνται στον HTML κώδικα μίας Ιστοσελίδας.
forms	Είναι ένας πίνακας αντικειμένων καθένα από τα οποία αντιστοιχεί σε μία φόρμα που υπάρχει σε μία Ιστοσελίδα.
head	Επιστρέφει την κεφαλή μίας Ιστοσελίδας.
images	Είναι ένας πίνακας αντικειμένων καθένα από τα οποία αντιστοιχεί σε μία εικόνα που υπάρχει σε μία Ιστοσελίδα.
lastmodified	Επιστρέφει ένα αλφαριθμητικό με την ημερομηνία και την ώρα της τελευταίας τροποποίησης μίας Ιστοσελίδας.
links	Επιστρέφει έναν πίνακα αντικειμένων καθένα από τα οποία αντιστοιχεί σε έναν υπερσύνδεσμο που υπάρχει σε μία Ιστοσελίδα.
referrer	Επιστρέφει τη URL εκείνης της Ιστοσελίδας η οποία φόρτωσε την εκάστοτε τρέχουσα Ιστοσελίδα.
title	Επιστρέφει ή ρυθμίζει τον τίτλο μίας Ιστοσελίδας.

URL	Επιστρέφει τη URL μίας Ιστοσελίδας.
-----	-------------------------------------

Πίνακας 18: Οι πιο βασικές ιδιότητες του αντικειμένου Document της JavaScript

Οι κυριότερες, τώρα, μέθοδοι του αντικειμένου Document της γλώσσας JavaScript παρατίθενται στον Πίνακα που έπεται:

Μέθοδος	Λειτουργία	Σύνταξη	Παράμετρος (-οι)	Τιμή Επιστροφής
open()	Ανοίγει ένα νέο HTML έγγραφο.	document.open(MIMEType,replace)	i) MIMEType: Προαιρετική. Ο τύπος κειμένου του εκάστοτε νέου HTML εγγράφου που πρόκειται να ανοιχθεί. Η προκαθορισμένη τιμή του προαναφερόμενου τύπου είναι: "text/html". ii) replace: Προαιρετική. Αν τοποθετηθεί, το εκάστοτε νέο HTML έγγραφο που πρόκειται να ανοιχθεί κληρονομεί το ιστορικό εκείνου του εγγράφου από το οποίο προήλθε.	Καμία.
close()	Κλείνει ένα HTML έγγραφο που έχει ανοίξει με τη μέθοδο open().	document.close()	Καμία.	Καμία.
write()	Προσθέτει κείμενο σε ένα HTML έγγραφο.	document.write(expr1, expr2,expr3,...,exprN)	expr1,expr2, expr3, ...,exprN: Προαιρετικές. Οι εκφράσεις 1,2,3, ..., N, αντίστοιχα, που πρόκειται να γραφούν στο κείμενο ενός HTML έγγραφο με τη σειρά που ορίζονται.	Καμία.
writeln()	Προσθέτει κείμενο σε ένα HTML έγγραφο, τοποθετώντας ένα χαρακτήρα αλλαγής γραμμής (\n) στο τέλος κάθε γραμμής του προαναφερόμενου κειμένου.	document.writeln (expr1, expr2,expr3,...,exprN)	expr1,expr2, expr3, ...,exprN: Προαιρετικές. Οι εκφράσεις 1,2,3, ..., N, αντίστοιχα, που πρόκειται να γραφούν στο κείμενο ενός HTML	Καμία.

			έγγραφο με τη σειρά που ορίζονται.	
--	--	--	------------------------------------	--

Πίνακας 19: Οι πιο βασικές μέθοδοι του αντικειμένου Document της JavaScript

Με τις μεθόδους `open()` και `close()` δίνεται η ευκαιρία στον εκάστοτε προγραμματιστή να ανοίξει και να κλείσει ένα καινούργιο HTML έγγραφο, ενώ με τις μεθόδους `write()` και `writeln()` δύναται ο εκάστοτε προγραμματιστής να προσθέσει κείμενο σε ένα HTML έγγραφο. Πρέπει, εδώ, να τονιστεί το γεγονός ότι για να γίνει εγγραφή σε ένα HTML έγγραφο είναι αναγκαίο να καλούνται οι μέθοδοι `write()` και `writeln()` κατά το χρόνο φόρτωσης αυτού του εγγράφου και όχι μέσα από το κυρίως σώμα του. Αυτό σημαίνει πως, αν είναι χρήσιμο να εμφανιστεί κείμενο σε ένα HTML έγγραφο, οι μέθοδοι `document.write()` και `document.writeln()` πρέπει να τοποθετούνται μέσα στις επικεφαλίδες `<script>` και `</script>` ενός JavaScript σεναρίου του εν λόγω εγγράφου. Αν δε συμβεί αυτό και τοποθετηθούν μέσα στο κυρίως σώμα του προαναφερόμενου εγγράφου (πχ σε ένα χειριστή συμβάντος), το αποτέλεσμα θα είναι να ανοίξει ένα νέο έγγραφο μέσα στο οποίο θα γραφεί μήνυμα το οποίο θα επικαλύπτει, όμως, το προηγούμενο έγγραφο. Γίνεται, όμως, κάλλιστα να εκμεταλλεύονται οι μέθοδοι `document.write()` και `document.writeln()` είτε στο κυρίως σώμα ενός εγγράφου είτε σε ένα χειριστή συμβάντος, για να αναδύεται κείμενο σε ένα άλλο έγγραφο, που βρίσκεται εκείνη τη στιγμή ανοικτό σε κάποιο άλλο παράθυρο, ή σε ένα έγγραφο που βρίσκεται τοποθετημένο σε ένα άλλο πλαίσιο.

Κατά τη διαδικασία της δημιουργίας ενός νέου HTML εγγράφου κανονικά καλείται η μέθοδος `open()`, για να ανοίξει το έγγραφο αυτό. Η χρήση της μεθόδου αυτής, όμως, είναι κατά κάποιον τρόπο προαιρετική, γιατί, αν χρησιμοποιηθούν απευθείας οι μέθοδοι `document.write()` και `document.writeln()`, ώστε να πραγματοποιηθεί εγγραφή στο προαναφερόμενο έγγραφο, η γλώσσα JavaScript θα το ανοίξει αυτόματα από μόνη της, σαν να είχε γίνει κλήση της μεθόδου `open()`. Αντίθετα, πρέπει να διατίθεται υποχρεωτικά η μέθοδος `close()`, προκειμένου να ολοκληρωθεί η φόρτωση ενός HTML εγγράφου από το εκάστοτε πρόγραμμα πλοήγησης.

Τελειώνοντας την αναφορά στις μεθόδους του αντικειμένου Document, κρίνεται επιθυμητή η επισήμανση της διαφοράς των μεθόδων `write()` και `writeln()`. Κατά την περιγραφή τους τονίστηκε ότι διαφέρουν μόνο στο ότι η `writeln()` προσθέτει απλώς ένα χαρακτήρα αλλαγής γραμμής στο κείμενο ενός HTML εγγράφου. Στην HTML, όμως, οι αλλαγές γραμμών αγνοούνται και έτσι δε θα υπάρχει κάποια ορατή διαφορά σε ένα HTML έγγραφο. Η διαφορετική τους δράση ισχύει σε μη HTML έγγραφα, όπου λαμβάνονται υπόψη οι χαρακτήρες αλλαγής γραμμής.

[41] [42] [43]

2.12.11.2 Το Αντικείμενο Image

Όπως προαναφέρθηκε, μία από τις βασικές ιδιότητες του αντικειμένου Document, είναι και η `images`, η οποία είναι ένας πίνακας αντικειμένων, κάθε ένα από τα οποία αντιστοιχεί σε μία εικόνα που υπάρχει στο εκάστοτε HTML έγγραφο. Τώρα, το αντικείμενο `Image` διαθέτει την πολύ σημαντική ιδιότητα `src`, η οποία είναι εκμεταλλεύσιμη για την ανάγνωση και την τοποθέτηση εικόνων εντός ενός HTML εγγράφου. Αν χρησιμοποιηθεί για ανάγνωση, διαβάζεται η διεύθυνση από την οποία φορτώθηκε μία εικόνα στο προαναφερόμενο έγγραφο, ενώ αν χρησιμοποιηθεί για τοποθέτηση, φορτώνεται σε αυτό το έγγραφο μία νέα εικόνα, την οποία το εκάστοτε πρόγραμμα περιήγησης μπορεί να την τοποθετήσει στην ίδια θέση όπου βρισκόταν μία άλλη. Εννοείται πως και οι δύο εικόνες πρέπει να έχουν τις ίδιες διαστάσεις.

Το αντικείμενο `Image` και η ιδιότητα `src` βοηθούν ιδιαίτερα στη δημιουργία εφέ εναλλαγής εικόνων ή, όπως είναι πιο γνωστά, εφέ `rollover`. Ένα εφέ εναλλαγής εικόνων είναι κάτι που βλέπει πολύ συχνά κάποιος που περιηγείται στις Ιστοσελίδες του Διαδικτύου. Σύμφωνα, λοιπόν, με το συγκεκριμένο εφέ, όταν ένας επισκέπτης μίας Σελίδας τοποθετεί το δείκτη του ποντικιού του πάνω σε μία εικόνα, αυτή η εικόνα αλλάζει και στη θέση της εμφανίζεται μία άλλη. Το εν λόγω εφέ αρέσει πολύ στους χρήστες χρησιμοποιείται συνήθως σε εικόνες που περιέχουν συνδέσμους προς άλλα έγγραφα, αλλά αυτό δεν είναι απαραίτητα η μόνη χρήση της προαναφερόμενης τεχνικής. Είναι γεγονός ότι ένα εφέ εναλλαγής εικόνων δίνει μία δυναμική αίσθηση στον εκάστοτε χρήστη που επισκέπτεται μία Ιστοσε-

λίδα. Ένα ακόμα πιο εντυπωσιακό εφέ, όμως, είναι αυτό της πολλαπλής εναλλαγής εικόνων. Στην περίπτωση αυτή, μόλις ένας επισκέπτης μίας Σελίδας τοποθετήσει το ποντίκι του σε μία εικόνα, αλλάζει η εμφάνιση όχι μόνο αυτής της εικόνας, αλλά δύο ή και περισσότερων εικόνων.

[41] [43]

2.12.12 Οι Φόρμες και η JavaScript

2.12.12.1 Εισαγωγή

Η χρήση φορμών και η επεξεργασία των στοιχείων τους είναι ένα φαινόμενο που απαντάται πολύ συχνά στο Διαδίκτυο. Η χρήση μίας φόρμας παρέχει την ευχέρεια στη συλλογή στοιχείων διαφόρων τύπων, τα οποία κατόπιν υπόκεινται σε επεξεργασία και για αυτό το λόγο είναι ένα από τα πιο βασικά συστατικά της γλώσσας HTML.

[36]

2.12.12.2 Το Αντικείμενο Form

Το αντικείμενο Form της γλώσσας JavaScript αντιστοιχεί σε μία φόρμα HTML, η οποία τοποθετείται στο εκάστοτε HTML έγγραφο με χρήση της ετικέτας form. Το αντικείμενο Document περιέχει την ιδιότητα forms, η οποία, όπως εξάλλου αναφέρθηκε παραπάνω, είναι ένας πίνακας με τις φόρμες που τυχόν υπάρχουν σε ένα έγγραφο. Ο πίνακας forms, όπως όλοι οι πίνακες της JavaScript, διαθέτει την ιδιότητα length, που δίνει τη δυνατότητα εύρεσης του συνολικού αριθμού των φορμών που υπάρχουν σε ένα έγγραφο. Έτσι, ένας τρόπος, για να γίνει αναφορά στην πρώτη φόρμα που υπάρχει στο εκάστοτε παρόν έγγραφο, είναι να ακολουθηθεί η σύνταξη:

```
document.forms[0];
```

και στην τελευταία:

```
document.forms[document.forms.length-1];
```

Πρέπει, εδώ, να τονιστεί ότι η ιδιότητα length χρησιμοποιείται τόσο για την εύρεση του αριθμού των φορμών που τυχόν υπάρχουν σε μία Ιστοσελίδα όσο και για το πλήθος των στοιχείων (πχ πλαίσια κειμένου, κουμπιά) που ίσως περιέχονται σε μία φόρμα. Για το λόγο αυτό η χρήση της απαιτεί ιδιαίτερη προσοχή. Έτσι, η εντολή:

```
document.forms.length;
```

βρίσκει το πλήθος των φορμών που υπάρχουν σε μία Ιστοσελίδα, ενώ η εντολή:

```
document.forms[i].length;
```

βρίσκει το πλήθος των στοιχείων που υπάρχουν στην i+1-οστή φόρμα μίας Ιστοσελίδας.

Η βασικότερη, τώρα, ιδιότητα του αντικειμένου Form είναι η ιδιότητα-πίνακας elements, που είναι ένας πίνακας του οποίου τα στοιχεία περιέχουν τα διάφορα συστατικά (πχ πλαίσια κειμένου, ραδιοπλήκτρα) που συναντώνται σε μία φόρμα ενός HTML εγγράφου. Η συγκεκριμένη ιδιότητα είναι πολύ χρήσιμη στην προσπέλαση των συστατικών μίας φόρμας. Για παράδειγμα, αν είναι επιθυμητή η προσπέλαση του τέταρτου συστατικού της τρίτης φόρμας του εκάστοτε παρόντος εγγράφου, θα δοθεί η εντολή:

```
document.forms[2].length[3];
```

Οι υπόλοιπες βασικότερες ιδιότητες του αντικειμένου Form, οι περισσότερες από τις οποίες είναι αντίστοιχες αναλόγων ιδιοτήτων της ετικέτας form της γλώσσας HTML, καταγράφονται στον Πίνακα που έπεται:

Ιδιότητα	Περιγραφή
action	Καθορίζει ή επιστρέφει την τιμή της HTML ιδιότητας action μίας φόρμας ενός HTML εγγράφου, η οποία με τη σειρά της καθορίζει πού θα αποσταλούν τα δεδομένα της προαναφερόμενης φόρμας κατά την υποβολή της σε ένα διακομιστή Ιστού.
autocomplete	Καθορίζει ή επιστρέφει την τιμή της HTML ιδιότητας autocomplete μίας φόρμας ενός HTML εγγράφου, η οποία με τη σειρά της υποστηρίζει την αυτόματη συμπλήρωση τιμών των πεδίων της προαναφερόμενης φόρμας με βάση προτύ-τερες τιμές που είχε εισάγει ο εκάστοτε χρήστης. Φυσικά, κάτι τέτοιο προϋποθέ-τει τα cookies να είναι ενεργά στον εκάστοτε Ιστότοπο που περιέχει την εν λόγω φόρμα.
enctype	Καθορίζει ή επιστρέφει την τιμή της HTML ιδιότητας enctype μίας φόρμας ενός HTML εγγράφου, η οποία με τη σειρά της καθορίζει πώς θα κωδικοποιηθούν τα δεδομένα της προαναφερόμενης φόρμας, πριν την υποβολή τους σε ένα διακο-μιστή Ιστού.
length	Επιστρέφει το συνολικό αριθμό των HTML στοιχείων που απαντώνται σε μία φόρμας ενός HTML εγγράφου.
method	Καθορίζει ή επιστρέφει την τιμή της HTML ιδιότητας method σε μία φόρμα ε-νός HTML εγγράφου, η οποία με τη σειρά της καθορίζει με τη χρήση ποιας με-θόδου (get ή post) θα αποσταλούν τα δεδομένα της προαναφερόμενης φόρμας, κατά την υποβολή της σε ένα διακομιστή Ιστού.
name	Καθορίζει ή επιστρέφει την τιμή της HTML ιδιότητας name σε μία φόρμα ενός HTML εγγράφου η οποία με τη σειρά της καθορίζει το όνομα της προαναφερό-μενης φόρμας.

Πίνακας 20: Οι πιο βασικές ιδιότητες του αντικειμένου Form της JavaScript

Πέραν, τώρα, από τις προαναφερθείσες ιδιότητες, το αντικείμενο Form φέρει, επιπρόσθετα, και δύο σημαντικές μεθόδους οι οποίες έχουν να κάνουν με την υποβολή και τον καθαρισμό των πεδίων μίας φόρμας που τυχόν απαντάται σε ένα HTML έγγραφο και οι οποίες παρουσιάζονται στον κάτωθι Πίνακα:

Συνάρτηση	Λειτουργία	Σύνταξη	Παράμετρος	Τιμή Επιστροφής
reset()	Καθαρίζει τα πεδία μίας φόρμας ενός HTML εγγράφου, μό-λις πατηθεί το κουμπί επαναφοράς που εμφα-νίζεται στην προαναφε-ρόμενη φόρμα.	formObject.reset()	Καμία.	Καμία.
submit()	Υποβάλλει μία φόρμα ενός HTML εγγράφου σε ένα διακομιστή Ι-στού, μόλις πατηθεί το κουμπί υποβολής που εμφανίζεται στην προ-αναφερόμενη φόρμα.	formObject.submit()	Καμία.	Καμία.

Πίνακας 21: Οι μέθοδοι του αντικειμένου Form της JavaScript

[41] [42] [43]

2.12.12.3 Τα Βασικά Συστατικά μίας HTML Φόρμας μίας Ιστοσελίδας

Τα κυριότερα στοιχεία μίας HTML φόρμας που πιθανόν απαντάται σε έναν Ιστότοπο είναι τα παρακάτω:

- Το κείμενο (text): Είναι διαθέσιμο με στόχο την ικανότητα γραφής κειμένου εντός μία γραμμής σε μία Ιστοσελίδα. Στη γλώσσα HTML γίνεται χρήση της ετικέτας `input` συνδυαστικά με την ιδιότητά της `type`, στην οποία αποδίδεται η τιμή `text`, για να προκύψει το παρόν στοιχείο σε μία φόρμα ενός HTML εγγράφου. Παράλληλα με την ιδιότητα `type` χρησιμοποιείται και η ιδιότητα `name` της ετικέτας `input`, στην οποία καταχωρείται το όνομα με το οποίο γίνεται αναφορά στο συγκεκριμένο στοιχείο εντός του HTML κώδικα του προαναφερόμενου εγγράφου, καθώς και η ιδιότητα `size` της ετικέτας `input`, η οποία ορίζει το μήκος της γραμμής που θα γραφεί το παραπάνω κείμενο.
- Τα πλαίσια κειμένου (text fields) ή οι περιοχές κειμένου (text areas): Χρησιμοποιούνται, όταν είναι επιθυμητό να γραφεί κείμενο σε μία Ιστοσελίδα, το οποίο, ωστόσο, να επεκτείνεται σε περισσότερες από μία γραμμές και όχι αποκλειστικά εντός μίας γραμμής, όπως συμβαίνει με το στοιχείο `text`. Στη γλώσσα HTML γίνεται χρήση της ετικέτας `textarea` σε συνδυασμό με τις ιδιότητές της `rows` και `cols`, ώστε να δημιουργηθεί μία κατάλληλη περιοχή κειμένου με συγκεκριμένο αριθμό γραμμών και στηλών, αντίστοιχα, σε μία φόρμα ενός HTML εγγράφου. Επίσης, πραγματοποιείται εκμετάλλευση και της ιδιότητας `name` της ετικέτας `textarea`, στην οποία εκχωρείται το όνομα με το οποίο γίνεται αναφορά στην εν λόγω περιοχή κειμένου εντός του HTML κώδικα του παραπάνω εγγράφου.
- Τα κουμπιά (buttons): Διατίθενται, για να εκτελεστεί μία συγκεκριμένη ενέργεια σχετικά με μία HTML φόρμα μίας Ιστοσελίδας, μόλις ο εκάστοτε χρήστης αυτής της Σελίδας πατήσει πάνω τους. Στη γλώσσα HTML γίνεται χρήση της ετικέτας `input` συνδυαστικά με την ιδιότητά της `type`, στην οποία αποδίδεται η τιμή `button`, προκειμένου να προκύψει ένα κουμπί σε μία φόρμα ενός HTML εγγράφου. Παράλληλα με την ιδιότητα `type` γίνεται εκμετάλλευση της ιδιότητας `name` της ετικέτας `input`, στην οποία καταχωρείται το όνομα με το οποίο γίνεται αναφορά στο συγκεκριμένο κουμπί εντός του HTML κώδικα του εν λόγω εγγράφου, καθώς και η ιδιότητα `value` της ετικέτας `input`, η οποία ορίζει το όνομα του εν λόγω κουμπιού, όπως, όμως, αυτό θα παρουσιάζεται στο προαναφερόμενο έγγραφο και όχι στον κώδικά του. Το πιο αντιπροσωπευτικό κουμπί μίας HTML φόρμας είναι το κουμπί υποβολής (submit button) αυτής της φόρμας, με το πάτημα του οποίου προκύπτει αυτό που μαρτυρά, εξάλλου, και το όνομά του, δηλαδή η υποβολή της τρέχουσας φόρμας.
- Τα ραδιοπλήκτρα (radiobuttons): Γίνεται χρήση τους, ώστε να είναι σε θέση ο εκάστοτε χρήστης μίας HTML φόρμας μίας Ιστοσελίδας να επιλέξει ένα και μόνο στοιχείο από μία λίστα ομοειδών αντικειμένων εντός της προηγούμενης φόρμας. Στη γλώσσα HTML γίνεται χρήση της ετικέτας `input` μαζί με την ιδιότητά της `type`, στην οποία δίνεται η τιμή `radio`, ώστε να υλοποιηθεί ένα ραδιοπλήκτρο σε μία φόρμα ενός HTML εγγράφου. Ταυτόχρονα με την ιδιότητα `type` γίνεται εκμετάλλευση της ιδιότητας `name` της ετικέτας `input`, στην οποία καταχωρείται το όνομα με το οποίο δηλώνεται το συγκεκριμένο πλήκτρο εντός του HTML κώδικα του προαναφερόμενου εγγράφου. Να τονιστεί, εδώ, πως σε μία HTML φόρμα πρέπει να υπάρχουν τουλάχιστον δύο ή περισσότερα ραδιοπλήκτρα, επειδή η ύπαρξη ενός μόνο δεν έχει νόημα.
- Τα πλαίσια ελέγχου (check boxes): Διαφέρουν από τα ραδιοπλήκτρα στο πως δίνουν τη δυνατότητα ταυτόχρονης επιλογής περισσότερων του ενός στοιχείων από μία ομάδα στοιχείων σε μία HTML φόρμα μίας Ιστοσελίδας. Στη γλώσσα HTML διατίθεται η ετικέτα `input` σε συνδυασμό με την ιδιότητά της `type`, στην οποία καταχωρείται η τιμή `checkbox`, με σκοπό να υλοποιηθεί ένα πλαίσιο ελέγχου σε μία φόρμα ενός HTML εγγράφου. Συνάμα με την ιδιότητα `type` γίνεται χρήση της ιδιότητας `name` της ετικέτας `input`, στην οποία καταχωρείται το όνομα με το οποίο πραγματοποιείται αναφορά στο συγκεκριμένο πλαίσιο εντός του HTML κώδικα του παραπάνω εγγράφου. Αν, τώρα, ένα πλαίσιο ελέγχου είναι επιλεγμένο, η ιδιότητά του `checked` έχει την τιμή `true`, αλλιώς η τιμή της είναι `false`.
- Οι λίστες (lists): Γίνεται εκμετάλλευση τους, από τη στιγμή που είναι αναγκαία η επιλογή ενός αντικειμένου από μία λίστα ομοειδών αντικειμένων σε μία HTML φόρμα μίας Ιστοσελίδας. Στη γλώσσα HTML μία λίστα κατασκευάζεται με χρήση της ετικέτας `select`. Η ετικέτα `select`

συνοδεύεται από την ιδιότητά της name, στην οποία εκχωρείται το όνομα με το οποίο γίνεται αναφορά στην συγκεκριμένη λίστα εντός του HTML κώδικα ενός εγγράφου. Επιπρόσθετα, καθένα από τα αντικείμενα της μίας λίστας ενός HTML εγγράφου δημιουργείται με την ετικέτα option συνδυαστικά με την ιδιότητα value, η οποία καθορίζει το όνομα καθενός από τα αντικείμενα της εν λόγω λίστας, όπως αυτό θα εμφανίζεται στο πιο πάνω έγγραφο και όχι στον κώδικά του. Η συγκεκριμένη λίστα δύναται να εμφανίζει ένα ή περισσότερα από τα αντικείμενα που περιέχει, ανάλογα με την τιμή που έχει η ιδιότητά της size. Ακόμα, υπάρχει η δυνατότητα χρήσης της ιδιότητας options μίας λίστας, που είναι ένας πίνακας για τον προσδιορισμό ενός συγκεκριμένου στοιχείου αυτής της λίστας, στον οποίο η αρίθμηση, όπως, εξάλλου, σε όλους τους πίνακες ξεκινάει από το μηδέν. Επίσης, είναι εφικτή η διάθεση της ιδιότητας selectedIndex μίας λίστας για τον καθορισμό του δείκτη που αντιστοιχεί σε εκείνο το στοιχείο αυτής της λίστας που έχει επιλέξει ο υφιστάμενος σε κάθε περίπτωση χρήσης.

[41] [42]

2.12.12.4 Η Υποβολή μίας HTML Φόρμας μίας Ιστοσελίδας σε ένα Διακομιστή Ιστού

Στις περισσότερες περιπτώσεις, οι φόρμες που υπάρχουν στο Διαδίκτυο ζητούν από τον εκάστοτε χρήστη να συμπληρώσει κάποια στοιχεία (πχ ονοματεπώνυμο, διεύθυνση, επάγγελμα) και στη συνέχεια τα στοιχεία αυτά υποβάλλονται στον κατάλληλο σε κάθε περίπτωση διακομιστή Ιστού. Στο διακομιστή αυτό υπάρχει ένα πρόγραμμα (στις περισσότερες περιπτώσεις σε μορφή σεναρίου CGI) το οποίο ελέγχει και επεξεργάζεται τα στοιχεία που έδωσε ο χρήστης. Μία τέτοια διαδικασία είναι γνωστή με τον όρο υποβολή διαδικτυακής φόρμας. Για την υποβολή των στοιχείων μίας HTML φόρμας υπάρχει η δυνατότητα να γίνει χρήση είτε του χειριστή συμβάντος onSubmit της γλώσσας JavaScript είτε της μεθόδου submit του αντικειμένου Form της γλώσσας JavaScript, όχι όμως και των δύο ταυτόχρονα. Σε κάθε περίπτωση πρέπει να δοθεί ιδιαίτερη προσοχή στα ακόλουθα σημεία:

- Ο μεν χειριστής συμβάντος onSubmit της γλώσσας JavaScript τοποθετείται στην HTML ετικέτα form εντός του κώδικα μίας Ιστοσελίδας που περιέχει μία HTML φόρμα.. Πρέπει, επίσης, να τοποθετηθεί στην εν λόγω φόρμα ένα ειδικό κουμπί υποβολής, το οποίο έχει οριστεί ως type="submit". Μόλις ο εκάστοτε χρήστης πατήσει σε αυτό το κουμπί, λαμβάνει χώρα ένα συμβάν submit και ακολουθούν οι κατάλληλες ενέργειες για την αποστολή των στοιχείων της συγκεκριμένης φόρμας. Τα δεδομένα της πιο πάνω φόρμας μπορεί να σταλούν ή να μη σταλούν σε περίπτωση που υπάρχει κάποιος λόγος ακύρωσής τους. Αυτό γίνεται με την εντολή return. Στη φάση αυτή γίνεται διαχωρισμός στις πιο κάτω περιπτώσεις:

- Αν είναι επιθυμητή η συνέχιση της διαδικασίας υποβολής μίας HTML φόρμας, ύστερα από το πάτημα ενός κουμπιού τύπου submit από τον εκάστοτε επισκέπτη μίας Ιστοσελίδας, που περιέχει την προαναφερόμενη φόρμα, θα πρέπει να δοθεί η εντολή:

```
onSubmit="return true;"
```

- Αν για κάποιο λόγο είναι επιθυμητή η ακύρωση της διαδικασίας υποβολής μίας HTML φόρμας, ύστερα από το πάτημα ενός κουμπιού τύπου submit από τον εκάστοτε επισκέπτη μίας Ιστοσελίδας, που περιέχει την προαναφερόμενη φόρμα, θα πρέπει να δοθεί η εντολή:

```
onSubmit="return false;"
```

- Αν είναι επιθυμητός ο χειρισμός της διαδικασίας υποβολής μίας HTML φόρμας, ύστερα από το πάτημα ενός κουμπιού τύπου submit από τον εκάστοτε επισκέπτη μίας Ιστοσελίδας, που περιέχει την προαναφερόμενη φόρμα, μέσα από μία συνάρτηση, όπως είναι, για παράδειγμα, η myfunction(), θα πρέπει να δοθεί η εντολή:

```
onSubmit="return myfunction();"
```

Στο σημείο αυτό να τονιστεί ότι, εφόσον κρίνεται αναγκαίο να συνεχιστεί η διαδικασία μετάδοσης της παραπάνω φόρμας, μέσα στην πρωτότερη συνάρτηση είναι επιτακτικό να υπάρχει δήλωση:

```
return true; ,
```

- Η δε μέθοδος submit της γλώσσας JavaScript είναι εφικτό να τοποθετείται οπουδήποτε μέσα σε ένα JavaScript σενάριο και να ενεργοποιείται με ένα χειριστή συμβάντος onclick της συγκεκριμένης γλώσσας, όπως είναι, για παράδειγμα, ο.

[41] [44] [46]

2.12.12.5 Η Επικύρωση των Δεδομένων μίας HTML Φόρμας μίας Ιστοσελίδας

Μία από τις μεγάλες υπηρεσίες που προσφέρει ο Παγκόσμιος Ιστός είναι το ότι δίνει τη δυνατότητα ανταλλαγής δεδομένων μεταξύ χρηστών και διακομιστών Ιστού μέσω των HTML φορμών που πολύ συχνά απαντώνται στις Ιστοσελίδες του Ιστού. Η διαδικασία αυτή της ανταλλαγής των δεδομένων συνίσταται από δύο μέρη: μία HTML φόρμα, όπου ο εκάστοτε χρήστης συμπληρώνει κάποια στοιχεία, και ένα σενάριο CGI (Common Gateway Interface), που βρίσκεται στο διακομιστή. Το σενάριο αυτό ελέγχει και επεξεργάζεται τα στοιχεία που στέλνονται από την προαναφερόμενη φόρμα.

Στον τομέα του ελέγχου των στοιχείων, η γλώσσα JavaScript μπορεί να προσφέρει καθοριστικές υπηρεσίες. Κατά τη συμπλήρωση των πεδίων μίας HTML φόρμας, πολλές φορές ο υφιστάμενος σε κάθε περίπτωση χρήστης κάνει απλά λάθη, όπως είναι, για παράδειγμα, η μη συμπλήρωση ενός τέτοιου πεδίου ή η πληκτρολόγηση μη επιτρεπόμενων χαρακτήρων. Ο έλεγχος της ορθότητας των στοιχείων μίας HTML φόρμας στον υπολογιστή του εκάστοτε χρήστη μπορεί να γίνει με ένα JavaScript σενάριο, κάτι που μειώνει το φόρτο εργασίας του εκάστοτε εξυπηρετητή Ιστού.

[41]

2.12.13 Η Βιβλιοθήκη jQuery της JavaScript

2.12.13.1 Εισαγωγή

Η jQuery είναι μία ελαφριά, τύπου «γράψε λιγότερα, κάνε περισσότερα» βιβλιοθήκη της γλώσσας JavaScript, η οποία επιτρέπει την αναζήτηση και κατά επέκταση την επιλογή συγκεκριμένων HTML στοιχείων μίας Ιστοσελίδας καθώς και την εκτέλεση ενεργειών πάνω σε αυτά τα στοιχεία. Ο σκοπός της συγκεκριμένης βιβλιοθήκης είναι να απλουστεύσει τη χρήση της JavaScript από τους εκάστοτε Ιστοτόπους. Για το λόγο αυτό πακετάρει σε μεθόδους, που μπορούν να καλούνται σε μία και μόνο γραμμή, πολλές συνήθεις εργασίες, οι οποίες απαιτούν αρκετές γραμμές JavaScript κώδικα, για να υλοποιηθούν. Επίσης, απλουστεύει πληθώρα περίπλοκων πραγμάτων της JavaScript, όπως είναι, για παράδειγμα, ο χειρισμός του DOM, οι κλήσεις της τεχνικής AJAX (που θα μελετηθούν στη συνέχεια) και αρκετά ακόμα.

Η βιβλιοθήκη jQuery περιέχει τα ακόλουθα χαρακτηριστικά:

- Χειρισμό του DOM και της γλώσσας HTML.
- Χειρισμό της γλώσσας CSS.
- Μεθόδους HTML γεγονότων.
- Συστατικά της τεχνικής AJAX.
- Διάφορα βοηθητικά για προγραμματισμό Διαδικτύου.

[44]

2.12.13.2 Οι Μέθοδοι Εισαγωγής της jQuery.

Υπάρχουν δύο κύριοι τρόποι εισαγωγής της βιβλιοθήκης jQuery εντός του κώδικα μίας Ιστοσελίδας, οι οποίοι είναι οι παρακάτω:

- I. Το κατέβασμα της βιβλιοθήκη jQuery από την ηλεκτρονική διεύθυνση “jQuery.com”. Σε αυτή την περίπτωση η βιβλιοθήκη jQuery είναι ένα απλό JavaScript αρχείο, στο οποίο υπάρχει η δυνατότητα να γίνει αναφορά μέσα στον κώδικα μίας Ιστοσελίδας, με το ζεύγος HTML ετικετών `<script>` `</script>`. Έπειτα το συγκεκριμένο έγγραφο πρέπει να ενσωματωθεί εντός του ζεύγους των HTML ετικετών `<head>` `</head>` σύμφωνα με την ακόλουθη σύνταξη:

```
<head>
<script src="jquery-1.11.2.min.js"></script>
</head>
```

Π. Η συμπερίληψη της βιβλιοθήκης jQuery μέσα στον κώδικα μίας Ιστοσελίδας από ένα Content Delivery Network (CDN) ,

- είτε της Google, ακολουθώντας τη σύνταξη:

```
<head>
<script src="http://ajax.goog-
leapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
</head>
```

- είτε της Microsoft, ακολουθώντας τη σύνταξη:

```
<head>
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.11.2.min.js"></script>
</head>
```

[44]

2.12.13.3 Η Σύνταξη της jQuery

Η σύνταξη της βιβλιοθήκης jQuery έχει τέτοια μορφή, ώστε να επιλέγει κάποια HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και να διεκπεραιώνει συγκεκριμένες ενέργειες πάνω σε αυτά. Η βασική, λοιπόν, σύνταξη της βιβλιοθήκης jQuery είναι η παρακάτω:

$\$(selector).action()$

Όπου:

- \$: Το σύμβολο του δολαρίου που ορίζει μία βιβλιοθήκη jQuery εντός του κώδικα μίας Ιστοσελίδας.
- selector: Ένας επιλογέας της βιβλιοθήκης jQuery που θα αναζητήσει και θα επιλέξει HTML στοιχεία που απαντώνται μέσα στον κώδικα μίας Ιστοσελίδας.
- action(): Η εκάστοτε jQuery μέθοδος που θα εκτελεστεί πάνω στα HTML στοιχεία που υφίστανται εντός του κώδικα μίας Ιστοσελίδας και τα οποία επιλέχθηκαν με το selector.

[44]

2.12.13.4 Οι Επιλογείς της jQuery

Οι επιλογείς της βιβλιοθήκης jQuery είναι ένα από τα πιο σημαντικά μέρη της jQuery, διότι επιτρέπουν την επιλογή και κατά επέκταση το χειρισμό HTML στοιχείων που βρίσκονται μέσα στον κώδικα μίας Ιστοσελίδας. Ειδικότερα, οι jQuery επιλογείς δίνουν τη δυνατότητα αναζήτησης και επιλογής HTML στοιχείων με βάση τα id τους, τις κλάσεις τους, τους τύπους τους, τις ιδιότητές τους, τις τιμές των ιδιοτήτων τους και αρκετών ακόμα στοιχείων. Οι επιλογείς αυτοί βασίζονται στους CSS επιλογείς και, επιπλέον, διαθέτουν κάποιους δικούς τους εξειδικευμένους επιλογείς. Όλοι, ωστόσο, οι επιλογείς της jQuery ξεκινούν με το σύμβολο \$ και περικλείονται μεταξύ της ακολουθίας χαρακτήρων (“ και ”).

Οι πιο χρήσιμοι επιλογείς της βιβλιοθήκης jQuery φαίνονται στον ακόλουθο Πίνακα:

Επιλογέας	Επιλέγει...
*	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας.
#id	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία έχουν ένα συγκεκριμένο id.

.class	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία ανήκουν σε μία συγκεκριμένη κλάση.
.class1,class2,class3,...,classN	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία ανήκουν τουλάχιστον σε μία από τις κλάσεις 1,2,3,...,N, αντίστοιχα.
element	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία ανήκουν σε μία συγκεκριμένη κατηγορία.
e1,e2,e3,...,eN	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας που ανήκουν τουλάχιστον σε μία από τις κατηγορίες e1,e2,e3,..., eN, αντίστοιχα.
:first	Το πρώτο HTML που υπάρχει μέσα στον κώδικα μίας Ιστοσελίδας.
:last	Το τελευταίο HTML στοιχείο που υπάρχει μέσα στον κώδικα μίας Ιστοσελίδας.
:even	Όλα τα άρτια HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας.
:odd	Όλα τα περιττά HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας.
:header	Όλα τα HTML στοιχεία επικεφαλίδας που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας.
:animated	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία είναι κινούμενες εικόνες με ή χωρίς ήχο.
:focus	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία είναι εστιασμένα.
:contains(text)	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία περιέχουν ένα συγκεκριμένο αλφαριθμητικό.
:has(selector)	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία περιέχουν ένα ή περισσότερα στοιχεία ενός συγκεκριμένου επιλογέα.
:empty	Όλα τα κενά HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας.
:hidden	Όλα τα κρυφά HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας
:visible	Όλα τα ορατά HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας
[attribute]	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία έχουν μία συγκεκριμένη ιδιότητα.
[attribute=value]	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία έχουν μία συγκεκριμένη ιδιότητα και μία καθορισμένη τιμή για αυτή την ιδιότητα.
:input	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία περιλαμβάνονται εντός της ετικέτας input.
:text	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περιλαμβάνονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="text".
:password	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περιλαμβάνονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="password".
:radio	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περιλαμβάνονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="radio".

:checkbox	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περικλείονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="checkbox".
:submit	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περικλείονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="submit".
:reset	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περικλείονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="reset".
:button	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περικλείονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="button".
:image	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περικλείονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="image".
:file	Όλα τα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία όχι μόνο περικλείονται εντός της ετικέτας input αλλά, επιπλέον, φέρουν και την ιδιότητα type="file".
:enabled	Όλα τα ενεργά HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία περικλείονται εντός της ετικέτας input.
:disabled	Όλα τα ανενεργά HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία περικλείονται εντός της ετικέτας input.
:selected	Όλα τα επιλεγμένα HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία περικλείονται εντός της ετικέτας input.
:checked	Όλα τα «τσεκαρισμένα» HTML στοιχεία που υπάρχουν μέσα στον κώδικα μίας Ιστοσελίδας και τα οποία περικλείονται εντός της ετικέτας input.

Πίνακας 22: Οι πιο βασικοί επιλογείς της jQuery

[44]

2.12.13.5 Οι Μέθοδοι της jQuery

Οι μέθοδοι της βιβλιοθήκης jQuery δεν είναι τίποτα άλλο από μία σειρά ενεργειών οι οποίες επηρεάζουν και τελικά διαμορφώνουν με ποικίλους τρόπους τη συμπεριφορά του κώδικα μίας Ιστοσελίδας.

Στο σημείο αυτό πρέπει να τονιστεί πως όλες οι jQuery μέθοδοι ενσωματώνονται σε ένα συμβάν φόρτωσης μίας Ιστοσελίδας εντός του κώδικά της σύμφωνα με τη σύνταξη:

```
$(document).ready(function(){
// jQuery methods go here...
});
```

ή την εναλλακτική, συντομότερη σύνταξη:

```
$function(){
// jQuery methods go here...
};
```

Το παραπάνω γεγονός συμβαίνει, προκειμένου να αποτρέψει τον εκάστοτε jQuery κώδικα να εκτελεστεί, προτού φορτωθεί το εκάστοτε HTML έγγραφο.

Είναι γενικά σωστή πρακτική να τηρείται στάση αναμονής έως ότου ένα HTML έγγραφο να φορτωθεί πλήρως από το εκάστοτε πρόγραμμα περιήγησης, πριν να λάβει χώρα οποιαδήποτε επέμβαση σε αυτό το έγγραφο με διάφορες μεθόδους. Σε διαφορετική περίπτωση, κάποιες ενέργειες που σε κάθε περίπτωση απαιτείται να πραγματοποιηθούν, μπορεί να αποτύχουν, όπως είναι, για παράδειγμα:

- Η προσπάθεια απόκρυψης ενός HTML στοιχείου μίας Ιστοσελίδας, το οποίο δεν έχει καν ακόμα δημιουργηθεί.
- Η προσπάθεια υπολογισμού του μεγέθους μίας εικόνας μίας Ιστοσελίδας, η οποία δεν έχει καν ακόμα φορτωθεί.

[44]

2.12.14 Σύγκριση της JavaScript

Μερικές από τις δυνατότητες που προσφέρει η γλώσσα JavaScript είναι οι εξής:

- Έλεγχος στο περιεχόμενο και την εμφάνιση των Ιστοσελίδων.
- Έλεγχος της συμπεριφοράς και των λειτουργιών του εκάστοτε προγράμματος περιήγησης.
- Αλληλεπίδραση με HTML φόρμες.
- Αλληλεπίδραση με τον εκάστοτε χρήστη, με τη βοήθεια των χειριστών γεγονότων (event handlers).
- Ανάγνωση ή καταγραφή της κατάστασης του υπολογιστή του εκάστοτε χρήστη, με χρήση cookies.
- Δυνατότητες, όπως οι εναλλαγές εικόνων, που δίνουν μία δυναμική αίσθηση στην εκάστοτε Ιστοσελίδα, η αλληλεπίδραση με μικροεφαρμογές (applets) (τις οποίες θα μελετήσουμε στη συνέχεια) της Java, ο χειρισμός της ημερομηνίας και της ώρας, παροχή πληροφοριών για την οθόνη του εκάστοτε χρήστη, και, βέβαια, όλες οι δυνατότητες υπολογισμών και λήψης αποφάσεων, που συναντώνται σε μία κανονική γλώσσα προγραμματισμού.

Αφού εξετάστηκαν οι δυνατότητες της γλώσσας JavaScript, ας ριχθεί μία σύντομη ματιά και στους βασικότερους περιορισμούς της εν λόγω γλώσσας:

- Για λόγους ασφαλείας, η γλώσσα JavaScript δεν παρέχει τη δυνατότητα προσπέλασης ή χειρισμού αρχείων στον υπολογιστή του υφιστάμενου σε κάθε περίπτωση χρήστη. Η εγγραφή δεδομένων είναι, επίσης, περιορισμένη αποκλειστικά και μόνο στη δημιουργία cookies. Τα cookies αποτελούνται αποκλειστικά και μόνο από κείμενο και δε μπορούν να εκτελεστούν στον υπολογιστή του εκάστοτε χρήστη. Μία τέτοια δυνατότητα προφυλάσσει τον τρέχων κάθε φορά χρήστη από σενάρια τα οποία θα μπορούσαν να προξενήσουν βλάβες στον υπολογιστή του.
- Η γλώσσα JavaScript δεν υποστηρίζει δυνατότητες δικτύωσης οποιοσδήποτε μορφής, ούτε διαθέτει εντολές δημιουργίας γραφικών ή σχεδίων.
- Ένα σενάριο της γλώσσας JavaScript εκτελείται στον υπολογιστή του χρήστη που επισκέπτεται μία Ιστοσελίδα, δηλαδή στον υπολογιστή πελάτη (client), χωρίς να έχει τη δυνατότητα να ενημερώνει αρχεία που βρίσκονται σε ένα διακομιστή του Ιστού.

[41][45][46]

2.13 Η Επεκτάσιμη Γλώσσα Σήμανσης Περιεχομένου Ιστοσελίδων, XML

2.13.1 Εισαγωγή στην XML

Το 1990 δημιουργήθηκε η γλώσσα HTML από τους Tim Berners-Lee και Anders Berglund, οι οποίοι καθόρισαν έναν τύπο SGML εγγράφου για έγγραφα υπερκειμένου, το οποίο ήταν συναφές και αποτελεσματικό. Είναι πολύ εύκολο να υλοποιηθεί λογισμικό που να υποστηρίζει την HTML, ενώ η δημιουργία ενός HTML εγγράφου είναι ακόμα πιο εύκολη. Για το λόγο αυτό η HTML χρησιμοποιείται ευρέως.

Για να επιτευχθεί η απαραίτητη απλότητα από τη γλώσσα HTML, μερικές από τις βασικές αρχές γενικού προγραμματισμού θυσιάστηκαν. Για παράδειγμα, ένας τύπος εγγράφου χρησιμοποιείται για όλους τους σκοπούς, με αποτέλεσμα να είναι απαραίτητη η υπερβολική χρήση ετικετών, αντί για τον καθορισμό νέων με συγκεκριμένο σκοπό. Επιπλέον, οι περισσότερες από τις ετικέτες σκοπεύουν μόνο

στη διαμόρφωση της εμφάνισης ενός εγγράφου. Για την επίλυση αυτών των προβλημάτων έγινε προσπάθεια υιοθέτησης της γλώσσας SGML για χρήση στο Διαδίκτυο, αλλά κάτι τέτοιο αποδείχθηκε πολύπλοκο και πολύ δύσκολο. Επομένως, ήταν αναγκαίος ο καθορισμός μίας νέας γλώσσας, η οποία να είναι υποσύνολο της SGML, αλλά να διατηρεί τα βασικά χαρακτηριστικά της. Η γλώσσα αυτή είναι η Επεκτάσιμη Γλώσσα Σήμανσης (Extensible Markup Language-XML).

Η XML, λοιπόν, είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση εγγράφων του Παγκόσμιου Ιστού. Οι σχεδιαστικοί της στόχοι είναι οι εξής:

- Να είναι εύχρηστη στο Διαδίκτυο.
- Να υποστηρίζει πλήθος διαδικτυακών εφαρμογών.
- Να είναι συμβατή με τη γλώσσα SGML.
- Να διευκολύνει την ανάπτυξη προγραμμάτων που επεξεργάζονται XML έγγραφα.
- Να έχει όσο το δυνατό μικρότερο αριθμό προαιρετικών χαρακτηριστικών.
- Ο σχεδιασμός των XML εγγράφων να προετοιμάζεται γρήγορα και να είναι περιεκτικός και τυπικός.
- Τα XML έγγραφα να δημιουργούνται εύκολα και να είναι ευανάγνωστα.
Τα κύρια, τώρα, χαρακτηριστικά της γλώσσας είναι τα επόμενα:
- Παρέχει ακριβή καθορισμό του περιεχομένου μίας Ιστοσελίδας, ώστε τα αποτελέσματα μίας αναζήτησης να είναι ικανοποιητικά.
- Τα XML έγγραφα διαθέτουν την ικανότητα καθορισμού των στοιχείων τους (π.χ. τίτλος, συγγραφέας, εκδότης).
- Επιτρέπει τη μεταφορά δομημένης πληροφορίας μέσω του Διαδικτύου, ανεξάρτητα από πλατφόρμες και συστήματα.

[47]

2.13.2 Η Μοντελοποίηση της XML

Ένα από τα σημαντικότερα πλεονεκτήματα της γλώσσας XML είναι η δυνατότητα που παρέχει για καθορισμό νέων γλωσσών σήμανσης, καθορίζοντας στοιχεία και ιδιότητες τα οποία περιγράφουν με τον καλύτερο τρόπο ένα συγκεκριμένο είδος πληροφορίας. Για τον πλήρη και τυπικό καθορισμό μίας νέας γλώσσας σήμανσης, για τον περιορισμό του συνόλου των στοιχείων και των ιδιοτήτων αυτής της καινούργιας γλώσσας και για τον έλεγχο της γραμματικής και της σύνταξης των ετικετών της προαναφερόμενης νέας γλώσσας χρησιμοποιούνται τα μοντέλα εγγράφων (document models).

Τα μοντέλα εγγράφων καθορίζουν ποια έγγραφα του Παγκοσμίου Ιστού συμμορφώνονται με τη γλώσσα XML. Το μοντέλο είναι ένα έγγραφο που έχει υλοποιηθεί με βάση το συντακτικό της XML και ορίζει μονοσήμαντα τη γραμματική και το σύνολο των ετικετών αυτής της γλώσσας. Η γλώσσα σήμανσης διαδικτυακού περιεχομένου που καθορίζεται από το προαναφερόμενο μοντέλο ονομάζεται XML τύπος εγγράφου ή XML εφαρμογή.

Η χρήση ενός μοντέλου εγγράφου είναι προαιρετική, καθώς η γλώσσα XML έχει σχεδιαστεί, ώστε να μην είναι απαραίτητο αυτό το έγγραφο. Επιπρόσθετα, η χρήση ενός τέτοιου εγγράφου εξαρτάται αποκλειστικά από την εφαρμογή και τον δημιουργό της. Η χρήση, όμως, μοντέλων εγγράφων έχει σημαντικά πλεονεκτήματα, τα σπουδαιότερα από τα οποία αναφέρονται παρακάτω:

- Οι προκαθορισμένοι κανόνες των μοντέλων εγγράφων καθιστούν πιο εύκολη τη δημιουργία λογισμικού για έγγραφα και ελαχιστοποιούν την πιθανότητα λαθών.
- Όταν ένα έγγραφο απαιτεί συγκεκριμένα στοιχεία, αυτά μπορούν να καθοριστούν με τη βοήθεια μοντέλων εγγράφων
- Τα μοντέλα εγγράφων καθοδηγούν τον αδαή χρήστη σε ένα περιβάλλον συγγραφής XML εγγράφων, παρέχοντάς του μία λίστα από τα απαραίτητα στοιχεία.

Τα μειονεκτήματα της χρήσης μοντέλων εγγράφων είναι η δυσκολία διαχείρισής τους, η καθυστέρηση κατά την επεξεργασία των εν λόγω εγγράφων και η παρεμπόδιση δημιουργίας νέων στοιχείων σήμανσης όσον αφορά τα αντικείμενα μίας Ιστοσελίδας, εφόσον το εκάστοτε μοντέλο καθορίζει ποια στοιχεία μπορούν ή όχι να χρησιμοποιηθούν, σχετικά με τη μορφοποίηση των προαναφερόμενων αντικειμένων.

Οι πιο δημοφιλείς τύποι μοντέλων εγγράφων του Παγκόσμιου Ιστού είναι οι δύο επόμενοι:

- I. DTD (Document Type Definition) Έγγραφα (DTD Documents).

II. XML Σχήματα (XML Schemas).

Ένα DTD Έγγραφο καθορίζει τον τύπο ενός εγγράφου του Παγκόσμιου Ιστού με τους ακόλουθους τρόπους:

- Προδιαγράφοντας το σύνολο των επιτρεπτών στοιχείων ενός εγγράφου του Ιστού. Απαγορεύεται η χρήση άλλων ονομάτων για τα στοιχεία ενός εγγράφου του Ιστού, εκτός από αυτά που καθορίζονται από το παραπάνω σύνολο. Το συγκεκριμένο σύνολο στοιχείων ονομάζεται και «λεξικό» μίας γλώσσας σήμανσης.
- Προσδιορίζοντας το μοντέλο περιεχομένου ενός εγγράφου του Ιστού. Το μοντέλο περιεχομένου είναι ένα υπόδειγμα το οποίο καθορίζει ποια στοιχεία ή δεδομένα μπορεί να είναι μέρος ενός εγγράφου του Ιστού, με ποια σειρά, πόσες φορές και αν είναι υποχρεωτικά ή προαιρετικά. Το μοντέλο περιεχομένου ονομάζεται και «γραμματική» μίας γλώσσας σήμανσης.
- Στοιχειοθετώντας ένα σύνολο επιτρεπτών ιδιοτήτων για κάθε στοιχείο ενός εγγράφου του Ιστού. Κάθε ορισμός μίας τέτοιας ιδιότητας καθορίζει το όνομά της, τον τύπο δεδομένων της, τις προκαθορισμένες τιμές της (αν υπάρχουν) και τη συμπεριφορά της (υποχρεωτική ή προαιρετική).
- Παρέχοντας ένα πλήθος μηχανισμών για την εύκολη διαχείριση ενός εγγράφου του Ιστού, όπως είναι, για παράδειγμα η χρήση οντοτήτων ή εισαγωγή τμημάτων από εξωτερικά αρχεία.

Ωστόσο τα DTD Έγγραφα παρουσιάζουν κάποια μειονεκτήματα, όπως είναι, για παράδειγμα η περιορισμένη δυνατότητα καθορισμού τύπων δεδομένων και η διαφορετική σύνταξη ενός DTD από τα XML έγγραφα. Για το λόγο αυτό, εναλλακτικά διατίθενται τα XML Σχήματα, δίνοντας με αυτό τον τρόπο λύση στους παραπάνω περιορισμούς. Τα σημαντικότερα πλεονεκτήματα των XML Σχημάτων σε σχέση με τα DTD Έγγραφα είναι τα ακόλουθα:

- Υποστηρίζουν περισσότερους από 44 τύπους δεδομένων σε ένα έγγραφο του Ιστού και επιτρέπουν τη δημιουργία καινούριων.
- Έχουν την ίδια σύνταξη με ένα XML έγγραφο.
- Είναι αντικειμενοστραφείς, επιτρέποντας την επέκταση ή τον περιορισμό ενός τύπου δεδομένων ενός εγγράφου του Ιστού.
- Επιτρέπουν την δημιουργία στοιχείων, χωρίς να είναι σημαντική η σειρά εμφάνισής τους σε ένα έγγραφο του Ιστού.
- Επιτρέπουν τον ορισμό περιεχομένου ενός στοιχείου εγγράφου του Ιστού ως μοναδικό (key).
- Επιτρέπουν τον καθορισμό πολλαπλών στοιχείων με το ίδιο όνομα και διαφορετικό περιεχόμενο για κάθε έγγραφο του Ιστού, χωρίς να δημιουργείται σύγχυση κατά την επεξεργασία των προαναφερόμενων στοιχείων.

Η επιλογή ανάμεσα στα δύο πιο πάνω μοντέλα εγγράφων είναι θέμα προγραμματιστή και εφαρμογής. Τα XML Σχήματα είναι μεταγενέστερα και παρουσιάζουν σημαντικά πλεονεκτήματα έναντι των DTD, αλλά δεν τα αντικαθιστούν. Τα DTD πλεονεκτούν σε μέγεθος, έχουν γνώριμη και απλή σύνταξη στην οποία πολλοί προγραμματιστές του Διαδικτύου είναι ήδη εξοικειωμένοι.

Μερικές εφαρμογές της γλώσσας XML, υλοποιημένες με βάση τα DTD ή τα XML Σχήματα, είναι οι παρακάτω:

- Chemical Markup Language (CML): Γλώσσα σήμανσης που χρησιμοποιείται από χημικούς για την περιγραφή πολύπλοκων χημικών μορίων.
- Mathematical Markup Language (MML): Γλώσσα σήμανσης για την περιγραφή εξισώσεων στο Διαδίκτυο.
- Vector Markup Language (VML): Γλώσσα σήμανσης για το σχεδιασμό γραφικών που περιγράφονται με διανύσματα.
- Resource Description Framework (RDF): Γλώσσα σήμανσης για την αναπαράσταση μεταδεδομένων.

[47]

2.13.3 Οι Κατηγορίες της XML

Υπάρχουν δύο τύποι XML εγγράφων:

- I. Τα ορθά μορφοποιημένα ή καλά ορισμένα (well-formed) έγγραφα: Ακολουθούν τους γενικούς κανόνες σύνταξης της γλώσσας XML, οι οποίοι θα αναπτυχθούν στη συνέχεια και είναι πιο αυστηροί από τους κανόνες των γλωσσών HTML και SGML.
- II. Τα έγκυρα (valid) έγγραφα: Ακολουθούν ένα μοντέλο εγγράφου, δηλαδή ένα σετ κανόνων που

ένα έγγραφο του Παγκοσμίου Ιστού ακολουθεί, και τους οποίους το εκάστοτε λογισμικό έχει την ικανότητα να διαβάσει πριν την επεξεργασία και την εμφάνιση ενός τέτοιου εγγράφου. Αυτοί οι κανόνες γενικά καθορίζουν το όνομα και το περιεχόμενο κάθε στοιχείου (element) μίας Ιστοσελίδας. Τα έγκυρα XML έγγραφα προσφέρουν πολύ περισσότερα στην επεξεργασία ενός XML εγγράφου από τα αντίστοιχα ορθά μορφοποιημένα. Έτσι, η συγγραφή, επεξεργασία, αποθήκευση, και εμφάνιση τέτοιας μορφής εγγράφων είναι ευκολότερες, γιατί τα έγγραφα αυτά υπάρχουν σε ένα δομικό περιβάλλον.

Η βασική διαφορά μεταξύ των ορθά μορφοποιημένων και των έγκυρων XML εγγράφων είναι η σχέση τους με το εκάστοτε μοντέλο εγγράφου. Με άλλα λόγια οι δημιουργοί των XML εγγράφων πρέπει να συντάσσουν έγκυρα έγγραφα, ενώ ένα ικανό XML λογισμικό πλοήγησης απαιτείται μόνο να ελέγχει τα XML Έγγραφα για ορθή μορφοποίηση, πριν τα εμφανίσει.

[47]

2.13.4 Ο Επεξεργαστής και ο Αναλυτής της XML

Το λογισμικό μοντέλο που καλείται XML επεξεργαστής (XML processor) χρησιμοποιείται, για να διαβάσει XML έγγραφα και παρέχει πρόσβαση στο περιεχόμενο και τη δομή τους. Ένας XML επεξεργαστής λειτουργεί εκ μέρους ενός άλλου μοντέλου που καλείται εφαρμογή. Η υφιστάμενη σε κάθε περίπτωση εφαρμογή περιγράφει την απαιτούμενη συμπεριφορά του κάθε φορά συσχετιζόμενου με αυτή την εφαρμογή XML επεξεργαστή και συγκεκριμένα τον τρόπο με τον οποίο θα πρέπει να ο συγκεκριμένος επεξεργαστής να διαβάζει τα XML δεδομένα καθώς και το είδος των πληροφοριών που ο εν λόγω επεξεργαστής πρέπει να παρέχει στην προαναφερόμενη εφαρμογή. Παραδείγματα XML επεξεργαστών συμπεριλαμβάνουν εφαρμογές ελέγχου εγκυρότητας (XML validation Tools), περιηγητές (browsers) εργαλεία συγγραφής (XML Editors) και επεξεργασίας-διαχείρισης (XML management tools) XML εγγράφων

Ένας XML αναλυτής (XML parser) είναι ένα λογισμικό μοντέλο το οποίο διαβάζει XML αρχεία, ελέγχει την ορθή μορφοποίησή τους και την εγκυρότητά τους και τα προωθεί για περαιτέρω επεξεργασία. Ένας XML αναλυτής είναι η πιο απλή μορφή ενός XML επεξεργαστή και είναι απαραίτητο τμήμα κάθε XML εφαρμογής. Ο εκάστοτε αναλυτής μετατρέπει την χωρίς νόημα ακολουθία χαρακτήρων ενός XML εγγράφου σε τμήματα (tokens) που έχουν νόημα. Τα τμήματα αυτά είτε ερμηνεύονται ως γεγονότα (events) τα οποία καθοδηγούν ένα πρόγραμμα είτε δημιουργούν μία προσωρινή δενδροειδής δομή (tree structure) στη μνήμη, πάνω στην οποία το εκάστοτε πρόγραμμα μπορεί να επέμβει.

[47] [48]

2.13.5 Οι Ενότητες της XML

Όλα τα XML έγγραφα αποτελούνται από δύο βασικές ενότητες: Τον XML πρόλογο (XML prologue) και το XML στοιχείο «ρίζα» (XML “root” element).

Ο πρόλογος, τώρα, ενός XML εγγράφου απαρτίζεται, επίσης, από δύο επιμέρους υποενότητες: Την XML δήλωση (XML declaration), η ύπαρξη της οποίας σε ένα XML έγγραφο κρίνεται υποχρεωτική, και τη δήλωση του τύπου ενός XML εγγράφου (XML document type declaration), που η παρουσία της σε ένα XML έγγραφο δεν είναι επιτακτική,

Όσον αφορά την XML δήλωση ενός XML εγγράφου, ενημερώνει τον εκάστοτε XML επεξεργαστή ότι το έγγραφο χρησιμοποιεί την XML ως γλώσσα σήμανσης. Η δήλωση της XML ξεκινά με τους πέντε χαρακτήρες `<?xml`, ακολουθείται από ορισμένες χαρακτηριστικά (γνωρίσματα) ιδιότητες (attributes) με τις αντίστοιχες τιμές τους, στη μορφή ιδιότητα=τιμή, και τερματίζει με έναν οριοθέτη δύο χαρακτήρων `>`. Η σύνταξή της είναι, με άλλα λόγια, στην πλήρη μορφή της, κάπως έτσι:

```
<?xml attribute1="value1" attribute2="value2" attribute3="value3" ... attributeN="valueN"?>
```

Κατά την XML δήλωση ενός XML εγγράφου, είναι δυνατό να χρησιμοποιηθούν οι εξής τρεις ιδιότητες:

- version: Ορίζει τον αριθμό της XML έκδοσης που διατίθεται για τη σήμανση του εκάστοτε XML εγγράφου. Η ιδιότητα αυτή πρέπει πάντα να ορίζεται στην εκάστοτε XML δήλωση.
- encoding: Ορίζει το είδος της κωδικοποίησης των χαρακτήρων που είναι χρησιμοποιείται στο εκάστοτε έγγραφο, και παίρνει διάφορες τιμές, όπως είναι, για παράδειγμα, οι UTF-8, US-

ASCII, ISO-8859-1. Η χρήση άλλου είδους κωδικοποίησης πέραν του προτύπου UTF-8, που παρέχει τους Λατινικούς χαρακτήρες, θα πρέπει να δηλώνεται στην εκάστοτε XML δήλωση.

- standalone: Μέσω της ιδιότητας αυτής, η χρήση της οποίας είναι προαιρετική, ο εκάστοτε XML επεξεργαστής εξετάζει αν υπάρχουν και άλλα αρχεία που συνοδεύουν το εκάστοτε XML έγγραφο και τα οποία θα πρέπει να φορτώσει μαζί με αυτό. Συνεπώς, η συγκεκριμένη ιδιότητα δέχεται τις τιμές:
 - yes: Όταν το εκάστοτε XML έγγραφο δε συνοδεύεται από άλλα επιμέρους αρχεία, γεγονός που οδηγεί στη βελτιστοποίηση της ταχύτητας φόρτωσης της εκάστοτε Ιστοσελίδας.
 - no: Όταν το εκάστοτε XML έγγραφο συνοδεύεται και από άλλα επιμέρους αρχεία, που πρέπει ο εκάστοτε XML επεξεργαστής να τα φορτώσει μαζί με το προαναφερόμενο κυρίως αρχείο, γεγονός που οδηγεί στη μείωση της ταχύτητας φόρτωσης της εκάστοτε Ιστοσελίδας.

Σχετικά με το δεύτερο μέρος του προλόγου ενός XML εγγράφου, που δεν είναι άλλο από τη δήλωση του τύπου ενός XML εγγράφου, είναι δυνατό να οριστούν ποικίλες παράμετροι, μεταξύ των οποίων συγκαταλέγονται το DTD Έγγραφο που χρησιμοποιείται, το όνομα του στοιχείου «ρίζας» και οι δηλώσεις οντοτήτων. Όταν γίνεται αναφορά σε ένα DTD, τότε ο XML εκάστοτε αναλυτής συγκρίνει το εκάστοτε παρόν XML έγγραφο με το προαναφερόμενο DTD. Μία τέτοια διαδικασία λέγεται έλεγχος εγκυρότητας και δεν είναι υποχρεωτική, αλλά θεωρείται χρήσιμη, προκειμένου να εξακριβωθεί, αν το εκάστοτε XML έγγραφο ακολουθεί ορισμένα πρότυπα δομής και περιλαμβάνει τα απαιτούμενα δεδομένα.

Έτσι, η σύνταξη της δήλωσης του τύπου ενός XML εγγράφου αρχίζει με το <!DOCTYPE ακολουθούμενο από το στοιχείο «ρίζα», το οποίο είναι το πρώτο XML στοιχείο που εμφανίζεται στο εκάστοτε έγγραφο και το οποίο περιέχει την υπόλοιπη περιοχή αυτού του εγγράφου. Αν μαζί με το εκάστοτε τρέχον XML έγγραφο, χρησιμοποιείται και ένα DTD Έγγραφο για τη μοντελοποίηση της δομής του, τότε θα πρέπει να συμπεριληφθεί ο Ενιαίος Αναγνωριστής Πόρων (Uniform Resource Identifier-URI) αυτού του DTD, έτσι ώστε ο εκάστοτε XML επεξεργαστής να είναι σε θέση να το αναζητήσει. Μετά από τα παραπάνω είναι πιθανή η ύπαρξη ενός εσωτερικού υποσυνόλου, που περικλείεται από ένα σετ αγκυλών [], και το οποίο παρέχει τη δυνατότητα εισαγωγής ποικίλων δηλώσεων, οι οποίες χρησιμοποιούνται από το υπάρχον σε κάθε φάση XML έγγραφο. Αυτές οι δηλώσεις ίσως να περιλαμβάνουν ορισμούς οντοτήτων και μέρη του πιο πάνω DTD. Τα εσωτερικά υποσύνολα είναι τα μόνα που μπορούν να δεχθούν αυτές τις δηλώσεις, μέσα στο ίδιο το έγγραφο. Η δήλωση τελειώνει με το χαρακτήρα > . Δηλαδή η γενική μορφή της δήλωσης του τύπου ενός XML εγγράφου είναι η πιο κάτω:

```
<!DOCTYPE root-element uri-of-dtd  
[internal-subset ]  
>
```

Αναφορικά, τώρα με στοιχείο «ρίζα» ενός XML εγγράφου, περιλαμβάνει το υπόλοιπο μέρος ενός XML εγγράφου, εκτός από τον πρόλόγο του.

[47] [49]

2.13.6 Η Δομή της XML

Η γλώσσα XML, όπως εξάλλου προαναφέρθηκε, περιγράφει μία κατηγορία πληροφοριών που καλούνται XML έγγραφα (XML documents) καθώς, επίσης, περιγράφει τμηματικά τη συμπεριφορά των προγραμμάτων που τα επεξεργάζονται. Τα XML έγγραφα είναι αυστηρά δομημένα και μπορούν να αναπαρασταθούν με δενδρική δομή. Για να κατανοηθεί επαρκέστερα η συγκεκριμένη δομή, παρατίθεται το παρακάτω παράδειγμα που απεικονίζει ένα απλό XML έγγραφο το οποίο με τη σειρά του αναπαριστά ένα βιβλίο ενός βιβλιοπωλείου.

```
<bookstore>  
  <book category=" COOKING">  
    <title lang="en">Everyday Italian</title>  
    <author>Giada De Laurentiis</author>  
    <year>2005</year>
```

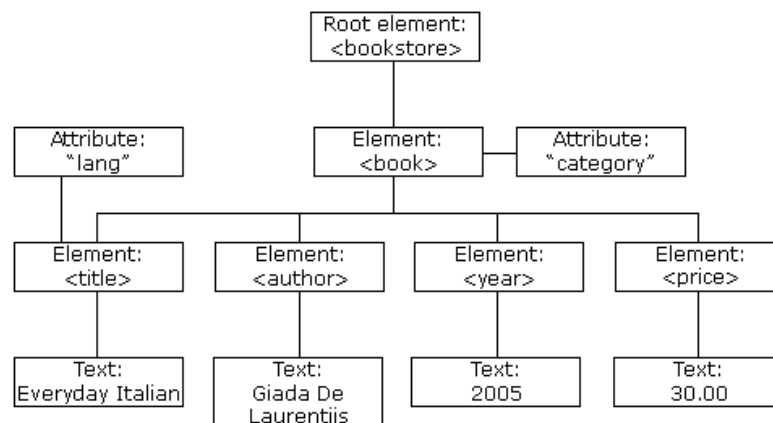
```

    <price>30.00</price>
  </book>
</bookstore>

```

Το πρώτο στοιχείο ενός XML εγγράφου, που περικλείει όλα τα άλλα και τοποθετείται στην κορυφή του δένδρου, ονομάζεται στοιχείο «ρίζα» (“root” element). Επίσης, το στοιχείο «ρίζα» ονομάζεται και στοιχείο ενός XML εγγράφου, επειδή περιλαμβάνει όλα τα υπόλοιπα στοιχεία του και, επιπλέον, καθορίζει τα όριά του. Ακόμα, το στοιχείο «ρίζα» ενός εγγράφου ονομάζεται και «πρόγονος» (“ancestor”) όλων των άλλων. Τα στοιχεία στο τέλος κάθε αλυσίδας της δενδρικής δομής ενός XML εγγράφου ονομάζονται «φύλλα» (“leaves”) και αναπαριστούν το περιεχόμενο αυτού του εγγράφου. Τα στοιχεία που βρίσκονται ακριβώς κάτω από το στοιχείο «ρίζα» ενός εγγράφου αποτελούν τα «παιδιά» του (“child” elements). Το καθένα από αυτά τα «παιδιά» με τη σειρά του μπορεί να είναι «γονέας» (“parent”) άλλων δικών του «παιδιών» και η τρέχουσα διαδικασία να συνεχίζεται μέχρι να φτάσει στο στοιχείο-κόμβο-«γονέας» που δεν έχει «παιδί». Τα στοιχεία ενός εγγράφου που μοιράζονται τον ίδιο «γονέα» ονομάζονται «αδέρφια» (“siblings”). Κάθε στοιχείο ενός εγγράφου πρέπει να ανήκει σε κάποιο άλλο, δηλαδή να είναι «παιδί» κάποιου «γονέα». Κλείνοντας, να αναφερθεί ότι κάθε στοιχείο ενός εγγράφου ονομάζεται και κόμβος» (node). Κάθε κόμβος σε ένα δένδρο που αναπαριστά ένα XML έγγραφο μπορεί να θεωρηθεί ως η ρίζα σε ένα μικρότερο υπόδενδρο. Τα υπόδενδρα αυτά έχουν όλες τις ιδιότητες ενός κανονικού δένδρου και η κορυφή κάθε υπόδενδρου είναι ο «γονέας» όλων των «παιδιών» που βρίσκονται από κάτω.

Ύστερα από την ανάλυση που πραγματοποιήθηκε είναι εφικτή η απόδοση της δενδρικής μορφή του παραπάνω XML κώδικα σύμφωνα με την ακόλουθη Εικόνα:



Εικόνα 18: Η δενδρική μορφή ενός εγγράφου της XML

[48]

2.13.7 Τα Βασικά Συστατικά της XML

Το πρώτο και βασικότερο συστατικό ενός XML εγγράφου είναι τα στοιχεία (elements), τα οποία αποτελούν τα κυριότερα μέρη ενός XML εγγράφου.

Ένα XML στοιχείο ορίζεται ως εξής:

```

<element attribute1="value1 attribute2="value2" attribute3="value3" ... attributeN="valueN">Content</element>

```

Όπου:

- element: Το όνομα του εκάστοτε XML στοιχείου ενός XML εγγράφου.
- attribute1, attribute2, attribute3, ..., attributeN: Τα ονόματα κατάλληλων ιδιοτήτων 1, 2, ..., N, αντίστοιχα, που προσδιορίζουν περαιτέρω το element.
- value1, value2, value3, ..., valueN: Οι τιμές των ιδιοτήτων attribute1, attribute2, attribute3, ..., attributeN, αντίστοιχα.
- Content: Ενδεδειγμένο κείμενο ή επιπλέον XML στοιχεία που δύναται να συμπεριλαμβάνει το element.

Αν, τώρα, το Content είναι κενό, τότε ορίζεται ένα XML κενό στοιχείο (empty element), το οποίο συνήθως δε συνοδεύεται από επιπρόσθετες ιδιότητες, οπότε και η σύνταξή του είναι η ακόλουθη:

```
<emptyElement> </emptyElement>
```

Όπου emptyElement το όνομα του εκάστοτε κενού XML στοιχείου ενός XML εγγράφου

Η πιο πάνω σύνταξη απαντάται και σε συνεπτυγμένη μορφή, η οποία έχει ως εξής:

```
<emptyElement />
```

Σε αυτό το σημείο, να σημειωθεί πως ένα XML στοιχείο ενός XML εγγράφου μπορεί να επεκταθεί, ώστε να κουβαλάει επιπρόσθετη πληροφορία, δίχως να προκαλεί σφάλμα στον εκάστοτε XML αναλυτή.

Τελειώνοντας, αναφορικά με το όνομα ενός XML στοιχείου ενός XML εγγράφου, θεωρείται ορθό να ξεκινάει με ένα γράμμα ή με το χαρακτήρα υπογράμμισης (_) και μπορεί να περιέχει έναν οποιονδήποτε αριθμό από γράμματα οποιουδήποτε αλφαβήτου, αριθμούς, παύλες, και χαρακτήρες υπογράμμισης. Ωστόσο, επειδή η άνω και κάτω τελεία (:) χρησιμοποιείται σε χώρους ονομασίας, συνιστάται να αποφεύγεται η χρήση της στα ονόματα των στοιχείων τα οποία δεν χρησιμοποιούν ένα χώρο ονομασίας. Κενά, ετικέτες, νέες σειρές, εισαγωγικοί χαρακτήρες και διαχωριστές καθώς και το σύμβολο ισότητας δεν πρέπει να χρησιμοποιούνται στα ονόματα τόσο των XML στοιχείων όσο των ιδιοτήτων και των τιμών τους (που θα μελετηθούν αμέσως παρακάτω).

Δεύτερο κύριο συστατικό ενός XML εγγράφου είναι οι τα χαρακτηριστικά (γνωρίσματα) ιδιότητες (attributes) ενός XML στοιχείου αυτού του εγγράφου. Οι ιδιότητες παρέχουν πρόσθετη πληροφορία στα XML στοιχεία πέρα από το όνομά τους ή το περιεχόμενό τους. Κάθε XML στοιχείο δύναται να έχει μία ή περισσότερες ιδιότητες, εφόσον το κάθε στοιχείο έχει μοναδικό όνομα. Ωστόσο η ίδια ιδιότητα πρέπει να εμφανίζεται μία μόνο φορά σε κάθε στοιχείο.

Μία XML ιδιότητα ενός XML στοιχείου ενός XML εγγράφου ακολουθεί την παρακάτω σύνταξη:

```
attribute="value"
```

Όπου:

- attribute: Το όνομα της εκάστοτε XML ιδιότητας ενός XML στοιχείου ενός XML εγγράφου.
- value: Η τιμή που καταχωρείται στην attribute.

Πιο σπάνια η XML ιδιότητα συντάσσεται και ως εξής:

```
attribute='value'
```

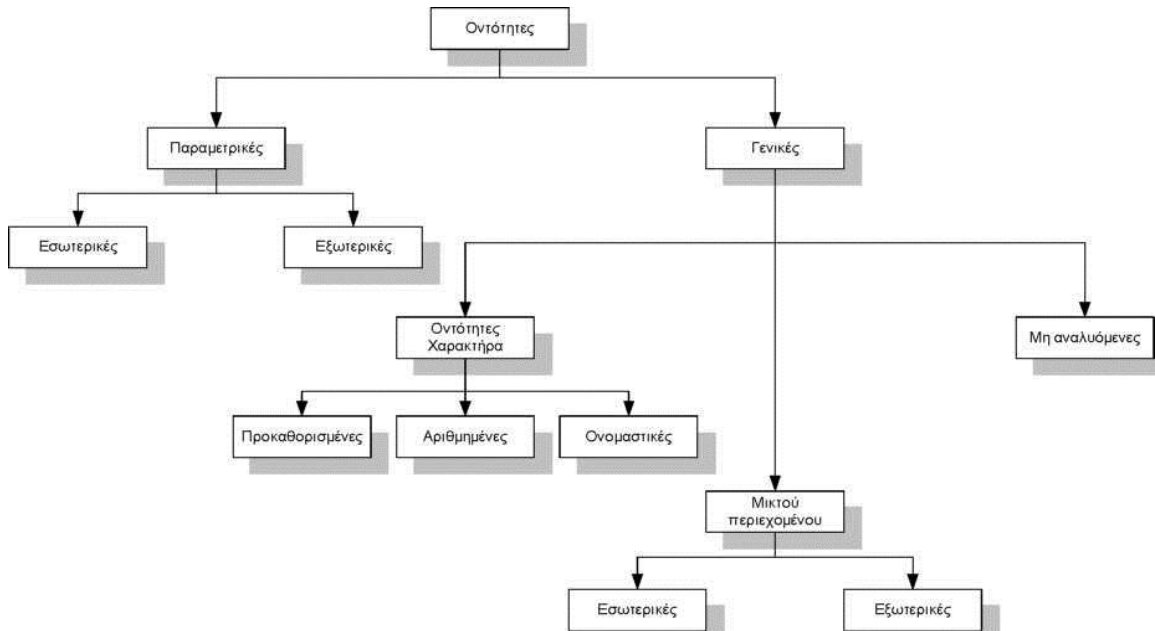
Κάπου εδώ πρέπει να γίνει λόγος για τις προκαθορισμένες ιδιότητες που παρέχει η γλώσσα XML, των οποίων η χρήση έχει καθοριστεί για ειδικούς σκοπούς. Το όνομα αυτών των ιδιοτήτων ξεκινάει με το πρόθεμα xml. Οι ιδιότητες xml:lang και xml:space αποτελούν δύο από πιο βασικές προκαθορισμένες ιδιότητες της XML, που περιγράφονται στη συνέχεια:

- xml:lang : Ταξινομεί ένα XML στοιχείο ενός XML εγγράφου με βάση τη γλώσσα αυτού του εγγράφου . Η τιμή που εκχωρείται στη συγκεκριμένη ιδιότητα είναι ο κωδικός της υφιστάμενης κάθε φορά γλώσσας ενός XML εγγράφου. Αυτό είναι χρήσιμο για τη δημιουργία υποθετικού κειμένου, του οποίου τα περιεχόμενα είναι επιλεγμένα από τον εκάστοτε XML επεξεργαστή με βάση τη γλώσσα που χρησιμοποιεί το συγκεκριμένο κείμενο.
- xml:space : Προσδιορίζει κατά πόσον τα διαδοχικά κενά πρέπει να διατηρούνται στο περιεχόμενο ενός XML στοιχείου ενός XML εγγράφου. Αν στην τρέχουσα ιδιότητα καταχωρείται η τιμή preserve, τότε ο εκάστοτε XML επεξεργαστής θα πρέπει να διατηρήσει όλα τα κενά, τις αλλαγές γραμμών ή και τις ετικέτες που τυχόν υπάρχουν στο περιεχόμενο του προαναφερόμενου στοιχείου. Αν στην παρούσα ιδιότητα αποδίδεται η τιμή default, τότε ο εκάστοτε XML επεξεργαστής μπορεί να διαχειριστεί τα κενά των στοιχείων με βάση

την προκαθορισμένη του συμπεριφορά. Αν η συγκεκριμένη ιδιότητα παραλείπεται, τότε ο εκάστοτε XML επεξεργαστής εξ ορισμού διατηρεί τα κενά στο περιεχόμενο των στοιχείων. Για το λόγο αυτό, αν είναι επιθυμητή η μείωση των διαδοχικών κενών στο περιεχόμενο των στοιχείων, τότε η ιδιότητα `xml:space` θα πρέπει να φέρει την τιμή `default`, υπό την προϋπόθεση, όμως, πως η προκαθορισμένη λειτουργία του κατάλληλου κάθε φορά XML επεξεργαστή είναι η αφαίρεση των επιπρόσθετων κενών των στοιχείων.

Το τρίτο σημαντικό κομμάτι των XML εγγράφων είναι οι οντότητες (entities). Οι οντότητες είναι μεταβλητές στις οποίες αποθηκεύεται περιεχόμενο το οποίο μπορεί να επαναχρησιμοποιηθεί σε αρκετά σημεία των εγγράφων.

Οι δύο σημαντικότεροι τύποι οντοτήτων είναι οι παραμετρικές οντότητες (που χρησιμοποιούνται μόνο στα DTD Έγγραφα) και οι γενικές οντότητες. Στην Εικόνα που ακολουθεί απεικονίζεται μία ταξινόμηση των χρησιμοποιότερων XML οντοτήτων:



Εικόνα 19: Η ταξινόμηση των οντοτήτων ενός εγγράφου της XML

Μία οντότητα της γλώσσας XML αποτελείται από ένα όνομα και μία τιμή. Όταν ένας XML αναλυτής ξεκινάει να επεξεργάζεται ένα έγγραφο, πρώτα διαβάζει μία σειρά από δηλώσεις, σε μερικές από τις οποίες ορίζονται οντότητες, όπου ένα όνομα συσχετίζεται με μία τιμή. Η τιμή αυτή μπορεί να είναι οτιδήποτε από ένα απλό χαρακτήρα ως ένα XML έγγραφο. Ενώ ο εκάστοτε αναλυτής αναλύει ένα XML έγγραφο, συναντάει αναφορές οντοτήτων, οι οποίες είναι ειδικοί δείκτες, που προέρχονται από ονόματα οντοτήτων. Σε μία τέτοια περίπτωση αντικαθιστά την εκάστοτε αναφορά οντότητας με κατάλληλο κείμενο ή σήμανση που έχει αποθηκευτεί στην αντίστοιχη οντότητα και από αυτό το σημείο και έπειτα συνεχίζει την ανάλυση των εγγράφων.

Η σύνταξη των μεν παραμετρικών οντοτήτων έχει ως εξής:

& entity;

Η σύνταξη των δε γενικών οντοτήτων είναι η επόμενη:

% entity;

Όπου `entity` το όνομα της εκάστοτε οντότητας ενός XML εγγράφου.

Όλες οι οντότητες πρέπει να δηλώνονται πριν χρησιμοποιηθούν σε ένα XML έγγραφο. Δύο αποδεκτά μέρη για να δηλωθούν είναι είτε στο εσωτερικό υποσύνολο της δήλωσης του τύπου ενός XML εγγράφου και μάλιστα στον πρόλογο αυτού του εγγράφου, όπως έχει ήδη προαναφερθεί, κάτι που

είναι ιδανικό για τοπικές οντότητες, είτε σε ένα εξωτερικό DTD Έγγραφο, κάτι που είναι καταλληλότερο για οντότητες που μοιράζονται μεταξύ εγγράφων.

Οι οντότητες που απαντώνται στα XML έγγραφα και περιέχουν ένα απλό χαρακτήρα λέγονται φυσικοί χαρακτήρες οντοτήτων. Αυτές διαιρούνται σε αρκετές ομάδες, οι σημαντικότερες από τις οποίες αναφέρονται στη συνέχεια:

- Προκαθορισμένοι χαρακτήρες οντοτήτων: Μερικοί χαρακτήρες δεν μπορούν να χρησιμοποιηθούν μέσα σε ένα κείμενο ενός XML εγγράφου, επειδή είναι δεσμευμένοι από τη γλώσσα XML, όπως, για παράδειγμα, οι γωνιακές αγκύλες (<>), που χρησιμοποιούνται, για να οριοθετήσουν τις ετικέτες των XML στοιχείων. Η XML παρέχει τους ακόλουθους σημαντικότερους προκαθορισμένους χαρακτήρες οντοτήτων που φαίνονται στον παρακάτω Πίνακα, προκειμένου να ξεπεραστεί η προαναφερθείσα δυσκολία:

Όνομα	Τιμή
amp	&
apos	'
gt	>
lt	<
quot	"

Πίνακας 23: Οι πιο βασικοί προκαθορισμένοι χαρακτήρες οντοτήτων της XML

- Αριθμητικοί χαρακτήρες οντοτήτων: Η γλώσσα XML υποστηρίζει το σύνολο χαρακτήρων Unicode, συνεπώς είναι δυνατή η χρήση οποιουδήποτε Unicode χαρακτήρα σε ένα XML έγγραφο. Το πρόβλημα που προκύπτει είναι με ποιον τρόπο θα εισαχθεί ένας χαρακτήρας από το πληκτρολόγιο που παρέχει έναν περιορισμένο αριθμό χαρακτήρων ή πώς θα αναπαρασταθεί ένας τέτοιος χαρακτήρας σε έναν απλό κειμενογράφο. Μία λύση είναι η χρήση ενός αριθμητικού χαρακτήρα οντοτήτων, που θα έχει τη μορφή #n. Ο ακέραιος n μπορεί να εκφραστεί σε δεκαδική ή δεκαεξαδική μορφή και το συγκεκριμένο σχήμα μπορεί να αναπαραστήσει τους χαρακτήρες από 0 ως 65.536 του συστήματος Unicode.
- Ονομαστικοί χαρακτήρες οντοτήτων: Το πρόβλημα που εμφανίζεται με τους αριθμητικούς χαρακτήρες οντοτήτων της γλώσσας XML είναι η δύσκολη απομνημόνευσή τους. Μία λύση είναι η χρήση μνημονικών ονομάτων. Αυτοί οι ονομαστικοί χαρακτήρες οντοτήτων εκμεταλλεύονται εύκολα στην απομνημόνευση ονόματα, που το καθένα τους αντιστοιχεί σε έναν κατάλληλο χαρακτήρα. Αντίθετα από τους προκαθορισμένους και αριθμητικούς χαρακτήρες, οι ονομαστικοί χαρακτήρες οντοτήτων πρέπει να δηλωθούν.

Το τέταρτο δομικό κομμάτι της XML είναι τα σχόλια (comments), δηλαδή σημειώσεις στο εκάστοτε XML έγγραφο, που δε μεταφράζονται από τον εκάστοτε XML αναλυτή του εγγράφου αυτού. Ο τρόπος με τον οποίο συντάσσονται τα σχόλια είναι ο κάτωθι:

```
<!-- Comment ->
```

Όπου Comment το εκάστοτε επιθυμητό σχόλιο που πρόκειται να εισαχθεί στον κώδικα του υφισταμένου σε κάθε περίπτωση XML εγγράφου.

Ένα XML σχόλιο μπορεί να περιέχει κάθε είδος κειμένου, συμπεριλαμβανομένων κενών, νέας γραμμής καθώς και σήμανσης. Παρόλα αυτά, αφού οι δύο παύλες χρησιμοποιούνται, για να δείχνουν στον εκάστοτε XML αναλυτή ενός XML εγγράφου πότε ένα σχόλιο αρχίζει και πότε τελειώνει και δεν μπορούν να χρησιμοποιηθούν μέσα στο σχόλιο. Αυτό σημαίνει ότι, αν είναι επιθυμητή, για παράδειγμα, η δημιουργία μίας νοητής γραμμής μέσα σε ένα XML έγγραφο, τότε δε γίνεται να εφαρμοστεί το σύμβολο της παύλας (-), αλλά πρέπει να εκμεταλλευτεί άλλο κατάλληλο σύμβολο, όπως είναι, για παράδειγμα, το ίσον (=) ή η υπογράμμιση (_). Επιπλέον, τα σχόλια μπορούν να μπουν σε οποιοδήποτε σημείο ενός XML εγγράφου, εκτός πριν από την XML δήλωση, ή μέσα σε ετικέτες αυτού του

εγγράφου, όπου ο επεξεργαστής θα τα αγνοήσει εντελώς. Τέλος κάτι έξυπνο: από τη στιγμή που τα σχόλια δύναται να περιέχουν σήμανση, είναι εφικτό να χρησιμοποιηθούν, ώστε να σβήσουν μέρη ενός XML εγγράφου. Κάτι τέτοιο κρίνεται σκόπιμο, εφόσον επιθυμείται να αφαιρεθεί προσωρινά μία ενότητα του παραπάνω εγγράφου, αποθηκεύοντας τη στο φάκελο που περιέχει το συγκεκριμένο έγγραφο για μετέπειτα χρήση.

Το πέμπτο συστατικό ενός XML εγγράφου είναι τα τμήματα CDATA (Character Data) τα οποία συνιστούν ένα εναλλακτικό τρόπο, για να τυπωθούν δεσμευμένοι χαρακτήρες σήμανσης (πχ <, >, &), όταν γίνεται συχνή εκμετάλλευση αυτών, οπότε και η χρήση των αντίστοιχων οντοτήτων των προαναφερόμενων χαρακτήρων (πχ <, >, &) καθίσταται κουραστική.

Η σύνταξη των CDATA δείχνεται πιο κάτω:

```
<![CDATA[Text]]>
```

Όπου Text το εκάστοτε κείμενο που είναι επιθυμητό να εμφανίζεται σε ένα XML έγγραφο, το οποίο περιέχει δεσμευμένους χαρακτήρες σήμανσης και τους οποίους οφείλει να αγνοήσει ο XML επεξεργαστής αυτού του εγγράφου, ώστε οι τρέχοντες χαρακτήρες να παρουσιαστούν στην εκάστοτε Ιστοσελίδα.

Εδώ, να σημειωθεί ότι μέσα στο Text η μόνη η αλληλουχία χαρακτήρων που δε μπορεί ο XML επεξεργαστής ενός XML εγγράφου να αγνοήσει είναι ο οριοθέτης τέλους ενός τμήματος CDATA, με άλλα λόγια η]]> . Αν απαιτείται οπωσδήποτε να τυπωθεί η συγκεκριμένη αλληλουχία σε ένα XML έγγραφο, τότε χρησιμοποιείται υποχρεωτικά μία προκαθορισμένη οντότητα και η παραπάνω ακολουθία τυπώνεται με τη μορφή]]>.

Ο έκτος και τελευταίος δομικός λίθος ενός XML εγγράφου είναι οι οδηγίες επεξεργασίας (editing instructions). Γενικά σε ένα XML έγγραφο οι πληροφορίες παρουσίασης πρέπει να διαχωρίζονται από αυτό το έγγραφο, όπου κάτι τέτοιο μπορεί να λάβει χώρα. Υπάρχουν και στιγμές που δεν υπάρχει άλλη επιλογή, όπως είναι, για παράδειγμα, όταν χρειάζεται να αποθηκευτούν αριθμημένες σελίδες σε ένα έγγραφο, ώστε να διευκολυνθεί η παραγωγή ενός ευρετηρίου. Αυτή η πληροφορία εφαρμόζεται μόνο σε ένα συγκεκριμένο XML επεξεργαστή και μπορεί να είναι παραπλανητική για τους άλλους. Η οδηγία για μία τέτοια πληροφορία είναι μία οδηγία επεξεργασίας. Είναι ένα δοχείο με δεδομένα που στοχεύουν στον προαναφερόμενο μοναδικό XML επεξεργαστή. Οι οδηγίες επεξεργασίας περιέχουν δύο κομμάτια πληροφορίας: το στόχο και τα δεδομένα. Ο συσχετιζόμενος σε κάθε περίπτωση XML αναλυτής με τον προηγούμενο XML επεξεργαστή περνάει τις οδηγίες επεξεργασίας στο επόμενο επίπεδο επεξεργασίας. Αν ο χειριστής των οδηγιών επεξεργασίας αναγνωρίσει το στόχο, μπορεί να διαλέξει να χρησιμοποιήσει τα δεδομένα; αλλιώς τα δεδομένα θα απορριφθούν.

Μετέπειτα αντικατοπτρίζεται η σύνταξη των οδηγιών επεξεργασίας ενός XML εγγράφου:

```
<?target data ?>
```

Όπου:

- target: Ο στόχος των οδηγιών επεξεργασίας ενός XML εγγράφου
- data: Τα δεδομένα του target.

Αν δεν υπάρχουν δεδομένα στο στόχο των οδηγιών επεξεργασίας ενός XML εγγράφου, σε αυτή την περίπτωση ο εν λόγω στόχος από μόνος του μπορεί να θεωρηθεί ως δεδομένο. Όπως μπορεί να γίνει αντιληπτό οι οδηγίες επεξεργασίας μοιάζουν με την XML δήλωση ενός εγγράφου. Ο στόχος είναι μία λέξη-κλειδί, η οποία, ωστόσο, δε σημαίνει απαραίτητα κάτι, και την οποία χρησιμοποιεί ο εκάστοτε XML επεξεργαστής, με σκοπό προσδιορίσει, αν τα δεδομένα προορίζονται για αυτόν ή όχι.

Επιπλέον, οι XML οδηγίες επεξεργασίας μπορεί να περιέχουν οτιδήποτε δεδομένα, εκτός από τον συνδυασμό ?> , ο οποίος μεταφράζεται ως ο τερματικός οριοθέτης

Ακόμα, ως παρατηρηθεί πως παραπάνω από ένα προγράμματα μπορούν να χρησιμοποιήσουν μία οδηγία επεξεργασίας καθώς, επίσης, και ότι ένα πρόγραμμα μόνο του μπορεί να δεχθεί πολλές XML οδηγίες επεξεργασίας.

Τέλος, ως αναφερθεί ότι η XML δήλωση ενός XML εγγράφου μπορεί να θεωρηθεί σε μία οδηγία επεξεργασίας για όλους τους XML επεξεργαστές, που λαμβάνουν γενικές πληροφορίες για αυτό το έγγραφο

[47] [48] [49]

2.13.8 Οι Συντακτικοί Κανόνες της XML

Τα XML έγγραφα προκειμένου να είναι καλά ορισμένα θα πρέπει να προσαρμόζονται στους ακόλουθους ακριβείς κανόνες σύνταξης:

- Όλα τα XML στοιχεία ενός XML εγγράφου οφείλουν να έχουν μία ετικέτα τέλους: Ακόμα και τα κενά στοιχεία στη γλώσσα XML θα πρέπει να τερματίζονται κανονικά με τη χρήση ετικέτας τέλους ή συνεπτυγμένα με την προσθήκη του χαρακτήρα / , πριν από τον οριοθέτη > της ετικέτας δήλωσής τους .
- Οι XML ετικέτες ενός XML εγγράφου πρέπει να είναι κατάλληλα τοποθετημένες: Τα επικαλυπτόμενα στοιχεία δεν επιτρέπονται. Ένα στοιχείο πρέπει να έχει μία ετικέτα τέλους πριν από την ετικέτα έναρξης του επόμενου στοιχείου και να είναι κατάλληλα διευθετημένα το ένα μέσα στο άλλο.
- Οι XML ετικέτες ενός XML εγγράφου διακρίνουν τα κεφαλαία γράμματα από τα μικρά.
- Τα XML έγγραφα κρίνεται αναγκαίο να έχουν οπωσδήποτε ένα στοιχείο «ρίζα»: Όλα τα XML έγγραφα πρέπει να περιέχουν ένα ενιαίο, μοναδικό ζευγάρι από ετικέτες, για να καθορίσουν το στοιχείο «ρίζα». Επίσης, όλα τα άλλα στοιχεία που απαντώνται σε ένα XML εγγράφου, πλην από το στοιχείο το στοιχείο «ρίζα», πρέπει να τοποθετηθούν ως ένθετα μέσα στο στοιχείο «ρίζα». Ακόμα, όλα τα στοιχεία ενός XML εγγράφου μπορούν να έχουν (ένθετα) στοιχεία «παιδιά», που πρέπει να είναι σε ζευγάρια και σωστά τοποθετημένα μέσα στο «γονικό» τους στοιχείο.
- Οι τιμές των XML ιδιοτήτων των XML στοιχείων ενός XML εγγράφου θεωρείται επιτακτικό πάντα να βρίσκονται σε εισαγωγικά: Ένα στοιχείο μπορεί προαιρετικά να περιέχει μία ή περισσότερες ιδιότητες στην ετικέτα δήλωσή του, οι τιμές των οποίων, όμως, είναι υποχρεωτικό να περικλείονται εντός διπλών εισαγωγικών ("").
- Στα XML έγγραφα θα πρέπει να γίνεται σωστή χρήση των ειδικών χαρακτήρων: Τα δεδομένα μίας XML Σελίδας δε θα πρέπει να περιέχουν δεσμευμένους χαρακτήρες σήμανσης. Τέτοιοι χαρακτήρες θα πρέπει να αντικαθίστανται από τη χρήση κατάλληλων οντοτήτων η αντίστοιχων τμημάτων CDATA.

[47] [48]

2.13.9 Οι Χώροι Ονομασίας της XML

Μία από τις δυσκολίες που ανακύπτουν κατά την εργασία με XML έγγραφα, που ανακτώνται από πολλαπλές πηγές, είναι πως η έννοια των ετικετών της κάθε εφαρμογής μπορεί να γίνει πολλές φορές ασαφής. Το πρόβλημα αυτό επιλύουν οι XML Χώροι Ονομασίας (XML Namespaces), οι οποίοι επιτρέπουν στους εκάστοτε χρήστες να εξαλείψουν την ασάφεια της έννοιας των XML στοιχείων και ιδιοτήτων ενός XML εγγράφου, προσδιορίζοντας ονόματα στοιχείων και ιδιοτήτων, κατόπιν συσχέτισμού αυτών με χώρους ονομασίας που συνδέονται με URI αναφορές.

Σε γενικές γραμμές ένας χώρος ονομασίας στη γλώσσα XML είναι μία συλλογή από ονόματα στοιχείων και χαρακτηριστικών. Ένας χώρος ονομασίας αντιστοιχεί σε ένα μοναδικό όνομα, που είναι ένα URI. Κατά συνέπεια ένα όνομα στοιχείου ή ιδιότητας σε ένα XML χώρο ονομασίας μπορεί να ταυτιστεί μοναδικά από ένα όνομα αποτελούμενο από δύο μέρη: το όνομα του εκάστοτε χώρου ονομασίας και του εκάστοτε τοπικού του ονόματος (local name).

Οι χώροι ονομασίας στη γλώσσα XML δηλώνονται με τη χρήση της ιδιότητας xmlns, μέσω της οποίας συσχετίζεται ένα πρόθεμα με ένα χώρο ονομασίας. Η δήλωση είναι εντός εμβέλειας για το κάθε στοιχείο που περιέχει την ιδιότητα και όλους τους απογόνους του. Πιο συγκεκριμένα:

- Αν μία δήλωση ενός XML χώρου ονομασίας περιέχει ένα πρόθεμα, τότε η αναφορά στα ονόματα των στοιχείων και των ιδιοτήτων που ανήκουν στο χώρο αυτό γίνεται με τη χρήση του προαναφερόμενου προθέματος.
- Αν μία δήλωση ενός XML χώρου ονομασίας δεν περιέχει πρόθεμα, τότε ο εν λόγω χώρος ονομασίας είναι ο προκαθορισμένος χώρος ονομασίας και δεν απαιτείται η χρήση προθέματος, προκειμένου να γίνει αναφορά στα στοιχεία που περιέχονται στο χώρο αυτό.

[47] [49]

2.13.10 Σύγκριση της XML

Η γλώσσα XML συμπληρώνει και δεν αντικαθιστά τη γλώσσα HTML. Ενώ η HTML χρησιμοποιείται για τη διατύπωση και την εμφάνιση των δεδομένων σε Ιστοσελίδες, η XML αναπαριστά τη συναφή έννοια των δεδομένων. Στην HTML οι ετικέτες είναι προκαθορισμένες, σε αντίθεση με την XML, που παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τις ετικέτες και τις δομημένες μεταξύ τους σχέσεις. Επομένως, ο εκάστοτε χρήστης με τη βοήθεια της XML μπορεί να δημιουργήσει μία νέα γλώσσα σήμανσης καθορίζοντας ένα νέο σύνολο από ετικέτες ή να χρησιμοποιήσει ένα σύνολο από ετικέτες το οποίο είναι καθορισμένο από κάποιον άλλο. Αυτό έχει ως αποτέλεσμα να υπάρχει απεριόριστος αριθμός γλωσσών σήμανσης που προέρχονται από την XML και για το λόγο αυτό η XML είναι μία «μετα-γλώσσα». Τα πλεονεκτήματα της XML έναντι της HTML είναι τα εξής:

- Παρέχει πιο ικανοποιητικές αναζητήσεις σε έγγραφα του Παγκόσμιου Ιστού.
- Διευκολύνει την ανάπτυξη ευέλικτων διαδικτυακών εφαρμογών.
- Ενσωματώνει στον κώδικα μίας Ιστοσελίδας δεδομένα από ανόμιες πηγές πληροφορίας και από πολλαπλές εφαρμογές.
- Υπολογίζει και διαχειρίζεται διάφορα δεδομένα των εγγράφων μίας Ιστοσελίδας σε τοπικό επίπεδο.
- Παρέχει πολλαπλούς τρόπους εμφάνισης των δεδομένων σε μία Ιστοσελίδα.

[47] [48] [49]

2.14 Η Επεκτάσιμη Μορφή της HTML, XHTML

2.14.1 Εισαγωγή στην XHTML

Σήμερα πολλές Ιστοσελίδες περιέχουν «κακό» κώδικα της γλώσσας HTML, δηλαδή κώδικα που δεν είναι αυστηρά ορισμένος, αλλά παρόλα αυτά τρέχει κανονικά στα περισσότερα προγράμματα περιήγησης στο Διαδίκτυο. Ωστόσο, στη σύγχρονη εποχή, επίσης, συναντάει κανείς στο Διαδίκτυο περιηγητές διαφόρων τεχνολογιών. Κάποιοι από αυτούς τρέχουν σε επιτραπέζιους υπολογιστές (desktop computers), κάποιοι άλλοι σε φορητούς υπολογιστές (laptops) και κάποιοι άλλοι σε μικρότερες συσκευές, όπως είναι, για παράδειγμα, οι υπολογιστές ταμπλέτα (tablets) ή τα «έξυπνα» κινητά τηλέφωνα (smartphones). Αυτές οι μικρότερες συσκευές συχνά στερούνται τον πόρων και της ισχύς, για να μεταφράσουν την προαναφερθείσα «κακή» σήμανση των Σελίδων. Για το λόγο αυτό επεκτάθηκε και αναβαθμίστηκε η γλώσσα HTML με τη βοήθεια της γλώσσας XML, στην οποία τα έγγραφα ακολουθούν αυστηρά ορθή, και προέκυψε έτσι η γλώσσα XHTML (Extensible HyperText Markup Language) (απόδοση στα Ελληνικά: Επεκτάσιμη Γλώσσα Σήμανσης Υπερκειμένου).

Η XHTML συνιστά μία γλώσσα σήμανσης του περιεχομένου των Ιστοσελίδων περίπου ταυτόσημη με την HTML, αλλά με αυστηρότερους κανόνες σύνταξης. Στην ουσία η XHTML είναι ένας επανασχεδιασμός της γλώσσας HTML ως μία XML εφαρμογή, η οποία υποστηρίζεται από την πλειονότητα των σύγχρονων φυλλομετρητών.

[50]

2.14.2 Μετατροπή της HTML σε XHTML

Για να μετατραπεί ένα HTML έγγραφο σε XHTML εκτελούνται οι ακόλουθες ενέργειες:

- Προστίθεται η XHTML ετικέτα `<!DOCTYPE xhtml>` στην αρχή κάθε εγγράφου.
- Τοποθετείται το χαρακτηριστικό γνώρισμα `xmlns` στην ετικέτα `<html>...</html>` κάθε εγγράφου.
- Γράφονται όλα τα ονόματα των στοιχείων ενός εγγράφου με πεζά γράμματα.
- Κλείνουν όλα τα κενά στοιχεία ενός εγγράφου.
- Καταχωρούνται, επίσης, με μικρά γράμματα όλα τα ονόματα των ιδιοτήτων των στοιχείων ενός εγγράφου.
- Περικλείονται εντός διπλών εισαγωγικών (") οι τιμές όλων των χαρακτηριστικών γνωρισμάτων των στοιχείων ενός εγγράφου.

[50] [51]

2.14.3 Σύγκριση της XHTML με την HTML

Η γλώσσα XHTML ως αναδιατύπωση ουσιαστικά της γλώσσας HTML εμφανίζει πολλές ομοιότητες με αυτή. Εφόσον, όμως, συνδυάζει και πολλές δυνατότητες της XML διαφοροποιείται τελικά ως ένα βαθμό, από την HTML τόσο ως προς τους κανόνες σύνταξης όσο και ως προς τους κανόνες «συμπεριφοράς».

Οι κρίσιμες διαφορές της γλώσσας XHTML από τη γλώσσα HTML από τη μεριά των κανόνων σύνταξης κατηγοριοποιούνται όπως παρακάτω:

- Σχετικά με τη δομή του εκάστοτε εγγράφου (document structure):
 - ο Η ετικέτα `<!DOCTYPE xhtml>`, η οποία δηλώνει τον XHTML τύπο ενός εγγράφου είναι υποχρεωτική, ενώ η αντίστοιχη ετικέτα `<!DOCTYPE html>`, που ορίζει τον HTML τύπο ενός εγγράφου, απαντάται μόνο στην τελευταία πέμπτη έκδοση της γλώσσας HTML.
 - ο Το χαρακτηριστικό γνώρισμα `xmlns` της ετικέτας `<html>...</html>` της γλώσσας XHTML είναι επιβεβλημένο, ενώ κάτι αντίστοιχο δεν ισχύει στη γλώσσα HTML.
 - ο Οι ετικέτες `<html>...</html>`, `<head>...</head>`, `<title>...</title>` και `<body>...</body>` στη γλώσσα XHTML είναι πάντα αναγκαίες, ενώ στη γλώσσα HTML κάποιες από αυτές (ή και όλες) μερικές φορές μπορούν να παρακαμφθούν (παρόλο που κάτι τέτοιο ειδικά στις νεότερες εκδόσεις της γλώσσας δε συνιστάται).
- Αναφορικά με τα στοιχεία (elements) του υφιστάμενου σε κάθε περίπτωση εγγράφου:
 - ο Όλα τα στοιχεία της γλώσσας XHTML πρέπει να είναι κανονικά «φωλιασμένα», σε αντιδιαστολή με κάποια στοιχεία της γλώσσας HTML, που δεν είναι ανάγκη να είναι κανονικά «φωλιασμένα».
 - ο Κάθε ετικέτα ανοίγματος στη γλώσσα XHTML έχει την αντίστοιχη ετικέτα κλεισίματος, αντίθετα με τη γλώσσα HTML, όπου μερικές ετικέτες δεν είναι υποχρεωτικό να κλείνουν.
 - ο Τα κενά στοιχεία της γλώσσας XHTML πρέπει κάθε φορά να κλείνουν, σε αντιδιαστολή με τα κενά στοιχεία της γλώσσας HTML, που δεν κλείνουν πάντοτε.
 - ο Τα ονόματα των στοιχείων της γλώσσας XHTML πρέπει να γράφονται με πεζά γράμματα σε αντίθεση με τα στοιχεία της γλώσσας HTML, τα οποία μπορούν να είναι γραμμένα είτε με μικρά είτε με κεφαλαία.
 - ο Τα XHTML έγγραφα έχουν ένα και μόνο στοιχείο «ρίζα», ωστόσο στη γλώσσα HTML κάτι τέτοιο δεν είναι επιβεβλημένο.
- Όσον αφορά τα χαρακτηριστικά (γνωρίσματα)\ιδιότητες (attributes) των στοιχείων του υπάρχοντος σε κάθε περίπτωση εγγράφου:
 - ο Τα ονόματα των χαρακτηριστικών γνωρισμάτων της γλώσσας XHTML οφείλουν να παρουσιάζονται με πεζούς χαρακτήρες, αντίθετα με τα ονόματα των χαρακτηριστικών γνωρισμάτων της γλώσσας HTML, όπου δεν είναι αναγκαία η διάκριση μικρών ή μεγάλων χαρακτήρων.
 - ο Οι τιμές των χαρακτηριστικών γνωρισμάτων της γλώσσας XHTML είναι υποχρεωτικό να περικλείονται πάντα εντός διπλών εισαγωγικών ("), ενώ κάτι τέτοιο μπορεί να παραληφθεί στη γλώσσα HTML.
 - ο Στη γλώσσα XHTML απαγορεύεται το «μίκρεμα» των χαρακτηριστικών γνωρισμάτων, ενώ στη γλώσσα HTML είναι επιτρεπτό.

Οι σημαντικές, τώρα, διαφορές της γλώσσας XHTML από τη γλώσσα HTML από τη σκοπιά των κανόνων «συμπεριφοράς» μπορούν να διακριθούν ως εξής:

- Από την πλευρά του εντοπισμού σφαλμάτων (debugging) στον κώδικα του εκάστοτε εγγράφου: Ένα μοιραίο λάθος κατά την ανάλυση των λαθών ενός XHTML εγγράφου (πχ λανθασμένη δομή ετικετών) προκαλεί τη διακοπή φόρτωσης της εκάστοτε Ιστοσελίδας, ενώ κάτι παρόμοιο στη γλώσσα HTML μπορεί να προσπεραστεί.
- Από την οπτική γωνία της χρήσης κανόνων ονομασίας του υπάρχοντος σε κάθε φάση εγγράφου: Η χρησιμοποίηση ενός σετ κανόνων κατά την ονομασία μεταβλητών στη γλώσσα XHTML είναι επιτακτική, ενώ στη γλώσσα HTML δεν ωφελεί.

- Προκειμένου για την εκμετάλλευση των CSS στο υφιστάμενο σε κάθε περίπτωση έγγραφο: Εξαιτίας του διαχωρισμού μικρών και μεγάλων γραμμάτων που λαμβάνει χώρα στη γλώσσα XHTML, όλοι οι CSS επιλογείς υιοθετούν την παραπάνω διάκριση, κάτι που φυσικά στη γλώσσα HTML δε συμβαίνει.
- Σε σχέση με την ενσωμάτωση της γλώσσας JavaScript στο υπό μελέτη σε κάθε περίπτωση έγγραφο: Η χρήση της JavaScript διαφοροποιείται στη γλώσσα XHTML σε σύγκριση με τη γλώσσα HTML όσον αφορά τη διάκριση πεζών κεφαλαίων χαρακτήρων στα ονόματα μερικών συναρτήσεων και τη λήψη περαιτέρω προφυλάξεων για τον περιορισμό της επεξεργασίας σε καλοσχηματισμένο περιεχόμενο.

[50] [51]

2.15 Η Τεχνική Προγραμματισμού Διαδικτύου, AJAX

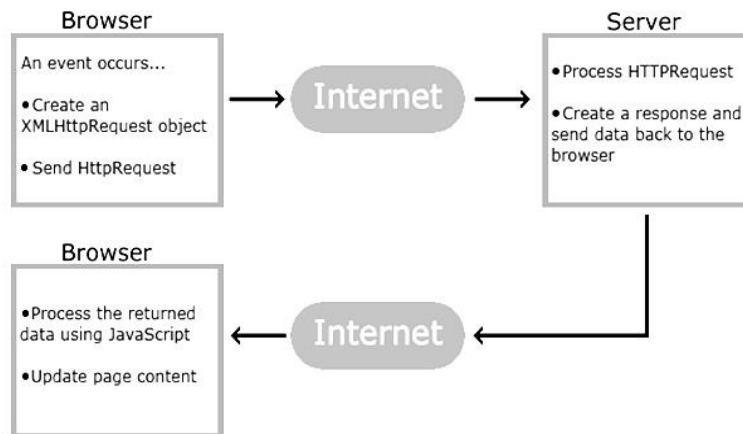
2.15.1 Εισαγωγή στην AJAX

Γενικά μία Ιστοσελίδα είναι ένα σύνολο HTML οδηγιών που στέλνει ένας εξυπηρετητής Ιστού στον εκάστοτε πελάτη του, μόλις αυτός αιτείται να δει το περιεχόμενο μίας Σελίδας. Η αποστολή των HTML οδηγιών γίνεται μία φορά και αφού αυτή ολοκληρωθεί, η επικοινωνία εξυπηρετητή-πελάτη κλείνει. Έτσι, κάθε φορά που ο κάτοχος-δημιουργός-διαχειριστής ενός Ιστοτόπου κάνει αλλαγές στο περιεχόμενό του, οι αλλαγές αυτές δεν είναι άμεσα ορατές από τον εκάστοτε επισκέπτη, εκτός και αν ο τελευταίος ανανεώσει την Ιστοσελίδα, πατώντας το πλήκτρο “F5” του πληκτρολογίου, οπότε ξανανοίγει η επικοινωνία αυτού του περιηγητή με τον διακομιστή παραλαμβάνοντας εκ νέου το ανανεωμένο περιεχόμενο.

Η χρήση της τεχνικής AJAX επιτρέπει σε μία Ιστοσελίδα να ανανεώνεται ασύγχρονα (asynchronously), ανταλλάσσοντας στο παρασκήνιο μικρού όγκου δεδομένα με τον υφιστάμενο σε κάθε περίπτωση εξυπηρετητή Ιστού, επιτρέποντας με τον τρόπο αυτό την ανανέωση συγκεκριμένων μερών της Σελίδας, δίχως να ανανεώνεται ολόκληρη η Σελίδα. Η τεχνική AJAX χρησιμοποιείται πολύ συχνά σε Σελίδες που περιέχουν φόρμες εγγραφής. Είναι γνωστό ότι στις φόρμες αυτές πρέπει να επιλεγεί ένα επιθυμητό όνομα χρήστη. Αν κατά τη συμπλήρωση μίας τέτοιας φόρμας εμφανιστεί ένα μήνυμα προειδοποίησης παρόμοιο με αυτό: «Το όνομα χρήστη που πληκτρολογήθηκε υπάρχει ήδη.», χωρίς να ξαναφορτωθεί η εν λόγω Σελίδα, τότε η συγκεκριμένη Σελίδα χρησιμοποιεί την τεχνική AJAX. Επίσης, το Facebook χρησιμοποιεί ευρέως την τεχνική αυτή σε διάφορα τμήματα της Σελίδας του, όπως, για παράδειγμα, προκειμένου να εμφανίσει τις νέες αναρτήσεις φίλων, οι οποίες συνεχώς «σκρολάρουν» στην λίστα επάνω δεξιά ή για να εμφανίσει σχόλια κάτω από μία ανάρτηση.

Ας οριστεί, όμως, στο σημείο τούτο, η τεχνική AJAX. Η AJAX, λοιπόν, δεν είναι γλώσσα σεναρίου όπως η JavaScript, αλλά ούτε και γλώσσα χαρακτηρισμού κειμένου, όπως η HTML. Δε θεωρείται καν γλώσσα. AJAX σημαίνει Asynchronous Javascript And XML και πρόκειται για μία τεχνική με την οποία είναι εφικτή η κατασκευή πολύ πιο γρήγορων και δυναμικών Σελίδων περιορίζοντας τον όγκο δεδομένων που ανταλλάσσει ο εκάστοτε διακομιστής Ιστού με τον πλοηγό του επισκέπτη. Η ιδιαίτερη αυτή τεχνική επιτρέπει την ανανέωση περιεχομένων μίας Ιστοσελίδας, χωρίς της ανανέωση του συνόλου της.

Η υλοποίηση της τεχνικής AJAX επιτυγχάνεται με το συνδυασμό του αντικειμένου XMLHttpRequest (πραγματοποιεί την ασύγχρονη επικοινωνία με τον εκάστοτε εξυπηρετητή Ιστού), της γλώσσας Javascript (αλληλεπιδρά με τα δεδομένα και τα εμφανίζει), της γλώσσας CSS (μορφοποιεί τα προς εμφάνιση δεδομένα) και της γλώσσας XML (συχνά χρησιμοποιείται για την μεταφορά δεδομένων), όπως φαίνεται και στην παρακάτω Εικόνα:



Εικόνα 20: Η λειτουργία της AJAX

[45] [52]

2.15.2 Η Δημιουργία του Αντικειμένου XMLHttpRequest της AJAX

Το αντικείμενο XMLHttpRequest χρησιμοποιείται για την ασύγχρονη μεταφορά δεδομένων από έναν εξυπηρετητή Ιστού στον εκάστοτε περιηγητή, μεταφορά που πραγματοποιείται στο παρασκήνιο χωρίς ο επισκέπτης μίας Σελίδας να ανανεώνει το περιεχόμενό της.

Το αντικείμενο XMLHttpRequest υποστηρίζεται από όλους τους περιηγητές, εκτός από την πέμπτη και την έκτη έκδοση του Internet Explorer (IE), για τους οποίους διατίθεται το αντικείμενο ActiveXObject. Για αυτό τον λόγο εκμεταλλεύεται μία if συνθήκη για τη δημιουργία αυτού του αντικειμένου και για τις δύο περιπτώσεις:

- Για τους περιηγητές IE 5 και IE 6 χρησιμοποιείται η παρακάτω σύνταξη:

```
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
```

- Για όλους τους υπόλοιπους πλοηγούς γίνεται χρήση της επόμενης σύνταξης:

```
xmlhttp=new XMLHttpRequest();
```

Έτσι, η if συνθήκη για τη δημιουργία του αντικειμένου XMLHttpRequest της AJAX σε κάθε περίπτωση έχει ως εξής:

```
var xmlhttp;
if (window.XMLHttpRequest)
  // Code for IE7+, Firefox, Chrome, Opera, Safari:
  xmlhttp=new XMLHttpRequest();
}
else
  // Code for IE 5, IE 6:
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
```

[52] [53]

2.15.3 Η Διαδικασία Αποστολής Αιτήματος σε έναν Εξυπηρετητή Ιστού στην AJAX

Για να πραγματοποιηθεί η μεταφορά δεδομένων από έναν εξυπηρετητή Ιστού στον εκάστοτε περιηγητή μέσω της τεχνικής AJAX, θα πρέπει αρχικά να υπάρχει ένα αίτημα από αυτόν το φυλλομετρητή. Για να σταλεί ένα τέτοιο αίτημα χρησιμοποιούνται οι μέθοδοι `open()` και `send()` του αντικειμένου `XMLHttpRequest` της AJAX. Η μέθοδος `open()` ανοίγει την επικοινωνία του προαναφερόμενου περιηγητή με τον προαναφερόμενο διακομιστή ορίζοντας τον τύπο της αποστολής του προηγούμενου αιτήματος καθώς και τη διεύθυνση στον Ιστό του αρχείου που θα επεξεργαστεί το αίτημα και είναι αποθηκευμένο στο συγκεκριμένο διακομιστή, ενώ η μέθοδος `send()` εκτελεί την αποστολή του πιο πάνω αιτήματος στο συγκεκριμένο εξυπηρετητή.

Η μέθοδος `open()` συντάσσεται ως εξής:

```
open(method,url,asynch);
```

Όπου:

- `method`: Η μέθοδος αποστολής ενός αιτήματος ενός πελάτη σε έναν διακομιστή Ιστού, δηλαδή GET ή POST. Η GET είναι απλούστερη και πιο γρήγορη από την POST. Παρόλα αυτά η POST πρέπει να χρησιμοποιείται στις εξής περιπτώσεις:
 - Όταν διαβάζονται δεδομένα από μία Βάση Δεδομένων.
 - Εφόσον ανταλλάσσονται μεγάλο όγκο δεδομένα.
 - Αν ο χρήστης πρέπει να πληκτρολογήσει κάτι το οποίο μπορεί να περιέχει μη αποδεκτούς χαρακτήρες.
- `url`: Η διεύθυνση στον Ιστό του αρχείου του εκάστοτε εξυπηρετητή που θα επεξεργαστεί το αίτημα ενός πελάτη και θα επιστρέψει ένα αποτέλεσμα. Το αρχείο αυτό μπορεί να είναι οποιoδήποτε είδους αρχείου (πχ .txt ή .xml) ή ένα αρχείο σεναρίου εξυπηρετητή (πχ .php ή .asp).
- `asynch`: Η παράμετρος αυτή είναι `true`, όταν απαιτείται η αποστολή ενός αιτήματος ενός πελάτη σε έναν εξυπηρετητή Ιστού να γίνει ασύγχρονα και `false`, άμα πρέπει η προαναφερόμενη αποστολή να γίνει μη-ασύγχρονα. Ουσιαστικά, για να γίνει εκμετάλλευση της ασύγχρονης επικοινωνίας, που είναι και η κύρια ιδιότητα της τεχνικής AJAX, η εν λόγω παράμετρος είναι `true`. Ο ασύγχρονος τρόπος επικοινωνίας σημαίνει πως ο εκάστοτε φυλλομετρητής δεν περιμένει να εδραιωθεί και να ολοκληρωθεί η επικοινωνία με τον εκάστοτε εξυπηρετητή, χωρίς να κάνει τίποτα άλλο, αλλά παράλληλα εκτελεί άλλα σενάρια της εκάστοτε τρέχουσας Ιστοσελίδας, με αποτέλεσμα να φορτώνουν όλα τα άλλα μέρη της. Η τιμή `false` δεν προτείνεται ως καλή λύση, μίας και ο περιηγητής μέχρις ότου να ολοκληρώσει επιτυχώς την σύνδεση του με το φυλλομετρητή, δε θα φορτώσει κανένα άλλο σενάριο της εκάστοτε τρέχουσας Σελίδας.

Η μέθοδος `send()` συντάσσεται ως εξής:

```
send(string);
```

Όπου `string` η κατάλληλη σε κάθε περίπτωση συμβολοσειρά, η οποία περιλαμβάνει τις παραμέτρους μαζί με τις τιμές τους χωρισμένες με το σύμβολο `&`, εφόσον πρόκειται για αποστολή μίας αίτησης ενός πελάτη σε έναν εξυπηρετητή Ιστού με χρήση της μεθόδου `post`.

Στη συνέχεια παρατίθενται μερικά παραδείγματα για την καλύτερη κατανόηση των παραπάνω μεθόδων `open()` και `send()`:

- Αποστολή δεδομένων από έναν πελάτη σε έναν εξυπηρετητή Ιστού με τη μέθοδο GET:

```
xmlhttp.open("GET","demo_get2.asp?fname=Henry&lname=Ford",true);  
xmlhttp.send();
```

- Αποστολή δεδομένων από έναν πελάτη σε έναν εξυπηρετητή Ιστού με τη μέθοδο POST:

```
xmlhttp.open("POST","demo_post.asp",true);
xmlhttp.send();
```

- Αποστολή δεδομένων από έναν πελάτη σε έναν εξυπηρετητή του Ιστού σε HTML μορφή φόρμας:

```
xmlhttp.open("POST","ajax_test.asp",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send("fname=Henry&name=Ford");
```

[52] [54]

2.15.4 Οι Ιδιότητες `responseText` και `responseXML` της AJAX

Οι ιδιότητες `responseText` και `responseXML` της AJAX είναι ιδιότητες του αντικειμένου `XMLHttpRequest`. Οι ιδιότητες αυτές περιέχουν τα δεδομένα που στέλνει ο εκάστοτε διακομιστής Ιστού, ύστερα από αποστολή αιτήματος του εκάστοτε πελάτη. Όταν τα δεδομένα αυτά δεν είναι σε μορφή XML, τότε χρησιμοποιείται η ιδιότητα `responseText` η οποία επιστρέφει τα δεδομένα ως μία συμβολοσειρά.

Η σύνταξη της ιδιότητας `responseText` του αντικειμένου `XMLHttpRequest` φαίνεται στο παρακάτω παράδειγμα:

```
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

Όπου `myDiv` είναι το `id` του `div` του εκάστοτε εγγράφου, όπου θα εμφανιστούν τα δεδομένα που στέλνει ο εκάστοτε διακομιστής Ιστού στον εκάστοτε πελάτη που τα αιτήθηκε.

Εφόσον, τώρα, τα δεδομένα που στέλνει ο εκάστοτε εξυπηρετητής Ιστού στον πελάτη είναι σε μορφή XML, τότε διατίθεται η ιδιότητα `responseXML` του αντικειμένου `XMLHttpRequest`, της οποίας η σύνταξη αποσαφηνίζεται στο παράδειγμα που έπεται:

```
xmlDoc=xmlhttp.responseXML;
txt="";
x=xmlDoc.getElementsByTagName("ARTIST");
for (i=0;i<x.length;i++)
{
    txt=txt + x[i].childNodes[0].nodeValue + "<br />";
}
document.getElementById("myDiv").innerHTML=txt;
```

Όπου `ARTIST` είναι το όνομα της XML ετικέτας του εκάστοτε εγγράφου, της οποίας το περιεχόμενο είναι επιθυμητό να παρουσιαστεί μέσα στο `div` με `id myDiv`.

[52] [54]

2.15.5 Το Γεγονός `onreadystatechange` της AJAX

Το γεγονός `onreadystatechange` αποτελεί την πρώτη ιδιαίτερης σημασίας ιδιότητα του αντικειμένου `XMLHttpRequest` της τεχνικής AJAX και περιέχει μία συνάρτηση, η οποία περιέχει είτε τον κώδικά της είτε το όνομα μίας άλλης συνάρτησης που έχει οριστεί σε κάποιο άλλο σημείο του κώδικα της εκάστοτε Ιστοσελίδας, και ενεργοποιείται (triggered) κάθε φορά που η τιμή μίας δεύτερης βασικής ιδιότητας του αντικειμένου `XMLHttpRequest`, της `readyState`, αλλάζει. Η ιδιότητα `readyState` καταχωρεί την κατάσταση του αντικειμένου `XMLHttpRequest` και παίρνει τιμές από 0 μέχρι 4. Παρακάτω θα φανεί η έννοια καθεμιάς από αυτές τις πέντε τιμές:

- 0: Δεν υπήρξε αρχικοποίηση του αιτήματος του εκάστοτε πελάτη προς τον εκάστοτε διακομιστή Ιστού.
- 1: Η σύνδεση του εκάστοτε πελάτη με τον εκάστοτε εξυπηρετητή Ιστού έχει εδραιωθεί.

- 2: Το αίτημα του εκάστοτε πελάτη στάλθηκε στον εκάστοτε διακομιστή Ιστού.
- 3: Γίνεται επεξεργασία από τον εκάστοτε εξυπηρετητή Ιστού του αιτήματος του εκάστοτε πελάτη προς αυτόν το διακομιστή.
- 4: Η επεξεργασία του αιτήματος του εκάστοτε πελάτη προς τον εκάστοτε εξυπηρετητή Ιστού έχει τελειώσει και το αποτέλεσμα είναι έτοιμο να σταλεί από αυτόν το διακομιστή στο συγκεκριμένο πελάτη.

Μία τρίτη σημαντική ιδιότητα του αντικειμένου XMLHttpRequest είναι η status. Αυτή η ιδιότητα έχει τις τιμές 200 και 404. Στη συνέχεια θα αναλυθεί το νόημα καθεμιάς από αυτές τις δύο τιμές:

- 200: Το αίτημα του εκάστοτε πελάτη προς τον εκάστοτε διακομιστή Ιστού είναι εφικτό να ολοκληρωθεί.
- 404: Δεν βρέθηκε η διεύθυνση στον Ιστό της πηγής δεδομένων ή του σεναρίου στο οποίο θα γίνει η επεξεργασία του αιτήματος του εκάστοτε πελάτη προς τον εκάστοτε εξυπηρετητή Ιστού.

Έτσι, η if συνθήκη που εξασφαλίζει ότι το αίτημα του εκάστοτε πελάτη προς τον εκάστοτε εξυπηρετητή Ιστού θα ολοκληρωθεί μόνο αν η readyState έχει την τιμή 4 και το status την τιμή 200 έχει τη μετέπειτα σύνταξη:

```
if (xmlhttp.readyState==4 && xmlhttp.status==200)
{
  document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
}
```

[52] [53] [54]

2.15.6 Τα Πολλαπλά Αντικείμενα XMLHttpRequest της AJAX

Μία καλή πρακτική, αν χρησιμοποιείται η τεχνική AJAX σε περισσότερα από ένα σημεία μίας Ιστοσελίδας είναι η χρήση μίας σταθερής συνάρτησης για τη δημιουργία του αντικειμένου XMLHttpRequest, η οποία θα καλείται μέσα από τις συναρτήσεις κάθε AJAX διεργασίας. Στις συναρτήσεις αυτές, λοιπόν, θα καλείται η συνάρτηση loadXMLDoc που δημιουργεί το αντικείμενο XMLHttpRequest με παραμέτρους τη διεύθυνση στον Ιστό του αρχείου που περιέχει το αποτέλεσμα-απάντηση του εκάστοτε εξυπηρετητή Ιστού στον εκάστοτε πελάτη και τη συγκεκριμένη σταθερή συνάρτηση, που περιέχει είτε τον κώδικά της είτε το όνομα μίας άλλης συνάρτησης, που έχει οριστεί σε κάποιο άλλο σημείο του κώδικα της εκάστοτε Ιστοσελίδας, και η οποία θα εκτελεστεί, εφόσον το γεγονός onreadystatechange είναι έτοιμο (readyState=4 και status=200). Για του λόγου το αληθές, στο παρακάτω παράδειγμα η συνάρτηση loadXMLDoc() είναι η σταθερή συνάρτηση στην οποία δημιουργείται το αντικείμενο XMLHttpRequest, εισάγοντας από τη συνάρτηση showHint() που την καλεί δύο παραμέτρους, οι οποίες έχουν ως εξής:

- I. Τη διεύθυνση στον Ιστό του αρχείου που περιέχει το αποτέλεσμα-απάντηση του εκάστοτε εξυπηρετητή Ιστού στον εκάστοτε πελάτη.
 - II. Τη σταθερή συνάρτηση που ελέγχει το onreadystatechange, ώστε να στείλει ο εκάστοτε διακομιστής Ιστού τα δεδομένα που θα εμφανιστούν στον εκάστοτε φυλλομετρητή.
- Το παράδειγμα είναι το πιο κάτω:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1253" />
<title>Untitled Document</title>

<script type="text/javascript">
var xmlhttp;
function loadXMLDoc(url,cfunc) {
  if (window.XMLHttpRequest) { // Code for IE7+, Firefox, Chrome, Opera, Safari:
    xmlhttp=new XMLHttpRequest();
```

```

    }
    else { // Code for IE6, IE5:
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=cfunc;
    xmlhttp.open("GET",url,true)
    xmlhttp.send();
}

function showHint(str) {
    loadXMLDoc("get.towns.php?" +str,function() {
        if (xmlhttp.readyState==4 && xmlhttp.status==200) {
            document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
        }
    });
}

</script>
</head>

<body>
<h3>Please type the name of the town you wish:</h3>
<form action="">
Town: <input type="text" id="text1" onkeyup="showhint(this.value) />
</form>
<p>Destination: <span id="txtHint"></span></p>
</body>
</html>

```

Επιπλέον, αν είναι ωφέλιμο να γίνει χρήση της τεχνικής AJAX και σε ένα δεύτερο σημείο μίας Ιστοσελίδας, όπως είναι, φερειπείν, σχετικά με το παραπάνω παράδειγμα, στο div με id="shout-box", όπου θα εμφανίζονται τα μηνύματα που θα γράφουν οι επισκέπτες της Σελίδας του παρόντος παραδείγματος, τότε απλά θα πρέπει στον κώδικα του εν λόγω παραδείγματος να προστεθεί αμέσως μετά τη συνάρτηση showHint() η παρακάτω συνάρτηση:

```

function getShoutBox() {
    loadXMLDoc("getShoutboxMsg.php",function() {
        if (xmlhttp.readyState==4 && xmlhttp.status==200) {
            document.getElementById("shoutbox").innerHTML=xmlhttp.responseText;
        }
    });
}

```

Στο πιο πάνω απόσπασμα κώδικα η συνάρτηση getShoutBox() καλεί τη συνάρτηση loadXMLDoc() και της περνάει δύο παραμέτρους:

- I. Τη url (getShoutboxMsg.php) του αρχείου που περιέχει το αποτέλεσμα-απάντηση του εκάστοτε εξυπηρετητή Ιστού στον εκάστοτε πελάτη.
- II. Τη συνάρτηση που κάθε φορά ελέγχει το γεγονός onreadystatechange της τεχνικής AJAX, προκειμένου να στείλει ο εκάστοτε διακομιστής Ιστού τα δεδομένα που θα εμφανιστούν στον εκάστοτε περιηγητή.

[52] [53] [54]

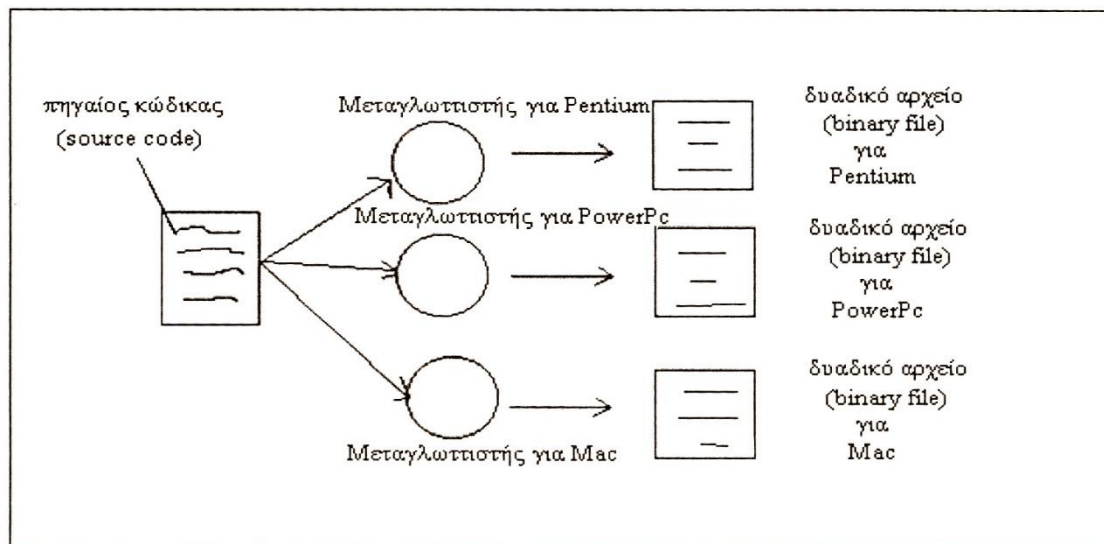
2.16 Η Γλώσσα Προγραμματισμού, Java

2.16.1 Εισαγωγή στην Java

Η Java είναι μία νέα σχετικά γλώσσα προγραμματισμού, η οποία σε αντίθεση με άλλες γλώσσες, γνωστές ήδη από τρεις και πάνω δεκαετίες, εμπεριέχει όλα τα χαρακτηριστικά της εξέλιξης της επιστήμης των υπολογιστών. Με δύο λόγια, θα μπορούσε να πει κανείς ότι είναι μία αντικειμενοστρεφής γλώσσα που επιτρέπει σε οποιονδήποτε υπολογιστή, όπου κι αν βρίσκεται αυτός, να έχει πρόσβαση και να χρησιμοποιεί μία εφαρμογή εγκατεστημένη σε κάποιο δίκτυο. Αν τη συγκρίνουμε από πλευράς δομής με τις άλλες γλώσσες προγραμματισμού, η Java μοιάζει περισσότερο με τη C. Της μοιάζει αρκετά στον τρόπο σύνταξης και τη φιλοσοφία, αλλά δεν είναι C.

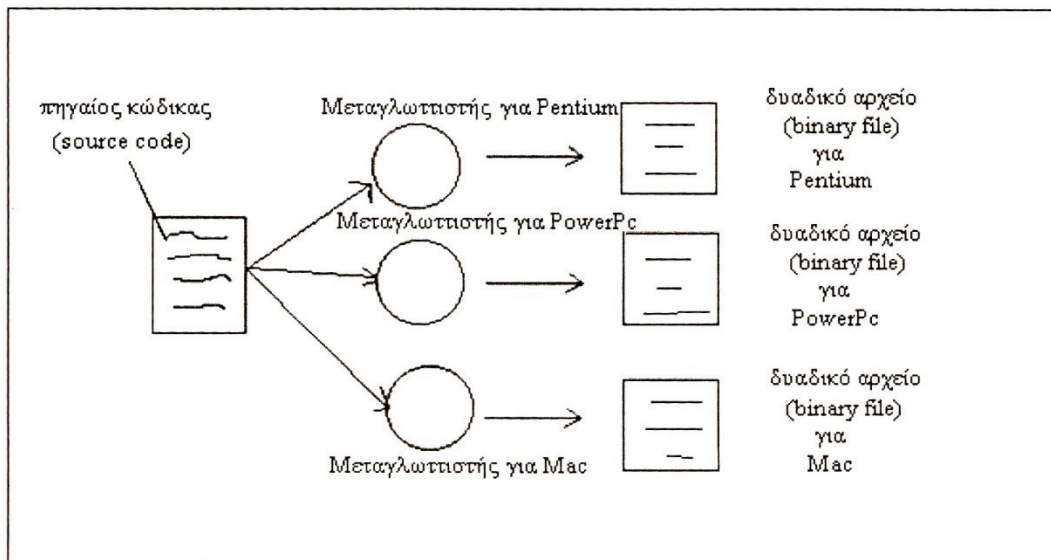
Η Java μπορεί να δημιουργήσει προγράμματα, τα οποία τρέχουν σε μία ποικιλία υπολογιστών και λειτουργικών συστημάτων. Αυτό γίνεται καλύτερα αντιληπτό, αν κατανοηθεί πώς λειτουργεί ο μεταγλωττιστής μίας παραδοσιακής γλώσσας και πώς ο μεταγλωττιστής της Java.

Σε μία παραδοσιακή γλώσσα (πχ Pascal, C), ο μεταγλωττιστής μετατρέπει το εκάστοτε πηγαίο πρόγραμμα (source code) σε δυαδικό αρχείο (binary file), δηλαδή σε κώδικα γλώσσας μηχανής κατάλληλο για τον εκάστοτε κατάλληλο επεξεργαστή, όπως παρουσιάζεται και στην παρακάτω Εικόνα:



Εικόνα 21: Η λειτουργία του μεταγλωττιστή μίας παραδοσιακής γλώσσας προγραμματισμού

Τα πράγματα, ωστόσο, είναι διαφορετικά στην περίπτωση της γλώσσας Java. Το περιβάλλον ανάπτυξης περιλαμβάνει δύο μέρη: α) το μεταγλωττιστή, ο οποίος, αντί για δυαδικό αρχείο, δημιουργεί ένα είδος κώδικα κάθε φορά που είναι ανεξάρτητος από πλατφόρμα και λέγεται κώδικας byte, και β) το διερμηνευτή κώδικα byte (προσαρμοσμένο σε κάθε πλατφόρμα), ο οποίος ονομάζεται και Εικονική Μηχανή Java (Java Virtual Machine-JVM) ή Διερμηνευτής Java Χρόνου Εκτέλεσης (Java Runtime Interpreter). Ο διερμηνευτής κώδικα byte διαβάζει τον εκάστοτε κώδικα byte και εκτελεί κατάλληλες εντολές γλώσσας μηχανής, που αντιστοιχούν στον αρμόζων σε κάθε φάση επεξεργαστή, όπως φαίνεται και στην παρακάτω Εικόνα:



Εικόνα 22: Η λειτουργία του μεταγλωττιστή της γλώσσας προγραμματισμού Java

Τα πιο συνηθισμένα προγράμματα πλοήγησης στο Διαδίκτυο εμπεριέχουν την JVM. Όταν, λοιπόν, φορτώσουν μία Ιστοσελίδα από το Διαδίκτυο, η οποία περιέχει οδηγίες για την εκτέλεση ενός Java προγράμματος, αυτόματα κατεβάζουν τον Java κώδικα από την περιοχή της εν λόγω Σελίδας και τρέχουν την JVM πάνω σε αυτόν τον κώδικα. Έτσι, επιτυγχάνεται η παρουσίαση κινούμενων εικόνων, μουσικής και πολυμέσων, φορμών εισαγωγής δεδομένων που είναι γραμμένα σε γλώσσα Java και περιέχονται στην εκάστοτε Σελίδα.

[55]

2.16.2 Οι Βασικοί Συντακτικοί Κανόνες της Java

Οι κύριοι κανόνες σύνταξης που χαρακτηρίζουν τη γλώσσα Java είναι οι ακόλουθοι:

- Η Java είναι γλώσσα που κάνει διάκριση ανάμεσα στα πεζά (μικρά) και τα κεφαλαία γράμματα, δηλαδή είναι case sensitive Language, κάτι που σημαίνει ότι οι λέξεις της γλώσσας (πχ μεταβλητές, εντολές) με κεφαλαία είναι διαφορετικές από τις ίδιες λέξεις με πεζά.
- Τα σύμβολα που συναντώνται στη γλώσσα Java (πχ παρενθέσεις, άγκιστρα, εισαγωγικά) τοποθετούνται πάντοτε ανά ζεύγη.
- Κάθε μία από τις εντολές ή τις προτάσεις της γλώσσας Java τερματίζεται υποχρεωτικά με το σύμβολο του ελληνικού ερωτηματικού (;). Αξίζει να σημειωθεί, σε αυτό το σημείο, πως η χρήση του συγκεκριμένου συμβόλου είναι πάντα υποχρεωτική στο τέλος εντολών η προτάσεων της Java, σε αντίθεση με τη γλώσσα JavaScript, όπου κάτω από προϋποθέσεις, η χρήση του μπορεί να παρακαμφθεί.
- Στη γλώσσα Java αρκετά συχνά πραγματοποιείται χρήση προσδιοριστών (identifiers). Οι προσδιοριστές είναι ονόματα που εκμεταλλεύονται οι προγραμματιστές, για να ονομάσουν μεταβλητές, κλάσεις, μεθόδους, μεταβλητές και άλλες συναφείς οντότητες της συγκεκριμένης γλώσσας. Οι προσδιοριστές έχουν συγκεκριμένους κανόνες ονομασίας. Έτσι, το όνομα ενός προσδιοριστή στη Java είναι μία απεριόριστου μήκους ακολουθία χαρακτήρων του συστήματος Unicode, η οποία, ωστόσο, πρέπει να αρχίζει είτε από γράμμα, είτε από το χαρακτήρα του δολαρίου (\$), είτε από το χαρακτήρα της υπογράμμισης (_) (παρόλο που η χρήση του \$ ή της _ δεν ενθαρρύνεται), και να μην περιέχει κενά. Δεν πρέπει σε καμία περίπτωση να αρχίζει με αριθμό ή κάποιον άλλο μη αλφαβητικό χαρακτήρα, εκτός του \$ ή της _, ούτε να ταυτίζεται με κάποιες από τις δεσμευμένες λέξεις της Java (που θα παρουσιαστούν στη συνέχεια). Μετά τον πρώτο χαρακτήρα, οι άλλοι μπορεί να είναι είτε γράμματα, είτε αριθμοί, είτε \$ είτε _. Επιπρόσθετα, αξίζει να τονιστεί πως το όνομα ενός προσδιοριστή είναι ωφέλιμο να αποτελείται από ολόκληρες λέξεις και όχι από κρυπτογραφημένες συντμήσεις. Τέλος, στην περίπτωση που το όνομα του προσδιοριστή αποτελείται από μία και μόνο λέξη, καλό είναι η λέξη αυτή να γράφεται με μικρά γράμματα, ενώ, αν συνίσταται από περισσότερες από μία λέξεις, είναι θεμιτό

- κάθε μία από αυτές τις λέξεις, πέραν της πρώτης, να ξεκινά με κεφαλαίο γράμμα.
- Η γλώσσα Java, εκτός από τους προσδιοριστές (identifiers), χρησιμοποιεί και ειδικούς προσδιοριστές (modifiers). Πρόκειται για λέξεις-κλειδιά που τοποθετούνται μπροστά από τα ονόματα κλάσεων, μεθόδων ή μεταβλητών τους, με στόχο να τροποποιήσουν τη λειτουργία τους. Οι προσδιοριστές αυτοί χωρίζονται σε δύο κύριες κατηγορίες, οι οποίες είναι οι παρακάτω:
 - ο Προσδιοριστές πρόσβασης (access modifiers): Χρησιμοποιούνται, για να καθορίσουν τα επίπεδα πρόσβασης κλάσεων, μεθόδων ή μεταβλητών τους. Χωρίζονται στις ακόλουθους:
 - public,
 - private,
 - protected.
 - ο Προσδιοριστές μη πρόσβασης (non access modifiers): Διατίθενται, ώστε να επιτελέσουν διαφορετικούς σκοπούς από αυτούς που εξυπηρετούν οι προσδιοριστές πρόσβασης. Ταξινομούνται στους παρακάτω:
 - static: Χρησιμοποιείται για τη δημιουργία στατικών μεταβλητών και μεθόδων κλάσεων.
 - final: Διατίθεται για την τελική οριστικοποίηση κλάσεων, μεθόδων και μεταβλητών τους.
 - abstract: Εκμεταλλεύεται για την κατασκευή αόριστων κλάσεων και μεθόδων τους.
 - synchronized και volatile: Εφαρμόζονται στα νήματα εκτέλεσης (threads).
 - Η γλώσσα Java χρησιμοποιεί ένα σύνολο από λέξεις για συγκεκριμένους σκοπούς (πχ τύπους δεδομένων, ονόματα εντολών και μεθόδων). Οι λέξεις αυτές καλούνται δεσμευμένες (reserved words) ή λέξεις-κλειδιά (keywords) και δεν πρέπει να διατίθενται από τους προγραμματιστές για άλλους σκοπούς. Οι δεσμευμένες λέξεις της Java είναι οι εξής: abstract, assert, boolean, break, byte, case, catch, char, class, const, continue, default, do, double, else, extends, final, finally, float, for, goto, if, implements, import, instanceof, int, interface, long, native, new, package, private, protected, public, return, short, static, strictfp, super, switch, synchronized, this, throw, throws, transient, try, void, volatile, while.
 - Μία λέξη ή φράση που πρέπει να τυπωθεί όπως είναι μέσα στον Java κώδικα της εκάστοτε Ιστοσελίδας, τοποθετείται εντός διπλών εισαγωγικών ("").
 - Σε πολλά σημεία ενός Java κώδικα είναι επιβεβλημένη η χρήση σχολίων. Τα σχόλια (comments) είναι κείμενο το οποίο δεν εκτελείται από έναν υπολογιστή, αλλά είναι θεμιτό να γράφεται σε διάφορα σημεία του προαναφερόμενου κώδικα, ώστε να θυμίζει στον εκάστοτε προγραμματιστή τη λειτουργία διαφόρων τμημάτων αυτού του κώδικα, ώστε να είναι εύκολη η συντήρηση ή η τροποποίησή τους, αν χρειαστεί, τόσο από τον ίδιο όσο και από άλλους προγραμματιστές. Τα σχόλια στην Java καταχωρούνται μετά από τους χαρακτήρες //, αν εκτείνονται σε μία γραμμή, ή ανάμεσα στα σύμβολα /* και */, αν επεκτείνονται σε περισσότερες από μία γραμμές.

[56]

2.16.3 Οι Μεταβλητές, τα Δεδομένα και οι Τελεστές της Java

2.16.3.1 Οι Μεταβλητές

Ως μεταβλητή (variable) στη γλώσσα Java, θεωρείται μία περιοχή μνήμης στην οποία έχει δοθεί ένα όνομα και χρησιμοποιείται για αποθήκευση δεδομένων ενός ορισμένου τύπου. Αφού καθοριστεί ο τύπος των δεδομένων που θα κρατά μία μεταβλητή στη Java, από κει και πέρα ο μεταγλωττιστής της γλώσσας έχει τη δυνατότητα να ελέγχει όσα δεδομένα περνούν από αυτήν. Προτού γίνει χρήση μίας μεταβλητής πρέπει να δηλωθεί το όνομά της και ο τύπος της. Οι δηλώσεις (declarations) μεταβλητών πραγματοποιούνται συνήθως στην αρχή του εκάστοτε Java προγράμματος, ωστόσο υπάρχει η δυνατότητα να γίνουν και σε άλλες θέσεις αυτού. Στην αρχή, τώρα, της δήλωσης μίας μεταβλητής μπαίνει ο τύπος των δεδομένων που είναι επιθυμητό να αποθηκευτούν σε αυτή τη μεταβλητή και στη συνέχεια το όνομά της, που επιλέγει ο εκάστοτε προγραμματιστής και το οποίο πρέπει να συμφωνεί με τους κανόνες ονομασίας που διέπουν τους προσδιοριστές της Java. Επίσης, να αναφερθεί ότι, αν σε έναν

Java κώδικα υπάρχουν πολλές μεταβλητές του ίδιου τύπου, δύναται να δηλωθούν στην ίδια γραμμή του συγκεκριμένου κώδικα, αφού χωριστούν με κόμμα (,) μεταξύ τους. Αφού, τώρα, οριστεί μία μεταβλητή πρέπει να της καταχωρηθεί μία αρχική τιμή (αν και αυτό δεν είναι πάντα απαραίτητο). Η εκχώρηση (assignment) αυτής της τιμής γίνεται με το σύμβολο ίσον (=). Αριστερά του = είναι το όνομα της προαναφερόμενης μεταβλητής και δεξιά η προαναφερόμενη τιμή της. Φυσικά, για λόγους απλότητας συχνά η δήλωση και η αρχικοποίηση μίας μεταβλητής γίνονται σε μία γραμμή κώδικα, συνδυάζοντας τα όσα προαναφέρθηκαν.

Στη γλώσσα Java διακρίνονται τέσσερις κατηγορίες μεταβλητών, που είναι οι επόμενες:

- I. Μεταβλητές κλάσης ή μεταβλητές-μέλη ή στατικά πεδία (class variables or member variables or static fields): Στην ουσία πρόκειται για πεδία της γλώσσας Java τα οποία δηλώνονται ως στατικά με τη χρήση της λέξης-κλειδί static. Έτσι, ο μεταγλωττιστής της Java αντιλαμβάνεται ότι υπάρχει μόνο ένα αντίγραφο μίας μεταβλητής κλάσης σε ισχύ, ανεξάρτητα από το πόσα στιγμιότυπα-αντικείμενα αυτής της κλάσης έχουν δημιουργηθεί. Αν, μάλιστα, σε μία μεταβλητή κλάσης της γλώσσας Java προστεθεί και ο προσδιοριστής final μετά τον προσδιοριστή static, τότε η τιμή της δεν θα αλλάξει σε καμία περίπτωση. Στο σημείο αυτό, να σημειωθεί πως στις μεταβλητές-μέλη, αν δεν τους καταχωρηθεί αρχική τιμή από τον εκάστοτε προγραμματιστή, τότε τους καταχωρείται αυτόματα από το σύστημα.
- II. Μεταβλητές στιγμιότυπου ή μη στατικά πεδία (instance variables or non static fields): Από τεχνικής άποψη οι κλάσεις της γλώσσας Java αποθηκεύουν ένα στιγμιότυπο-αντικείμενό τους, με άλλα λόγια μία μοναδική, ανεξάρτητη κατάσταση τους, σε μη στατικά πεδία, δηλαδή σε πεδία που έχουν δηλωθεί χωρίς τη χρήση της λέξης-κλειδί static. Τα μη στατικά πεδία της Java καλούνται, επίσης, και μεταβλητές στιγμιότυπου, διότι οι τιμές τους είναι μοναδικές σε κάθε στιγμιότυπο μίας κλάσης.
- III. Τοπικές μεταβλητές (local variables): Όπως ακριβώς το στιγμιότυπο-αντικείμενο μίας κλάσης της γλώσσας Java αποθηκεύει την ξεχωριστή κατάστασή του σε πεδία, έτσι και οι μέθοδοί του αποθηκεύουν τις προσωρινές καταστάσεις τους σε τοπικές μεταβλητές. Δεν υπάρχει λέξη-κλειδί για την επισήμανση μίας μεταβλητής της Java ως τοπική. Ο καθορισμός μίας μεταβλητής ως τοπική, προέρχεται εξολοκλήρου από το γεγονός πως η συγκεκριμένη μεταβλητή ορίζεται «τοπικά» εντός μίας μεθόδου. Αυτός είναι και ο λόγος που οι τοπικές μεταβλητές είναι ορατές μόνο από τις μεθόδους μίας κλάσης εντός των οποίων δηλώθηκαν και δεν μπορούν να προσπελαστούν από το υπόλοιπο κομμάτι της κλάσης. Εδώ, να παρατηρηθεί πως οι τοπικές μεταβλητές δεν αρχικοποιούνται από το σύστημα. Τη δουλειά αυτή είναι υποχρεωτικό να την πράξει ο εκάστοτε προγραμματιστής, αλλιώς ο μεταγλωττιστής της Java θα εμφανίσει μήνυμα λάθους.
- IV. Παράμετροι (parameters): Καλούνται τα ορίσματα που πιθανόν να δέχεται μίας μέθοδος-συνάρτηση ενός στιγμιότυπου-αντικείμενου μίας κλάσης της γλώσσας Java.

[55] [57]

2.16.3.2 Τα Δεδομένα

Η γλώσσα Java υποστηρίζει ένα σύνολο τύπων δεδομένων, δηλαδή μία ομάδα τιμών που είναι σε θέση να αναπαριστά και να διαχειρίζεται ανεξάρτητα από την υφιστάμενη σε κάθε περίπτωση πλατφόρμα λογισμικού που χρησιμοποιείται. Τα πρωταρχικά δεδομένα της συγκεκριμένης γλώσσας είναι τα ακόλουθα:

- I. Οι ακέραιοι αριθμοί (integers): Είναι τεσσάρων τύπων: byte, short, int και long και μπορεί να είναι είτε θετικοί είτε αρνητικοί. Εύλογα, εδώ, γεννιέται το ερώτημα ποιος τύπος ακεραίου της γλώσσας Java θα επιλέγεται στο εκάστοτε πρόγραμμα. Ο κανόνας είναι να διαλέγεται ο μικρότερος τύπος δεδομένων στον οποίο μπορεί να αποθηκευτεί ο μεγαλύτερος αριθμός που θα εκμεταλλευτεί αυτό το πρόγραμμα, ώστε να αυξηθεί η ταχύτητα εκτέλεσής του. Παρόλα αυτά, συνήθως χρησιμοποιείται ο τύπος int. Όλοι οι ακέραιοι αριθμοί, τώρα, θεωρούνται ότι βρίσκονται στο δεκαδικό σύστημα, αλλά είναι δυνατόν να εκφραστούν και με οκταδική ή δεκαεξαδική μορφή. Αξίζει, ακόμα, να αναφερθεί ότι οι ακέραιοι αριθμοί παρουσιάζουν μία ιδιότητα που ονομάζεται αναδίπλωση (wrapping). Αν, δηλαδή, αυξηθεί ή μειωθεί η τιμή μίας μεταβλητής πέρα από τη μέγιστη ή την ελάχιστη τιμή της, αντίστοιχα, η συγκεκριμένη μεταβλητή θα αναδιπλωθεί. Αυτό πρακτικά σημαίνει ότι, για παράδειγμα, αν αυξηθεί κατά ένα μία ακέραια μεταβλητή τύπου byte με τιμή 127 (που είναι η μέγιστη τιμή της), τότε αυτή η μεταβλητή θα

πάρει την τιμή -128 (που είναι η ελάχιστη τιμή της). Αντίστροφα, αν μειωθεί κατά ένα η ελάχιστη τιμή της προαναφερόμενης μεταβλητής (που είναι το -128), τότε η εν λόγω μεταβλητή θα πάρει τη μέγιστη τιμή της (που είναι το 127). Στον Πίνακα που έπεται δείχνονται οι ακέραιοι τύποι δεδομένων, το μέγεθος της μνήμης που πιάνουν καθώς και το εύρος των τιμών τους:

Τύπος	Δεσμευόμενη μνήμη σε bits	Εύρος τιμών
byte	8	-128 ως 127
short	16	-32.768 ως 32.767
int	32	-2^{31} ως $2^{31}-1$
long	64	-2^{63} ως $2^{63}-1$

Πίνακας 24: Οι ακέραιοι τύποι δεδομένων της Java

- II. Οι αριθμοί κινητής υποδιαστολής (floating point numbers): Περιλαμβάνουν ακέραιο και δεκαδικό μέρος και δύναται να είναι, επίσης, είτε θετικοί είτε αρνητικοί. Ανάλογα με την ακρίβεια που παρέχουν οι αριθμοί κινητής υποδιαστολής της γλώσσας Java χωρίζονται σε float και double. Στον Πίνακα που έπεται παρουσιάζονται οι αριθμοί κινητής υποδιαστολής, το μέγεθος της μνήμης που καταλαμβάνουν, το εύρος των τιμών τους και η ακρίβεια που αυτοί προσφέρουν:

Τύπος	Δεσμευόμενη μνήμη σε bits	Εύρος τιμών	Ακρίβεια
float	4	$-3,4 \times 10^{38}$ ως $3,4 \times 10^{38}$	7 bits
double	8	$-1,7 \times 10^{308}$ ως $1,7 \times 10^{308}$	18 bits

Πίνακας 25: Οι αριθμοί κινητής υποδιαστολής της Java

- III. Οι λογικές τιμές (boolean): Είναι δύο ειδών: η αληθής (true) τιμή ή 1 και η ψευδής (false) τιμή ή 0. Οι λογικές τιμές της γλώσσας Java εφαρμόζονται κατά την επεξεργασία λογικών προτάσεων, όταν πρέπει να αποφασιστεί, αν μία συνθήκη ισχύει ή δεν ισχύει.
- IV. Οι χαρακτήρες (chars, από τη λέξη characters): Ένα δεδομένο τύπου χαρακτήρα της γλώσσας Java ισοδυναμεί με έναν οποιονδήποτε χαρακτήρα του συστήματος Unicode, ο οποίος περικλείεται σε απλά εισαγωγικά ('). Το σύστημα αυτό χρησιμοποιεί 2 bytes (16 bits) για την παράσταση ενός χαρακτήρα και με τον τρόπο αυτό συμπεριλαμβάνει τα αλφάβητα όλων των γλωσσών, διότι έχει τη δυνατότητα αποθήκευσης 65536 χαρακτήρων. Για να γίνει αναφορά σε ένα συγκεκριμένο χαρακτήρα Unicode, διατίθεται η ακολουθία διαφυγής \u, ακολουθούμενη από ένα τετρανηφίο δεκαδικό αριθμό. Έτσι, η ελάχιστη τιμή ενός χαρακτήρα Unicode είναι η '\u0000' (ή 0) και η μεγαλύτερη η '\uffff' (ή 65.535). Όσον αφορά δε τις ακολουθίες διαφυγής (escape sequences) πρόκειται για συγκεκριμένες σειρές χαρακτήρων, οι οποίες διατίθενται για την παράσταση ειδικών χαρακτήρων, όπως είναι, για παράδειγμα, η αλλαγή γραμμής ή το πλήκτρο οπισθοδρόμησης. Μία ακολουθία διαφυγής, λοιπόν, αποτελείται από μία ανάποδη κάθετο (\) μαζί με ένα χαρακτήρα. Μερικές από τις βασικότερες ακολουθίες διαφυγής της Java φαίνονται στον παρακάτω Πίνακα:

Ακολουθία	Σημασία
\b	Οπισθοδρόμηση (backspace)
\t	Χαρακτήρας στηλοθέτη (Tab)
\n	Αλλαγή γραμμής (line feed)
\f	Αλλαγή σελίδας (form feed)
\r	Χαρακτήρας επιστροφής (carriage return)
\"	" (διπλά εισαγωγικά) (double quotes)
\'	' (απλά εισαγωγικά) (single quotes)
\\	\ (ανάποδη κάθετος) (backslash)

Πίνακας 26: Οι πιο βασικές ακολουθίες διαφυγής της Java

Στο σημείο αυτό, να τονιστεί ιδιαίτερα πως, αν, αντί για ένα χαρακτήρα, καταχωρηθεί σε μία μεταβλητή ένα σύνολο από χαρακτήρες (δύο ή περισσότεροι), δηλαδή ένα αλφαριθμητικό ή μία συμβολοσειρά, όπως διαφορετικά καλείται, τότε γίνεται λόγος για μία μεταβλητή τύπου αλφαριθμητικού δεδομένου (string). Στη γλώσσα Java, ένα αλφαριθμητικό δεδομένο είναι κάτι περισσότερο από ένα επιπρόσθετο δεδομένο, πέρα των τεσσάρων μόλις προαναφερθέντων. Πρόκειται για ένα αντικείμενο της κλάσης String (που θα αναπτυχθεί εκτενέστερα στη συνέχεια). Αφού, λοιπόν, το αλφαριθμητικό δεδομένο είναι αντικείμενο, διαθέτει μεθόδους, που βοηθούν στον καλύτερο χειρισμό του. Κάθε αλφαριθμητικό δεδομένο πρέπει να περικλείεται σε διπλά εισαγωγικά ("), σε αντιδιαστολή με τους χαρακτήρες που περικλείονται σε απλά εισαγωγικά ('). Εδώ, να επισημανθεί ότι αν σε μία μεταβλητή καταχωρηθεί ως τιμή ένας και μόνο χαρακτήρας, εντός, όμως, διπλών και όχι μονών, εισαγωγικών, τότε η συγκεκριμένη μεταβλητή θα είναι τύπου String και όχι char, όπως, ίσως, λανθασμένα να θεωρούνταν.

Συνεχίζοντας την ανάλυση για τους τύπους δεδομένους της γλώσσας Java, θα γίνει αναφορά στις αρχικές τιμές των μεταβλητών-μελών της Java. Στην Java, λοιπόν, όταν δηλωθεί μία μεταβλητή-μέλος και δεν της δοθεί αρχική τιμή από τον εκάστοτε προγραμματιστή (πρακτική που, ωστόσο, πρέπει να αποφεύγεται, διότι θεωρείται κακή τεχνική προγραμματισμού), τότε ο μεταγλωττιστής της εν λόγω γλώσσας της καταχωρεί αυτόματα μία αρχική τιμή, σε αντίθεση με τις τοπικές μεταβλητές, στις οποίες αν δεν καταχωρηθεί μία αρχική τιμή από τον εκάστοτε προγραμματιστή, θα προκύψει λάθος κατά το μεταγλωττισμό του Java κώδικα. Η τιμή αυτή εξαρτάται από τον τύπο της μεταβλητής και είναι εν γένει 0 ή null. Οι αρχικές τιμές που είναι δυνατό να καταχωρήσει αυτόματα ο μεταγλωττιστής της Java σε μία μεταβλητή-μέλος, με βάση τον τύπο αυτής, φαίνονται στον παρακάτω Πίνακα:

Τύπος δεδομένου	Αρχική τιμή
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
boolean	false
char	'\u0000'
String (ή οποιοδήποτε αντικείμενο)	null

Πίνακας 27: Οι αρχικές τιμές των μεταβλητών-μελών της Java

Κλείνοντας τη συζήτηση για τους τύπους δεδομένους της γλώσσας Java, θα ήταν παράλειψη να γίνει μνεία στον όρο μετατροπή δεδομένου (casting). Με τον συγκεκριμένο όρο νοείται μία διαδικασία κατά την οποία μετατρέπεται έναν τύπος δεδομένου της Java σε έναν άλλο. Η αλλαγή αυτή γίνεται τόσο για πρωταρχικούς τύπους μεταβλητών όσο και για οποιονδήποτε τύπο κλάσεων. Σύμφωνα, λοιπόν, με τη διαδικασία αυτή, τοποθετείται ο τύπος που είναι επιθυμητό να έχει η εκάστοτε μεταβλητή σε παρένθεση, πριν από το όνομα της μεταβλητής, σύμφωνα με την παρακάτω σύνταξη:

(variableType) variableName;

Όπου:

- *variableType*: Ο επιθυμητός τύπος δεδομένων της γλώσσας Java στον οποίο θα μετατραπεί μία μεταβλητή ενός Java κώδικα.
- *variableName*: Το όνομα της εκάστοτε μεταβλητής ενός Java κώδικα, η οποία είναι επιθυμητό να μετατραπεί σε άλλο τύπο δεδομένων της γλώσσας Java, από αυτόν που είναι.

[55] [56]

2.16.3.3 Οι Τελεστές

Η γλώσσα Java χρησιμοποιεί κατά βάση τους τελεστές που εκμεταλλεύεται και η γλώσσα σεναρίων JavaScript.

[55]

2.16.4 Οι Δομές Προγραμματισμού της Java

Οι δομές προγραμματισμού της γλώσσας Java είναι ίδιες με τις δομές προγραμματισμού της JavaScript, με την ειδοποιό διαφορά πως στην Java δεν απαντάται ο βρόγχος επανάληψης `for...in`, ο οποίος χρησιμεύει για την κυκλική κίνηση διαμέσου των ιδιοτήτων ενός αντικειμένου της JavaScript [55]

2.16.5 Οι Μέθοδοι της Java

Τα προβλήματα που συνήθως καλείται να επιλύσει ένας προγραμματιστής, κατά τη δημιουργία μίας πραγματικής εφαρμογής, είναι αρκετά πολύπλευρα και πολυεπίπεδα. Η πείρα έχει δείξει ότι ο καλύτερος τρόπος για να κατασκευάσει και να συντηρήσει κάποιος ένα πρόγραμμα, είναι να το φτιάξει χρησιμοποιώντας μικρά και απλά κομμάτια κώδικα. Τα κομμάτια αυτά είναι γνωστά στον προγραμματισμό με τον όρο μονάδες (modules).

Στην Java είναι εφικτή η κατασκευή μονάδων χρησιμοποιώντας μεθόδους (methods) και κλάσεις (classes). Εδώ θα εξεταστούν οι μέθοδοι, ενώ λίγο πιο κάτω οι κλάσεις.

Με τον όρο μέθοδο (method) νοείται ένα σύνολο από δηλώσεις και εντολές, οι οποίες ομαδοποιούνται, ώστε να αποτελέσουν ένα ανεξάρτητο μίνι-πρόγραμμα. Σε άλλες γλώσσες προγραμματισμού οι μέθοδοι ονομάζονται συναρτήσεις (functions), διαδικασίες (procedures), υπορουτίνες (subroutines) ή υποπρογράμματα (subprograms).

Μία μέθοδος δηλώνεται με την ακόλουθη σύνταξη:

```
modifiers returnType methodName( parameters);  
{  
statements;  
}
```

Όπου:

- *modifiers*: Ο συνδυασμός δύο εκ των ειδικών προσδιοριστών (modifiers) της Java, που παρουσιάστηκαν παραπάνω. Ο πρώτος καθορίζει τον τύπο προσπέλασης της εκάστοτε μεθόδου της γλώσσας (public, private, protected) και ο δεύτερος είναι ένας από τους ειδικούς προσδιοριστές μη-προσπέλασης (static, final, abstract). Ωστόσο, υπάρχει η δυνατότητα ένας από τους δύο ή, ακόμα, και οι δύο προσδιοριστές να παραλείπονται.
- *returnType*: Ο τύπος δεδομένου της τιμής επιστροφής της εκάστοτε Java μεθόδου, δηλαδή ο τύπος δεδομένου που επιστρέφει η μέθοδος σε αυτόν που την καλεί. Αν η μέθοδος δεν επιστρέφει καμία τιμή, τότε στη θέση αυτής της λέξης πρέπει να μπει η λέξη void.
- Το όνομα της εκάστοτε Java μεθόδου, το οποίο πρέπει να συμφωνεί με τους κανόνες ονομασίας των προσδιοριστών της γλώσσας, που αναπτύχθηκαν πιο πάνω.
- *parameters*: Οι παράμετροι ή, με άλλα λόγια, τα ορίσματα που τυχόν δέχεται η εκάστοτε Java μέθοδος και συνιστούν τις τιμές που περνούν σε αυτή, κατά την κλήση της από τα κατάλληλο σημείο του εκάστοτε προγράμματος, και οι οποίες θα υποστούν επεξεργασία στο εσωτερικό της. Μία μέθοδος δύναται να έχει ένα ή δύο και περισσότερα ορίσματα. Στην περίπτωση που είναι δύο και πάνω χωρίζονται μεταξύ τους με κόμμα (,). Μάλιστα, υπάρχουν φορές που μία μέθοδος δε φέρει κανένα όρισμα. Τότε γράφεται αποκλειστικά το ζεύγος των παρενθέσεων, δίχως να περικλείεται τίποτα ανάμεσά τους.
- *statements*: Οι δηλώσεις και οι εντολές οι οποίες ανήκουν στην εκάστοτε Java μέθοδο και αποτελούν τον κορμό-σώμα της μεθόδου. Το σώμα της μεθόδου αποτελείται συνήθως από μία ή περισσότερες δηλώσεις και από μία ή περισσότερες εντολές. Ωστόσο, δύναται να αποτελείται αποκλειστικά από μία και μόνο εντολή ή, ακόμα, και να είναι εντελώς κενό, δίχως, όμως, σε καμία περίπτωση να παραλείπεται το ζεύγος των αγκίστρων {} που το περικλείουν.

Για να καλέσει, τώρα, μία μέθοδο ο εκάστοτε χρήστης μέσα σε ένα πρόγραμμα της Java, αρκεί να καλέσει το όνομά της στο σημείο που τη χρειάζεται, τοποθετώντας τα κατάλληλα ορίσματα. Μόλις η μέθοδος τελειώσει, η εκτέλεση του προγράμματος συνεχίζεται από το σημείο ακριβώς που έγινε η κλήση της μεθόδου. Αν η μέθοδος δεν επιστρέφει αποτέλεσμα, η επιστροφή στο κυρίως πρόγραμμα γίνεται μετά το δεξί άγκιστρο ή με την κλήση της εντολής:

```
return;
```

Αν, όμως, η μέθοδος επιστρέφει αποτέλεσμα, η επιστροφή στο κυρίως πρόγραμμα γίνεται με την εντολή:

```
return expression;
```

Όπου *expression* μία οποιαδήποτε αποδεκτή έκφραση της Java.

[55]

2.16.6 Οι Πίνακες και τα Αλφαριθμητικά της Java

2.16.6.1 Οι Πίνακες

Σε σύνθετες εφαρμογές παρουσιάζεται συχνά η ανάγκη να γίνεται χρήση ομάδων τιμών του ίδιου τύπου. Αντί, λοιπόν, να ορίζεται μία ξεχωριστή μεταβλητή, για να κρατά κάθε μία από τις τιμές αυτού του τύπου, είναι προτιμότερο να δημιουργείται μία μεταβλητή με δείκτη ή αλλιώς ένας πίνακας (*array*).

Τα στοιχεία του πίνακα σημειώνονται με κοινό όνομα, που συνοδεύεται, όμως, από ένα δείκτη (*index**subscript*). Ο δείκτης είναι ένας ακέραιος αριθμός (ή μία ακέραιη έκφραση) τοποθετημένος μέσα σε αγκύλες. Μία μεταβλητή με δείκτη (πίνακας) μίας διάστασης μπορεί να θεωρηθεί ως μία στήλη δεδομένων. Πρέπει να τονιστεί ότι το πρώτο στοιχείο κάθε πίνακα είναι πάντα αυτό με δείκτη 0, ενώ το *n*-οστό έχει δείκτη *n*-1. Το πλήθος των μεταβλητών που περιλαμβάνει ο πίνακας αποτελεί το μήκος του. Όπως κάθε μεταβλητή, έτσι και ο πίνακας στη Java πρέπει να έχει έναν συγκεκριμένο τύπο, όπως *byte*, *short*, *int*, *long*, *float*, *double*, *boolean*, *char* ή *String*. Εκτός από τους βασικούς τύπους δεδομένων της συγκεκριμένης γλώσσας, ένας πίνακας μπορεί να έχει και τύπο που καθορίζει ο εκάστοτε προγραμματιστής της γλώσσας. Όλα τα στοιχεία του πίνακα είναι του ίδιου τύπου. Δε γίνεται, για παράδειγμα, να αποθηκευτούν στον ίδιο πίνακα δεδομένα τύπου *int* και *double*. Για να δημιουργήσουμε έναν πίνακα ακολουθούμε τρία βήματα: δήλωση, καταχώρηση και απόδοση αρχικής τιμής.

Αρχικά, ο πίνακας δηλώνεται με τον ίδιο τρόπο, όπως και οι άλλες μεταβλητές. Στη δήλωση, δίπλα στο όνομα της μεταβλητής τοποθετείται διπλή αγκύλη [] ως πρόθεμα ή επίθεμα, για να δειχθεί ότι η μεταβλητή αυτή είναι ένας πίνακας. Έτσι ένας πίνακας στη Java δηλώνεται σύμφωνα με τη σύνταξη:

```
arrayType [arrayName]; ή arrayType arrayName[];
```

Όπου:

- *arrayType*: Ο τύπος των δεδομένων που είναι επιθυμητό να κρατά ο πίνακας που δηλώνεται σε κάθε περίπτωση.
- *arrayName*: Το όνομα που είναι επιθυμητό να έχει ο πίνακας που δηλώνεται σε κάθε περίπτωση.

Ωστόσο, με τη δήλωση ενός πίνακα, απλώς καθορίζεται το είδος των στοιχείων που μπορεί αυτός να περιέχει. Δεν δημιουργούνται στοιχεία του πίνακα, ούτε δεσμεύεται κάποιος χώρος μνήμης, προκειμένου να αποθηκευτούν τα συγκεκριμένα στοιχεία. Για να δηλωθεί στο μεταγλωττιστή της Java ότι χρειάζεται να δημιουργηθεί ένας νέος πίνακας με όνομα *arrayName* ο οποίος επιθυμείται να δύναται να αποθηκεύσει *N* στοιχεία τύπου *arrayType* χρησιμοποιείται η δεσμευμένη λέξη *new* κατά την παρακάτω σύνταξη:

```
arrayType []arrayName; ή arrayType[] arrayName;  
arrayName=new arrayType[N];
```

Η προηγούμενη σύνταξη γράφεται και πιο σύντομα, όπως στη συνέχεια:

```
arrayType []arrayName=new arrayType[N]; ή arrayType arrayName[]=new arrayType[N];
```

Αφότου, τώρα, δηλωθεί και καταχωρηθεί ένας πίνακας, αυτό που απομένει είναι να του αποδοθούν αρχικές τιμές. Η διαδικασία αυτή υλοποιείται σύμφωνα με την επόμενη σύνταξη:

```
arrayName[0]=a0;  
arrayName[1]=a1;  
arrayName[2]=a2;  
arrayName[3]=a3;  
.....  
arrayName[N]=an;
```

Όπου $a_0, a_1, a_2, a_3, \dots, a_n$ οι τιμές που καταχωρούνται στις θέσεις $0, 1, 2, 3, \dots, N$, αντίστοιχα, του πίνακα arrayName.

Στην περίπτωση, βέβαια, που ένας πίνακας έχει πολλά στοιχεία, που οφείλουν να έχουν όλα την ίδια αρχική τιμή (πχ 0), τότε δεν ενδείκνυται να ορίζεται η τιμή κάθε στοιχείου χωριστά, αλλά είναι πιο βολικό να γίνεται χρήση μίας δομής επανάληψης (κατά προτίμηση της for), ώστε να αποδίδεται αρχική τιμή σε όλα τα στοιχεία ταυτόχρονα, σύμφωνα με την κάτωθι σύνταξη:

```
for (int i=0;i<=N;i++)  
{  
    arrayName[i]=0;  
}
```

Ένας άλλος τρόπος απόδοσης αρχικών σε έναν πίνακα είναι η τοποθέτηση αυτών μέσα σε αγκύλες. Στην περίπτωση τούτη ο ορισμός και η αρχικοποίηση του πίνακα γίνεται γράφοντας το πιο κάτω απόσπασμα κώδικα:

```
arrayType []arrayName; ή arrayType arrayName[];  
arrayName={a0, a1, a2, a3, ... , an };
```

Ο παραπάνω κώδικας έχει και πιο σύντομη μορφή, η οποία είναι η εξής:

```
arrayType []arrayName={a0, a1, a2, a3, ... , an }; ή arrayType arrayName[]={a0, a1, a2, a3, ... ,  
an };
```

Επιπρόσθετα υπάρχει η ευχέρεια να αποδοθούν αρχικές τιμές στα στοιχεία ενός πίνακα arrayName1 με τη βοήθεια ενός άλλου πίνακα arrayName2, ο οποίος είναι ήδη ορισμένος και του έχουν αποδοθεί αρχικές τιμές στα στοιχεία του.. Έτσι δημιουργούνται δύο μεταβλητές με δείκτη (πίνακες), οι οποίες δείχνουν στα ίδια στοιχεία της μνήμης. Στη συγκεκριμένη περίπτωση η δήλωση, η καταχώρηση και η αρχικοποίηση του πίνακα πραγματοποιείται με τις κατώτερες εντολές:

```
arrayType []arrayName1; ή arrayType arrayName1[];  
arrayName1=arrayName2;
```

Οι πιο πάνω εντολές ενσωματώνονται σε μία ως εξής

arrayType [*arrayName1=arrayName2*; ή *arrayType arrayName1[]=arrayName2*;

Συνεχίζοντας την κουβέντα για τους πίνακες, να τονιστεί ότι για να υπολογιστεί το πλήθος των στοιχείων ενός πίνακα, διατίθεται η πολύ βασική ιδιότητα `length`, κατά τη μετέπειτα σύνταξη:

arrayName.length;

Για το χειρισμό, τώρα, των πινάκων διατίθενται μία σειρά από μεθόδους. Ανάμεσά τους ξεχωρίζουμε την `binarySearch()`, η οποία ψάχνει ένα προς ένα τα στοιχεία ενός πίνακα προκειμένου να βρει μία συγκεκριμένη τιμή, με τη βοήθεια του δυαδικού αλγορίθμου αναζήτησης, την `copyOf()`, που αντιγράφει τα ένα προς ένα τα στοιχεία ενός πίνακα σε έναν άλλο πίνακα, την `equals()`, η οποία συγκρίνει αν δύο πίνακες είναι ίσοι μεταξύ τους, τη `fill()`, που συμπληρώνει τα στοιχεία ενός πίνακα με μία επιθυμητή τιμή, τη `sort()`, η οποία ταξινομεί τα στοιχεία ενός πίνακα με αύξουσα σειρά και την `toString()`, που μετατρέπει σε αλφαριθμητικό δεδομένο κάθε στοιχείο ενός πίνακα, ο οποίος κρατά οποιουδήποτε τύπου δεδομένα, εκτός από `String`.

[57]

2.16.6.2 Τα Αλφαριθμητικά

Με τον όρο αλφαριθμητικό ή συμβολοσειρά (`string`) εννοείται μία σειρά από χαρακτήρες. Διάφορες λέξεις αλλά και ολόκληρες προτάσεις συνιστούν παραδείγματα αλφαριθμητικών. Κάθε αλφαριθμητικό δεδομένο στη Java πρέπει να περικλείεται μεταξύ διπλών εισαγωγικών (`"`). Η λέξη `"George"` ή η φράση `"Hello world!"` είναι παραδείγματα τέτοιων αλφαριθμητικών. Για την αποθήκευση αλφαριθμητικών δεδομένων η συγκεκριμένη γλώσσα διαθέτει διάφορες κλάσεις. Οι πιο σημαντικές, οι οποίες θα μελετηθούν στη συνέχεια, είναι η `String` και η `StringBuffer`.

Στη Java μία μεταβλητή αλφαριθμητικού τύπου είναι αντικείμενο της κλάσης `String`. Με τον πιο άμεσο τρόπο δηλώνεται και παίρνει αρχική τιμή σύμφωνα με την πιο κάτω σύνταξη:

```
String stringVariableName;  
stringVariableName="str";
```

Όπου:

- `stringVariableName`: Το όνομα που είναι επιθυμητό να έχει η μεταβλητή που ορίζεται σε κάθε περίπτωση και στην οποία πρόκειται να αποθηκευτεί ένα αλφαριθμητικό.
- `str`: Η τιμή που καταχωρείται στη μεταβλητή `stringVariableName`.

Η παραπάνω σύνταξη έχει και πιο σύντομη μορφή, η οποία είναι η εξής:

```
String stringVariableName="str";
```

Επίσης, είναι δυνατόν απλά να δηλωθεί η μεταβλητή `stringVariableName` (με την εντολή `String stringVariableName;`) και να της αποδοθεί τιμή, αργότερα, σε κάποιο άλλο σημείο του προγράμματος που την περιέχει, πέρα, δηλαδή, από το σημείο που αυτή δηλώνεται.

Επιπρόσθετα, εκτός από μεταβλητές είναι δυνατό να δημιουργηθούν μεταβλητές που να κρατούν πίνακες, τα στοιχεία των οποίων είναι αλφαριθμητικά. Κάτι τέτοιο γίνεται με τη δημιουργία πινάκων τύπου `String` και είναι η ίδια διαδικασία που ακολουθείται για τη δημιουργία πινάκων οποιουδήποτε τύπου και η οποία μελετήθηκε διεξοδικά πιο πάνω.

Συνεχίζοντας τη συζήτηση για τα αλφαριθμητικά στη Java θα γίνει αναφορά στον τρόπο χειρισμό αυτών, με τη βοήθεια μεθόδων που παρέχει τόσο η κλάση `String` όσο και η κλάση `StringBuffer` καθώς και στη σημαντική διαφορά των συγκεκριμένων κλάσεων. Πριν, όμως, γίνει μνεία στις μεθόδους των προαναφερθέντων κλάσεων, αξίζει να σημειωθεί πως μία ενέργεια που απαντάται κατά κόρον στα προγράμματα, και δεν είναι άλλη από τη συνένωση δύο ή περισσότερων αλφαριθμητικών, στη Java επιτυγχάνεται με τη χρήση του τελεστή `+`.

Περνώντας, τώρα, στις πιο βασικές μεθόδους της κλάσης `String`, πρώτιστα, να τονιστεί πως

προκειμένου να εξεταστεί η ισότητα δύο αλφαριθμητικών χρησιμοποιείται η μέθοδος equals() της κλάσης String, αντί του τελεστή ==, που διατίθεται στη σύγκριση των βασικών τύπων δεδομένων της Java. Αν, μάλιστα, είναι αναγκαίο να μη λαμβάνεται υπόψιν το αν οι χαρακτήρες είναι κεφαλαίοι ή πεζοί, τότε χρησιμοποιείται η μέθοδος equalsIgnoreCase().

Η Java, όπως έχει ήδη αναφερθεί, κάνει διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων. Η κλάση String περιέχει μεθόδους με τις οποίες είναι εφικτό να μετατραπούν τα κεφαλαία γράμματα σε πεζά και αντίστροφα. Η μέθοδος toLowerCase(), μόλις εφαρμοστεί σε ένα αλφαριθμητικό, επιστρέφει ένα άλλο με τους ίδιους χαρακτήρες, αλλά με πεζά γράμματα. Η μέθοδος toUpperCase() εκτελεί την αντίστροφη μετατροπή.

Με τον όρο, τώρα, υποαλφαριθμητικό (substring) εννοείται ένα αλφαριθμητικό, το οποίο βρίσκεται μέσα σε ένα άλλο αλφαριθμητικό μεγαλύτερο ή ίσο από αυτό. Η κλάση String περιέχει την κύρια μέθοδο substring(int start, int end) για την εξαγωγή του υποαλφαριθμητικού, όπου το ακέραιο όρισμα start προσδιορίζει το δείκτη για τον πρώτο χαρακτήρα του υποαλφαριθμητικού, ενώ το ακέραιο όρισμα end προσδιορίζει το δείκτη της θέσης του αμέσως επόμενου χαρακτήρα μετά τον τελευταίο χαρακτήρα του υποαλφαριθμητικού (ο χαρακτήρας του αλφαριθμητικού που αντιστοιχεί στη θέση αυτή δεν περιλαμβάνεται).

Ύστερα, κατά την επεξεργασία ενός αλφαριθμητικού, πολλές φορές χρειάζεται να βρεθεί το μήκος του. Για το σκοπό αυτό διατίθεται η μέθοδος length() της κλάσης String.

Επίσης, συχνά εμφανίζεται η ανάγκη να προσπελαστούν μεμονωμένοι χαρακτήρες ενός αλφαριθμητικού. Για να γίνει κάτι τέτοιο, κυρίως διατίθεται μία μεταβλητή τύπου int, ώστε να προσδιοριστεί η θέση του χαρακτήρα, η οποία περνάει ως όρισμα στη μέθοδο charAt() της κλάσης String.

Ακόμα, με τη χρήση της μεθόδου replace() της κλάσης String είναι εφικτό να αντικατασταθούν κάποιοι χαρακτήρες ενός αλφαριθμητικού με άλλους. Η μέθοδος replace() δέχεται δύο ορίσματα: Το πρώτο είναι ο χαρακτήρας που πρόκειται να αντικατασταθεί, και το δεύτερο ο χαρακτήρας που θα πάρει τη θέση του. Να σημειωθεί πως αντικαθίστανται όλοι οι χαρακτήρες που είναι ίδιοι με το χαρακτήρα προς αντικατάσταση.

Επιπλέον, να παρατηρηθεί πως υπάρχουν δύο βασικές μέθοδοι της κλάσης String οι οποίες αναζητούν μέσα σε ένα αλφαριθμητικό κάποιο συγκεκριμένο χαρακτήρα. Οι μέθοδοι αυτές είναι η indexOf() και η lastIndexOf(). Η πρώτη θα ψάξει ένα αλφαριθμητικό από το τέλος προς την αρχή (δηλαδή από τα αριστερά προς τα δεξιά) και θα επιστρέφει το δείκτη της θέσης εκείνης, όπου θα συναντήσει για πρώτη φορά το χαρακτήρα που είναι επιθυμητό να εντοπιστεί. Αν ο χαρακτήρας που επιθυμείται να εντοπιστεί δε βρεθεί, τότε η μέθοδος θα επιστρέφει την τιμή -1. Η δεύτερη μέθοδος ερευνά ένα αλφαριθμητικό από το τέλος προς την αρχή (δηλαδή από τα δεξιά προς τα αριστερά) και όπου συναντήσει το ζητούμενο χαρακτήρα, θα επιστρέφει τη θέση του δείκτη του ή την τιμή -1, αν ο χαρακτήρας δε βρεθεί.

Συνάμα, στη Java υπάρχει η δυνατότητα να μετατραπούν διάφοροι τύποι δεδομένων σε αλφαριθμητικά με τη χρήση της μεθόδου valueOf(), περνώντας της κάθε φορά ως παράμετρο τα δεδομένα που πρόκειται να υποστούν μετατροπή.

Για την μετατροπή, τώρα, αλφαριθμητικών σε διάφορους τύπους δεδομένων η Java διαθέτει τις επόμενες μεθόδους:

- parseByte() της κλάσης Byte, για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε ακέραιο τύπου Byte.
- parseShort(), της κλάσης Short(), για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε ακέραιο τύπου Short.
- parseInt(), της κλάσης Integer, για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε ακέραιο τύπου Int.
- parseLong() της κλάσης Long, για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε ακέραιο τύπου Long.
- parseFloat() της κλάσης Float, για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε αριθμό κινητής υποδιαστολής.
- parseDouble() της κλάσης Double, για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε αριθμό κινητής υποδιαστολής διπλής ακρίβειας.
- parseBoolean() της κλάσης Boolean, για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε λογική τιμή.

- `parseChar()` της κλάσης `Character`, για τη μετατροπή του αλφαριθμητικού που δέχεται ως όρισμα σε χαρακτήρα.

Στο σημείο αυτό, να τονιστεί πως στην Java καθεμία από τις κλάσεις `Byte`, `Short`, `Integer`, `Long`, `Float`, `Double`, `Boolean` και `Character` ονομάζεται και κλάση συσκευαστής (wrapper class), επειδή εμπεριέχει το βασικό τύπο δεδομένων `byte`, `short`, `int`, `long`, `float`, `double`, `boolean` και `char`, αντίστοιχα. Ο σκοπός των κλάσεων-συσκευαστών είναι να παρέχουν μεθόδους, όπως αυτές που αναπτύχθηκαν ακριβώς παραπάνω.

Όπως διαπιστώνεται από την πιο πάνω ανάλυση, η κλάση `String` είναι μία από τις πιο χρήσιμες κλάσεις της Java, τα αντικείμενά της, όμως, δηλαδή τα αλφαριθμητικά, δεν μπορούν να αλλάξουν. Για να γίνει αυτό, υπάρχει μία άλλη κλάση, η κλάση `StringBuffer`, η οποία διατίθεται για αλφαριθμητικά που μεταβάλλονται. Τα αντικείμενα, λοιπόν, της κλάσης `StringBuffer` έχουν το χαρακτηριστικό ότι είναι σε θέση να κρατήσουν αχρησιμοποίητο χώρο για χαρακτήρες, ενώ τα αντικείμενα τύπου `String` δεν έχουν αυτή την ικανότητα.

Η κλάση `StringBuffer` διαθέτει και αυτή με τη σειρά της ένα πλήθος μεθόδων που συντελούν στο χειρισμό των αντικειμένων της. Στη συνέχεια επιχειρείται μία αναφορά στις βασικότερες μεθόδους της συγκεκριμένης κλάσης. Έτσι με τις μεθόδους `length()` και `capacity()` της κλάσης `StringBuffer` μπορεί κάποιος να υπολογίσει το χρησιμοποιούμενο μήκος και τη συνολική χωρητικότητα ενός κενού αντικειμένου τύπου `StringBuffer`, ενώ με τη χρήση της μεθόδου `append()` της κλάσης `StringBuffer` έχει τη δυνατότητα να προσθέσει τους χαρακτήρες που θέλει σε ένα άδειο αντικείμενο τύπου `StringBuffer`. Μάλιστα, νέα κλήση της μεθόδου `append()` θα προσθέσει νέους χαρακτήρες στους ήδη υπάρχοντες. Έπειτα, με τις μεθόδους `setCharAt()` και `insert()` της κλάσης `StringBuffer` είναι εφικτό να γίνουν προσθήκες στο αλφαριθμητικό περιεχόμενο ενός αντικειμένου τύπου `StringBuffer`. Επιπλέον, για τη μετατροπή ενός αντικειμένου `StringBuffer` σε αλφαριθμητικό εκμεταλλεύεται η μέθοδος `toString()` της κλάσης `StringBuffer`. Τέλος, η μέθοδος `reverse()` της κλάσης `StringBuffer` αντιστρέφει ένα αλφαριθμητικό, που έχει εκχωρηθεί σε ένα αντικείμενο τύπου `StringBuffer`.

[57]

2.16.7 Οι Βασικές Έννοιες του Αντικειμενοστραφούς Προγραμματισμού

2.16.7.1 Οι Κλάσεις και τα Αντικείμενα

Στην τρέχουσα ενότητα θα εξεταστεί η καρδιά της Java, δηλαδή τις κλάσεις και τα αντικείμενα, που δημιουργούνται από αυτές. Οι έννοιες κλάση (class) και αντικείμενο (object) είναι βασικές στον Αντικειμενοστραφή Προγραμματισμό (Object Oriented Programming-OOP) και μία πρώτη εικόνα διαμορφώθηκε στην αντίστοιχη ενότητα της JavaScript.

Η κλάση είναι μία γενική φόρμα για την κατασκευή αντικειμένων που έχουν παρόμοια χαρακτηριστικά. Με άλλα λόγια είναι μία «συνταγή» δημιουργίας αντικειμένων. Αφού, λοιπόν, οριστεί μία κλάση, στη συνέχεια μπορεί να γίνει χρήση του ορισμού της (της «συνταγής») για τη δημιουργία αντικειμένων αυτής της κλάσης. Κάθε αντικείμενο λέγεται και στιγμιότυπο (instance) της κλάσης.

Παίρνοντας το πρώτο παράδειγμα από τη ζωή μας, θα μπορούσε να θεωρηθεί ως κλάση η έννοια δένδρο και ως αντικείμενα της κλάσης είτε η λεμονιά, είτε η μηλιά, είτε το έλατο. Όλα τα αντικείμενα έχουν τα γενικά χαρακτηριστικά της κλάσης δένδρο (ρίζα, κορμό, κλαδιά, φύλλα), αλλά το καθένα έχει και κάποιες ιδιαίτερες ιδιότητες, όπως το είδος του καρπού, το σχήμα, το αν είναι φυλλοβόλο ή όχι.

Ας αναφερθεί και ένα ακόμα παράδειγμα, πιο κοντινό στην Java: Συχνά στις εφαρμογές της Java απαιτείται η δημιουργία κουμπιών (buttons). Στην Java υπάρχει η κλάση `JButton`, η οποία καθορίζει τα γενικά χαρακτηριστικά που πρέπει να έχει ένα κουμπί, δηλαδή το μέγεθος του, την ετικέτα του, τον τρόπο με τον οποίο θα ανταποκρίνεται (απλό ή διπλό πάτημα). Με βάση την κλάση `JButton` μπορούμε να δημιουργήσουμε αντικείμενα (συγκεκριμένα κουμπιά), τα οποία να έχουν όλα τα χαρακτηριστικά και τις ιδιότητες που περιέχονται στη `JButton`, αλλά επιπλέον, το καθένα από αυτά να μπορεί να έχει και τα δικά του ιδιαίτερα χαρακτηριστικά και ιδιότητες.

Στην Java υπάρχουν έτοιμες κλάσεις, ωστόσο, μπορεί και ο ίδιος ο χρήστης να δημιουργήσει τις δικές του. Προκειμένου να ορίσει οριστεί μία κλάση, γράφεται η λέξη `class` ακολουθούμενη από το όνομα που επιλέγεται για την κλάση. Η πιο απλή κλάση της Java θα μπορούσε να υλοποιηθεί σύμφωνα με την πιο κάτω σύνταξη:

```
class ClassName
{
}

```

Όπου `ClassName` το όνομα της Java κλάσης που ορίζεται σε κάθε περίπτωση.

Η προηγούμενη κλάση είναι μία ολοκληρωμένη κλάση της Java. Δύναται να αποθηκευτεί με όνομα `ClassName.java` και να τη μεταγλωττιστεί δημιουργώντας το αρχείο `ClassName.class`. Το αρχείο αυτό βέβαια δεν κάνει τίποτα, γιατί η κλάση `ClassName` δεν ορίστηκε να κάνει κάτι (εφόσον το σώμα-κορμός της κλάσης, δηλαδή το τμήμα της κλάσης εντός του ζεύγους των αγκίστρων `{}` είναι κενό. Αν είναι αναγκαίο μία κλάση να επιτελεί ένα σκοπό, όπως, εξάλλου, σχεδόν πάντα απαιτείται, τότε συμπεριλαμβάνονται στον ορισμό της τα δύο κύρια συστατικά μίας κλάσης, τα οποία είναι τα ακόλουθα:

- I. Οι μεταβλητές (*variables*): Διαφοροποιούν το ένα αντικείμενο μία κλάσης από το άλλο. Οι μεταβλητές μίας κλάσης χωρίζονται σε δύο κατηγορίες, που είναι οι επόμενες:
 - A. Οι μεταβλητές κλάσης (*class variables*): Υπάρχουν, όπως έχει ήδη αναφερθεί παραπάνω, έστω και αν δεν έχει ακόμα δημιουργηθεί κανένα αντικείμενο της τρέχουσας σε κάθε περίπτωση κλάσης. Κάθε αντικείμενο της κλάσης, όταν δημιουργηθεί, περιέχει και τις τιμές των μεταβλητών αυτής της κατηγορίας. Αν οι τιμές αυτές αλλάζουν, τότε οι νέες τιμές τοποθετούνται σε όλα τα αντικείμενα της κλάσης. Δηλώνονται με τη δεσμευμένη λέξη `static`.
 - B. Οι μεταβλητές στιγμιότυπου (*instance variables*): Υφίστανται μόνο, όπως έχει ήδη εξηγηθεί πρωτύτερα, εφόσον έχει δημιουργηθεί κάποιο αντικείμενο της τρέχουσας σε κάθε περίπτωση κλάσης και έχουν διαφορετική τιμή από αντικείμενο σε αντικείμενο, αντίθετα από τις μεταβλητές κλάσης. Δηλώνονται με το συνήθη τρόπο, δηλαδή με το είδος του τύπου και το όνομά της.
- II. Οι μέθοδοι (*methods*): Καθορίζουν τι μπορεί να γίνει στα (ή με τα) αντικείμενα μίας κλάσης. Πρόκειται ουσιαστικά για μία ομάδα από δηλώσεις και εντολές, η οποία εκτελεί μία εξειδικευμένη λειτουργία ενός αντικειμένου. Οι μέθοδοι μίας κλάσης εμφανίζουν ανάλογη συμπεριφορά με τις μεταβλητές αυτής, που μόλις μελετήθηκαν. Έτσι κατατάσσονται σε δύο είδη, που είναι τα μετέπειτα:
 - A. Οι μέθοδοι κλάσης (*class methods*): Εκτελούνται ακόμα και αν δεν έχει δημιουργηθεί κανένα αντικείμενο της τρέχουσας σε κάθε φάση κλάσης. Επομένως, δεν αναφέρονται σε μεταβλητές στιγμιότυπου. Δηλώνονται με τη δεσμευμένη λέξη `static`.
 - B. Οι μέθοδοι στιγμιότυπου (*instance methods*): Εκτελούνται μόνο αν υπάρχουν αντικείμενα της τρέχουσας σε κάθε χρονική στιγμή κλάσης, σε αντίθεση με τις μεθόδους κλάσης. Αν δεν υπάρχουν αντικείμενα δεν έχει νόημα να εκτελεστεί μία τέτοια μέθοδος. Κάθε μέθοδος στιγμιότυπου, τώρα, είναι σε θέση να αναφέρεται στο εκάστοτε τρέχον αντικείμενο για το οποίο έχει κληθεί η συγκεκριμένη μέθοδος, με τη χρήση της λέξης `this`.

Στο σημείο τούτο να τονιστεί για να πραγματοποιηθεί αναφορά στις μεταβλητές και τις μεθόδους που περιέχει ένα αντικείμενο μίας κλάσης χρησιμοποιείται ο τελεστής της τελείας (`.`)

Επίσης, θα ήταν μεγάλη παράλειψη να μην τονιστεί η έννοια του κατασκευαστή μίας κλάσης της Java. Με τον όρο κατασκευαστής (*constructor*), λοιπόν, εννοείται μία ειδική μέθοδος που πρέπει να υπάρχει στον ορισμό μίας κλάσης. Έχει το ίδιο όνομα με εκείνο της κλάσης και χρησιμοποιείται, για να δώσει αρχικές τιμές στις μεταβλητές ενός νέου αντικειμένου και να κάνει όποιες άλλες ενέργειες χρειάζεται, ώστε να προετοιμαστεί σωστά το νέο αντικείμενο. Επιπλέον, κάθε κατασκευαστής μπορεί να έχει όσες παραμέτρους είναι επιθυμητό, ενδεχομένως, όμως να μην έχει και καμία.

Ακόμα, υπάρχει η ευχέρεια να διατίθενται περισσότεροι από ένας κατασκευαστές για την ίδια κλάση, κάτι που προκύπτει από την ανάγκη να υλοποιούνται αντικείμενα με ξεχωριστά χαρακτηριστικά. Ωστόσο οι διαφορετικοί κατασκευαστές έχουν απαραίτητα το ίδιο όνομα, αλλά διαφορετική λίστα παραμέτρων. Επιπλέον, είναι εφικτή και η περίπτωση να μην έχει δημιουργηθεί κανένας κατασκευαστής για μία κλάση της Java, οπότε και η γλώσσα δημιουργεί έναν από μόνη της, ο οποίος, ωστόσο, δεν κάνει τίποτα.

Εδώ, να παρατηρηθεί ότι η κατάσταση στην οποία μπορεί να υφίστανται περισσότεροι του ενός κατασκευαστές με το ίδιο όνομα, αλλά με τα δικά του ιδιαίτερα γνωρίσματα ο καθένας, ονομάζεται υπερφόρτωση κατασκευαστή (constructor overloading). Το όνομα του κάθε κατασκευαστή σε συνδυασμό με τον αριθμό, τον τύπο και τη σειρά των ορισμάτων του αποτελούν την υπογραφή του (signature).

Ωστόσο, το φαινόμενο της υπερφόρτωσης ισχύει και στις μεθόδους (method overloading). Υπάρχει, με άλλα λόγια, η δυνατότητα δημιουργίας μεθόδων με το ίδιο όνομα, αλλά με διαφορετικό ορισμό και άλλα ορίσματα, άρα με διαφορετική υπογραφή. Η Java έχει τη δυνατότητα να τις ξεχωρίζει και κάθε φορά να εκτελεί τη μέθοδο που πρέπει από τον αριθμό και τον τύπο των ορισμάτων τους.

Κλείνοντας, να δοθεί ιδιαίτερη προσοχή στο γεγονός ότι κάθε κατασκευαστής έχει δύο χαρακτηριστικά που τον διαφοροποιούν από τις άλλες μεθόδους και είναι τα επόμενα:

- I. Έχει το ίδιο όνομα με την κλάση.
- II. Δεν επιστρέφει ποτέ τιμή, συνεπώς δεν τοποθετείται ποτέ μπροστά του τύπος επιστροφής, όπως συμβαίνει με τις υπόλοιπες μεθόδους, αλλά, ούτε και η λέξη void.

[55] [56]

2.16.7.2 Τα Πακέτα

Ένα πακέτο (package) της Java είναι μία συλλογή από κλάσεις της γλώσσας, οι οποίες μπορούν να προσθέσουν δυνατότητες σε αυτήν. Ο κώδικας που βρίσκεται σε ένα πακέτο μπορεί να ξαναχρησιμοποιηθεί σε διαφορετικά προγράμματα. Με άλλα λόγια, το πακέτο είναι μία βιβλιοθήκη κώδικα. Κάθε κλάση στην Java πρέπει να περιέχεται σε ένα πακέτο.

Για να χρησιμοποιηθεί μία μέθοδος ή μία κλάση ενός πακέτου, υπάρχουν δύο τρόποι, που είναι οι εξής:

- I. Ο πρώτος είναι η εκμετάλλευση του πλήρους ονόματος του πακέτου κατά τον ορισμό του αντικειμένου της κλάσης. Για παράδειγμα, σύμφωνα με την ακόλουθη σύνταξη:

```
java.util.Date thisDate = new java.util.Date();
```

χρησιμοποιείται η κλάση Date, που βρίσκεται στο πακέτο util και δημιουργείται ένα αντικείμενο αυτής της κλάσης (το thisDate).

- II. Ο δεύτερος τρόπος είναι η εισαγωγή της κλάσης στην αρχή του προγράμματος με τη λέξη import και κατόπιν η εκμετάλλευσή της μέσα στο πρόγραμμα, γράφοντας μόνο το όνομά της. Έτσι η σύνταξη του αμέσως παραπάνω παραδείγματος, τροποποιείται όπως στη συνέχεια:

```
import java.util.Date;  
.....  
Date thisDate=new Date();
```

Πολλές φορές, όμως, αντί να εισαχθεί μία κλάση, είναι καλύτερα να εισαχθεί ολόκληρο το πακέτο στο οποίο περιέχεται αυτή η κλάση, χρησιμοποιώντας έναν αστερίσκο (*). Με αυτό τον τρόπο είναι διαθέσιμες όλες οι κλάσεις του συγκεκριμένου πακέτου. Έτσι τα παραπάνω αποσπάσματα κώδικα θα τροποποιούνταν ελαφρώς ως εξής:

```
java.util.* thisDate = new java.util.Date();
```

ή

```
import java.util.*;  
.....  
Date thisDate=new Date();
```

Συνεχίζοντας τη συζήτηση για τα πακέτα, αξίζει να σημειωθεί ότι, από τη στιγμή που η γλώσσα Java, τώρα, διαθέτει σχετικά ένα μικρό αριθμό δεσμευμένων λέξεων, οφείλει τη δύναμή της στα έτοιμα πακέτα που διαθέτει. Όπως προαναφέρθηκε, ένα πακέτο είναι μία συλλογή από κλάσεις και μεθόδους

τις οποίες ο εκάστοτε προγραμματιστής μπορεί να χρησιμοποιήσει σαν να αποτελούσαν μέρος της γλώσσας. Η βιβλιοθήκη αυτή των κλάσεων της Java αναφέρεται ως Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface-API), και περιλαμβάνει πάρα πολλά πακέτα τα οποία αυξάνονται σε κάθε νέα έκδοση της γλώσσας. Μερικά από τα βασικότερα πακέτα της Java παρουσιάζονται στον κατώτερο Πίνακα:

Πακέτο	Περιγραφή
java.lang	Περιέχει τις βασικές κλάσεις της Java. Όλες οι κλάσεις του πακέτου είναι διαθέσιμες αυτόματα στον εκάστοτε προγραμματιστή και δε χρειάζεται να χρησιμοποιηθεί η λέξη import.
java.io	Περιέχει κλάσεις για την είσοδο και την έξοδο δεδομένων.
java.awt	Περιέχει κλάσεις για το γραφικό περιβάλλον της Java.
java.awt.event	Περιέχει κλάσεις για το χειρισμό γεγονότων.
java.applet	Περιέχει κλάσεις για τη συγγραφή προγραμμάτων που ενσωματώνονται στις Ιστοσελίδες.
java.util	Περιέχει κλάσεις για την επιτέλεση μερικών βασικών εργασιών σε δεδομένα, το χειρισμό ημερομηνιών καθώς και το χειρισμό αλφαριθμητικών.
java.awt.peer	Περιέχει κλάσεις που καθορίζουν το μέσον μετασχηματισμού των κλήσεων των μεθόδων του γραφικού περιβάλλοντος της Java προς τις φυσικές τους υλοποιήσεις για μία συγκεκριμένη πλατφόρμα.
java.security	Περιέχει κλάσεις για την αύξηση της ασφάλειας εντός των Java προγραμμάτων, με τη χρήση μεθόδων κρυπτογραφίας.
java.awt.image	Περιέχει κλάσεις για την επεξεργασία εικόνας.
java.net	Περιέχει κλάσεις για λειτουργίες δικτύου.
java.sql	Περιέχει κλάσεις για το χειρισμό των λειτουργιών μίας Βάσης Δεδομένων, μέσω εντολών SQL.
java.beans	Περιέχει κλάσεις για τη δημιουργία αυτοτελών τμημάτων Java κώδικα, τα οποία μπορούν να ενσωματωθούν σε μία εφαρμογή.
java.rmi	Περιέχει κλάσεις για την επίκληση απομακρυσμένων μεθόδων.
java.swing	Περιέχει κλάσεις και διασυνδέσεις για την υλοποίηση των συστατικών του νέου περιβάλλοντος επικοινωνίας με το χρήστη, που αναφέρεται ως Swing.

Πίνακας 28: Οι Τα πιο βασικά τυπικά πακέτα της Java

Κλείνοντας την κουβέντα για τα πακέτα της Java να επισημανθεί πως ο εκάστοτε προγραμματιστής της Java έχει τη δυνατότητα να πακετάρει τις δικές του συσχετιζόμενες κλάσεις, για να βοηθήσει κατά τη συγγραφή του εκάστοτε κώδικα. Για να προσθέσει, τώρα μία κλάση σε ένα πακέτο, προτού ακόμα την ορίσει πρέπει να ακολουθήσει την επόμενη σύνταξη:

package packetName;

Όπου packetName το όνομα του εκάστοτε πακέτου που επιθυμεί να δημιουργήσει ο εκάστοτε προγραμματιστής της Java.

[55] [56] [57]

2.16.7.3 Οι Αρχές του Αντικειμενοστραφούς Προγραμματισμού

Τη βασική ιδέα του αντικειμενοστραφούς προγραμματισμού υλοποιούν οι παρακάτω τρεις αρχές:

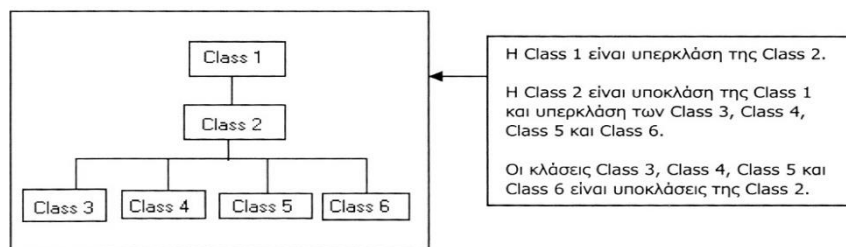
Πρώτον, η ενθυλάκωση (encapsulation): Παρουσιάζει την εξωτερική όψη ενός αντικειμένου

και αποκρύπτει την πολύπλοκη λογική που υπάρχει μέσα του. Αυτό επιτρέπει στον εκάστοτε προγραμματιστή να τροποποιεί ή να βελτιώνει τους αλγορίθμους, οι οποίοι υπάρχουν σε μία κλάση, χωρίς να ανησυχεί μήπως προκαλέσει απρόβλεπτα αποτελέσματα. Με άλλα λόγια, τη στιγμή που σε μία κλάση τα δεδομένα και οι χρησιμοποιούμενες μέθοδοι δεν αλλάζουν, κάθε αλλαγή στον τρόπο υλοποίησής τους παραμένει διαφανής για το χρήστη. Έτσι τα δεδομένα και οι μέθοδοι μίας κλάσης μπορούν να δηλωθούν είτε ως public, είτε ως private, είτε ως protected, είτε χωρίς κανέναν από τους προηγούμενους τρεις προσδιοριστές προσπέλασης (access modifiers). Η δυνατότητα προσπέλασης για καθεμιά από τις προηγούμενες κατηγορίες φαίνεται στον Πίνακα που έπεται:

Δυνατότητα προσπέλασης	Χωρίς προσδιοριστή	Public	Private	Protected
Από την ίδια την κλάση	Ναι	Ναι	Ναι	Ναι
Από τις κλάσεις του ίδιου πακέτου.	Ναι	Ναι	Όχι	Όχι
Από οποιαδήποτε κλάση εκτός του πακέτου όπου ανήκει η κλάση.	Όχι	Ναι	Όχι	Όχι
Από μία υποκλάση του ίδιου πακέτου.	Ναι	Ναι	Όχι	Όχι
Από μία υποκλάση έξω από το πακέτο όπου ανήκει η κλάση.	Όχι	Ναι	Όχι	Όχι

Πίνακας 29: Η δυνατότητα προσπέλασης των προσδιοριστών πρόσβασης της Java

Δεύτερον, η κληρονομικότητα (inheritance), που συνιστά μία από τις βασικές ιδέες και είναι ένα από τα σημαντικότερα πλεονεκτήματα των αντικειμενοστραφών γλωσσών σε σχέση με τις συμβατικές γλώσσες, διότι χάρη σε αυτήν επιτυγχάνεται η επαναχρησιμοποίηση κώδικα, ο οποίος έχει γραφεί πριν. Σύμφωνα, λοιπόν, με την ιδέα αυτή ένα αντικείμενο δύναται να κληρονομεί τα δεδομένα και τις μεθόδους ενός άλλου αντικειμένου. Μπορεί, επίσης, να τους επιφέρει τροποποιήσεις. Στην Java όλες οι κλάσεις (οι διαθέσιμες από τη γλώσσα καθώς και αυτές που μπορούν να δημιουργηθούν από τον εκάστοτε προγραμματιστή) ακολουθούν μία ιεραρχία. Κάθε κλάση έχει μία υπερκλάση (superclass), η οποία βρίσκεται ιεραρχικά πάνω από αυτή. Μπορεί, συνάμα, να έχει μία ή περισσότερες υποκλάσεις, οι οποίες βρίσκονται σε κατώτερο επίπεδο ιεραρχίας, όπως φαίνεται στην πιο κάτω Εικόνα:



Εικόνα 23: Η κληρονομικότητα των κλάσεων της Java

Οι υποκλάσεις κληρονομούν αυτόματα όλες τις μεταβλητές και τις μεθόδους από τις υπερκλάσεις τους. Η Java παρέχει τη δεσμευμένη λέξη super για την κλήση μελών της υπερκλάσης μίας κλάσης.

Μάλιστα μία υποκλάση καλεί τον κατάλληλο σε κάθε περίπτωση κατασκευαστή μίας υπερκλάσης σύμφωνα με τη σύνταξη:

```
super();
```

Όταν, όμως, επιθυμεί να κάνει κλήση μίας συγκεκριμένης μεθόδου, εκτός του κατασκευαστή, τότε χρησιμοποιείται η σύνταξη:

```
super.methodName();
```

Όπου methodName το όνομα της εκάστοτε επιθυμητής μεθόδου μίας υπερκλάσης μίας κλάσης της Java,

την οποία η κλάση χρειάζεται να καλέσει εντός ενός Java κώδικα.

Στην κορυφή, τώρα, όλων των κλάσεων της Java βρίσκεται η κλάση `Object`. Όλες οι κλάσεις κληρονομούν από αυτή την υπερκλάση. Έτσι, όταν ορίζεται μία υπερκλάση για την οποία δεν υπάρχει ορισμός υπερκλάσης, η Java θεωρεί αυτόματα πως η οριζόμενη κλάση είναι υποκλάση της `Object`.

Επιπρόσθετα, να τονιστεί ότι στην κληρονομικότητα συχνά εμφανίζεται το φαινόμενο της παράκαμψης μεθόδου (*method overriding*) (το οποίο στα ελληνικά αποδίδεται πολλές φορές και ως ακύρωση μεθόδου). Σύμφωνα με αυτό το φαινόμενο, λοιπόν, σε μία υποκλάση ορίζεται ξανά μία μέθοδος, παρόλο που είχε οριστεί και σε μία κλάση ανωτέρου επιπέδου, επειδή ο αρχικός ορισμός της δεν εξυπηρετεί ακριβώς αυτό που θέλει ο εκάστοτε προγραμματιστής. Η μέθοδος στην υποκλάση έχει την ίδια υπογραφή με αυτή της υπερκλάσης (όνομα αριθμό και τύπο ορισμάτων). Κατά κάποιον τρόπο, δηλαδή, η νέα μέθοδος καταργεί τη δράση της αρχικής.

Κλείνοντας να σημειωθεί πως η έννοια της κληρονομικότητας πολλών επιπέδων είναι κάτι διαφορετικό από την έννοια της πολλαπλής κληρονομικότητας (*multiple inheritance*), που απαντάται σε άλλες γλώσσες (πχ C), και η οποία αναφέρεται στη δυνατότητα μίας κλάσης να κληρονομεί από διαφορετικές και μη συσχετιζόμενες υπερκλάσεις.

Τρίτον, ο πολυμορφισμός (*polymorphism*), ο οποίος παρατηρείται στην περίπτωση που η ίδια μέθοδος ή ο ίδιος τελεστής χρησιμοποιείται με διαφορετικούς τύπους δεδομένων. Οπότε, για παράδειγμα, μπορεί να υπάρχουν περισσότεροι από ένας κατασκευαστές μίας κλάσης, που να φέρουν το ίδιο όνομα, αλλά διαφορετικά ορίσματα, όπως μελετήθηκε παραπάνω, με συνέπεια την κατασκευή διαφορετικών αντικειμένων της κλάσης, ή ο τελεστής της πρόσθεσης (+) δύναται να διατίθεται τόσο για την πρόσθεση δύο αριθμών όσο και για τη συνένωση δύο αλφαριθμητικών.

[55] [56] [57]

2.16.7.4 Οι Τελικές Κλάσεις και Μέθοδοι

Σε περιπτώσεις κατά τις οποίες είναι επιθυμητό μία κλάση να δημιουργεί υποκλάσεις, δηλώνεται η κλάση αυτή ως τελική (*final*). Μία κλάση που έχει δηλωθεί ως τελική, δεν μπορεί να είναι υπερκλάση. Επίσης, υπάρχει η δυνατότητα μία μέθοδος να δηλωθεί ως τελική, προκειμένου να μην μπορεί η συγκεκριμένη μέθοδος να ακυρωθεί. Τέλος, η δήλωση *final* είναι εφικτό να χαρακτηρίζει και μία μεταβλητή καθιστώντας την με αυτό τον τρόπο σταθερά.

[55]

2.16.7.5 Οι Αφηρημένου Τύπου Κλάσεις και Μέθοδοι

Στην Java υπάρχει η δυνατότητα να δηλωθεί μία κλάση ή μία μέθοδος ως αφηρημένη (*abstract*). Μία μέθοδος αφηρημένου τύπου σε μία κλάση απλώς ορίζεται, χωρίς να υλοποιείται. Το κύριο μέρος της μεθόδου αυτής υλοποιείται σε μία υποκλάση της αρχικής κλάσης (ως αρχική εννοείται η κλάση στην οποία η μέθοδος ορίστηκε ως αφηρημένη). Να τονιστεί, εδώ, ότι η δήλωση της μεθόδου τελειώνει με το σύμβολο του ελληνικού ερωτηματικού (;). Επιπρόσθετα, για να οριστεί μία κλάση ως αφηρημένη χρησιμοποιείται και πάλι η λέξη *abstract*.

[55]

2.16.7.6 Οι Διασυνδέσεις

Η έννοια της διασύνδεσης (*interface*) είναι κατά κάποιον τρόπο η απάντηση της Java στην αρχή της πολλαπλής κληρονομικότητας, η οποία απαντάται σε άλλες γλώσσες, όπως είναι, για παράδειγμα, η C. Μία διασύνδεση παρουσιάζει πολλές ομοιότητες με μία συνηθισμένη κλάση, με τη σημαντική διαφορά ότι καμία από τις μεθόδους που ορίζονται μέσα στη διασύνδεση δεν υλοποιείται. Αντί για αυτό, κάθε τέτοια μέθοδος πρέπει να υλοποιείται μέσα σε μία άλλη κλάση, η οποία καλεί τη διασύνδεση. Η κλήση γίνεται με τη λέξη *implements* μαζί με το όνομα της διασύνδεσης. Η διασύνδεση είναι, επίσης, μία κατάλληλη περιοχή για να δηλώνονται σταθερές. Συνοψίζοντας, μπορεί να ειπωθεί ότι η

διασύνδεση είναι ο χώρος όπου δύναται να τοποθετηθούν μία συλλογή από σταθερές καθώς και αφηρημένες μεθόδους, οι οποίες μπορούν να χρησιμοποιηθούν σε μία άλλη κλάση, γράφοντας εκεί τον κώδικα για κάθε μέθοδο. Για να οριστεί μία διασύνδεση, ακολουθείται όμοια διαδικασία με αυτή που ακολουθείται, για να οριστεί μία κλάση, αλλά αντί για τη λέξη class γίνεται χρήση της λέξης interface.

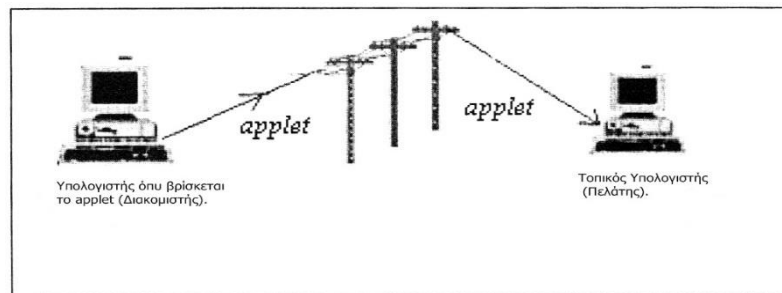
Κάπου εδώ, αξίζει να σημειωθεί πως σε μία κλάση που υλοποιεί τις μεθόδους μίας διασύνδεσης είναι δυνατό να παραλειφθούν μία ή περισσότερες από τις μεθόδους αυτές, αλλά τότε πρέπει η συγκεκριμένη κλάση να δηλωθεί ως αφηρημένη.

[55]

2.16.8 Τα Applets της Java

Οι δομές και οι εντολές της Java που εξετάστηκαν ως τώρα δε διαφέρουν και πολύ από τις αντίστοιχες μίας άλλης αντικειμενοστρεφούς γλώσσας προγραμματισμού, όπως είναι, για παράδειγμα είναι η C++. Αυτό όμως που κάνει την Java να ξεχωρίζει από τις άλλες γλώσσες είναι πως αυτήν παρέχει τη δυνατότητα για τη συγγραφή μικροεφαρμογών. Οι μικροεφαρμογές αυτές (applets) είναι κώδικας γραμμένος σε Java που εκτελείται μέσα από ένα πρόγραμμα περιήγησης του Διαδικτύου. Τα applets δεν είναι το μόνο είδος προγραμμάτων που μπορούν να παραχθούν με την Java, αλλά μόνο η Java είναι η κατάλληλη γλώσσα για να υλοποιήσει ένα applet. Τα applets δε μπορούν να τρέχουν απευθείας από το λειτουργικό σύστημα, αλλά εκτελούνται με τη βοήθεια HTML αρχείων μέσα από ένα πρόγραμμα περιήγησης.

Τα applets φτιάχνονται κυρίως για να εκτελούνται στον Παγκόσμιο Ιστό. Μέσα στις Ιστοσελίδες του Διαδικτύου, υπάρχουν άφθονα applets τα οποία κατεβαίνουν από τον υπολογιστή όπου βρίσκονται στο εκάστοτε τοπικό υπολογιστικό σύστημα. Η προαναφερθείσα διαδικασία φαίνεται στην πιο κάτω Εικόνα:



Εικόνα 24: Η θέση των applet της Java

Όταν ένας χρήστης του Διαδικτύου μπαίνει σε έναν άγνωστο για αυτόν Ιστότοπο, έχει το φόβο πως κάποιος ίσως μπορεί να προσβάλει τον υπολογιστή του. Έχοντας αυτό κατά νου η εταιρεία Sun, έβαλε συγκεκριμένους περιορισμούς στις δυνατότητες που μπορεί να έχει ένα applet, ώστε να είναι ασφαλές στο περιβάλλον στο οποίο εκτελείται. Έτσι ένα applet:

- Δε μπορεί να έχει πρόσβαση σε αρχεία του τοπικού υπολογιστή του εκάστοτε χρήστη.
- Δε μπορεί να καλέσει κανένα άλλο πρόγραμμα που υπάρχει στον τοπικό υπολογιστή του εκάστοτε χρήστη.
- Δε μπορεί να επικοινωνήσει με άλλον υπολογιστή εκτός από εκείνον που περιέχει την HTML Ιστοσελίδα, η οποία φορτώνεται στον τοπικό υπολογιστή του εκάστοτε χρήστη.

Από τα παραπάνω είναι φανερό ότι ένα applet που έχει φορτωθεί στον τοπικό υπολογιστή ενός χρήστη του Διαδικτύου, δεν είναι σε θέση να γράψει ή να διαβάσει δεδομένα από το σκληρό δίσκο του χρήστη, να σβήσει κάποια αρχεία, να εισάγει κάποιον ιό ή γενικά να μπλοκάρει το σύστημά του χρήστη.

Όσον αφορά, τώρα, τις πιο χρήσιμες μεθόδους που λαμβάνουν χώρα σε ένα applet, είναι οι κατώτερες:

- `public void init():` Καλείται από το εκάστοτε πρόγραμμα περιήγησης, όταν φορτώνεται ένα applet και αρχίζει η εκτέλεσή του. Αντιστοιχεί στη μέθοδο `main()`, η οποία χρησιμοποιείται στις

εφαρμογές προγραμματισμού (applicatio programming). Με τη μέθοδο αυτή στην ουσία γίνεται η απαραίτητη προετοιμασία. Αυτή μπορεί να περιλαμβάνει απόδοση αρχικών τιμών σε μεταβλητές, ανάγνωση και πέρασμα παραμέτρων στο applet καθώς και φόρτωση εικόνων ή γραμματοσειρών.

- `public void start()`: Καλείται αμέσως μετά με τη μέθοδο `init()`, καθώς μετά την προετοιμασία ένα applet ξεκινά. Το ξεκίνημα διαφέρει από την προετοιμασία, γιατί μπορεί να επαναλαμβάνεται πολλές φορές όσο το applet είναι ανοικτό, ενώ η προετοιμασία γίνεται μόνο μία φορά. Ξεκίνημα μπορεί, επίσης, να υφίσταται και αφού το applet έχει σταματήσει. Ένα applet σταματά, για παράδειγμα, όταν ο εκάστοτε χρήστης μεταβαίνει σε κάποια άλλη Σελίδα, ακολουθώντας έναν σύνδεσμο της εκάστοτε τρέχουσας Ιστοσελίδας, και ξαναρχίζει, αφότου ο χρήστης επιστρέψει πάλι στην προηγούμενη-τρέχουσα Σελίδα.
- `public void stop()`: Καλείται, όταν ο εκάστοτε χρήστης εγκαταλείπει τον Ιστοχώρο στον οποίο βρίσκεται ένα applet: Επιπρόσθετα, χρησιμοποιείται, η συγκεκριμένη μέθοδος, εφόσον είναι επιθυμητό να σταματήσουν κάποιες εργασίες που έχουν καθοριστεί στη μέθοδο `start()`.
- `public void destroy()`: Καλείται μετά τη μέθοδο `stop()`, μόλις κλείσει το εκάστοτε πρόγραμμα ανάγνωσης Ιστοσελίδων. Με αυτή τη μέθοδο δύναται να απελευθερωθούν κάποιοι πόροι που είναι δεσμευμένοι για όση διάρκεια τρέχει ένα applet. Συνήθως δεν πραγματοποιείται επέμβαση στη μέθοδο αυτή, παρά μόνο σε ειδικές περιπτώσεις.
- `public void paint()`: Καλείται κάθε φορά που χρειάζεται να σχεδιαστεί κάτι στην οθόνη του εκάστοτε χρήστη (πχ γράμματα, σχήματα, εικόνες, χρώματα) και μπορεί να κληθεί πάρα πολλές φορές κατά τη διάρκεια ζωής ενός applet. Αυτό μπορεί να συμβεί, για παράδειγμα, καθώς το παράθυρο ενός προγράμματος περιήγησης καλυφθεί από το παράθυρο ενός άλλου προγράμματος, αλλά κατόπιν επανέλθει στο προσκήνιο. Η παραπάνω μέθοδος παρακάμπτεται, από τη στιγμή που είναι επιθυμητό το παράθυρο του applet να έχει μία συγκεκριμένη εμφάνιση, πράγμα που συμβαίνει στις περισσότερες περιπτώσεις. Η `paint()` είναι η μόνη από τις παραπάνω μεθόδους που έχει όρισμα, ένα αντικείμενο της κλάσης `Graphics`.

[55] [56]

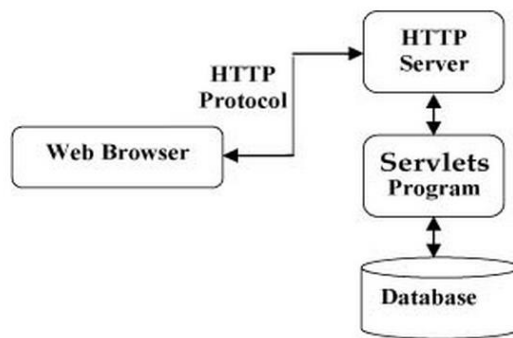
2.16.9 Οι Μικροεφαρμογές Διακομιστή Ιστού, Java Servlets

2.16.9.1 Εισαγωγή στις Java Servlets

Με τον όρο Java Servlet (μικροεφαρμογή διακομιστή Ιστού) εννοείται μία εφαρμογή (application) ή ένα σενάριο (script) γραμμένο σε γλώσσα Java το οποίο εκτελείται σε έναν εξυπηρετητή Ιστού (Web server) σε αντίθεση με τα προγράμματα applets που εκτελούνται σε υπολογιστές-πελάτες (hosts). Μία Servlet μπορεί να κάνει δουλειά ανάλογη με ένα σενάριο CGI.

Το Common Gateway Interface (CGI) είναι ένα πρωτόκολλο που επιτρέπει στους υπολογιστές-πελάτες χρησιμοποιώντας ένα πρόγραμμα ανάγνωσης Ιστοσελίδων να στέλνουν και να λαμβάνουν πληροφορίες από τους υπολογιστές-εξυπηρετητές με τη βοήθεια προγραμμάτων εγκατεστημένων στους εξυπηρετητές. Το CGI παρέχει μία ομάδα από μεθόδους οι οποίες μπορούν να δεχθούν πληροφορίες προερχόμενες από το πρόγραμμα περιήγησης. Μόλις ο εξυπηρετητής λάβει τις πληροφορίες αυτές, τις επεξεργάζεται με τη βοήθεια ενός προγράμματος γραμμένου συνήθως σε γλώσσα Perl. Το αποτέλεσμα είναι να δημιουργείται κώδικας HTML ο οποίος διοχετεύεται στο πρόγραμμα περιήγησης του πελάτη σαν να ήταν ένα κανονικό αρχείο .html εγκατεστημένο στον εξυπηρετητή.

Οι Java Servlets εκτελούν παρόμοια εργασία, όπως φαίνεται και στην παρακάτω Εικόνα: Είναι εγκατεστημένες στον εξυπηρετητή, δέχονται δεδομένα μέσω του πρωτοκόλλου HTTP και απαντούν στέλνοντας στο πρόγραμμα περιήγησης αρχεία τύπου .html. Επιπλέον, οι Servlets θεωρούνται κάτι πολύ περισσότερο από τα απλά σενάρια CGI. Οι Servlets είναι ανεξάρτητες πλατφόρμας, είναι ασφαλείς, φορτώνονται μία φορά σε αντίθεση με τα σενάρια CGI, τα οποία φορτώνονται σε κάθε αίτηση, μπορούν να αντλούν εύκολα στοιχεία από βάσεις δεδομένων και έχουν την ικανότητα να επικοινωνούν πολύ καλά με τα applets.



Εικόνα 25: Η θέση των Java Servlets σε μία διαδικτυακή εφαρμογή

Εκτός από τον Java Web Server, οι Java servlets υποστηρίζονται και από μία σειρά άλλων διακομιστών Ιστού, όπως είναι, για παράδειγμα ο Apache και ο IIS. Μάλιστα, οι Servlets, είναι τοποθετημένες στον υποκατάλογο servlet ενός διακομιστή Ιστού και μπορούν να κληθούν μέσω του ακόλουθου URL: `http://your.host.com:port/servlet/ServletName.class`, όπου, αντί για `your.host.com:port`, πρέπει κανείς να βάλει το όνομα του διακομιστή Ιστού και τον αριθμό της θύρας που χρησιμοποιεί. Το `ServletName.class` είναι η Java Servlet που έχει τοποθετηθεί στον υποκατάλογο servlet του εξυπηρετητή Ιστού. [58]

2.16.9.2 Ο Τρόπος Λειτουργίας των Java Servlets

Η βιβλιοθήκη API Servlet ανήκει στα API των επεκτάσεων, με την έννοια ότι δεν είναι μέρος του πυρήνα της πλατφόρμας της Java. Περιλαμβάνει τα πακέτα `javax.servlet` και `javax.servlet.http`. Αρχιτεκτονικά, κάθε Java Servlet πρέπει να υλοποιεί τη διασύνδεση Servlet. Η διασύνδεση αυτή διαθέτει πέντε μεθόδους που περιγράφονται στον παρακάτω Πίνακα:

Μέθοδος	Λειτουργία
<code>void init (ServletConfig config)</code>	Καλείται αυτόματα μία φορά και μόνο στην έναρξη του κύκλου ζωής μιας Servlet για την αρχικοποίηση της λειτουργίας της.
<code>ServletConfig getServletConfig()</code>	Επιστρέφει μία αναφορά σε ένα αντικείμενο που υλοποιεί τη διασύνδεση <code>ServletConfig</code> .
<code>void service (ServletRequest request, ServletResponse response)</code>	Είναι η πρώτη μέθοδος που καλείται σε κάθε Servlet, για να αποκριθεί σε μία κλήση από τον πελάτη.
<code>String getServletInfo()</code>	Ορίζεται από τον προγραμματιστή, ώστε να επιστρέφει ένα αλφαριθμητικό με πληροφορίες σχετικές με την εκάστοτε Servlet (π.χ. για την έκδοσή της ή το συγγραφέα της)
<code>void destroy()</code>	Εκτελεί έναν «καθαρισμό», όταν λήξει μία Servlet.

Πίνακας 30: Οι μέθοδοι της διασύνδεσης Servlet

Στα πακέτα των Java Servlets υπάρχουν δύο κλάσεις που υλοποιούν τη διασύνδεση Servlet, η κλάση `GenericServlet` (ανήκει στο `javax.servlet`) και η κλάση `HttpServlet` (ανήκει στο `javax.servlet.http`). Αυτές οι κλάσεις υλοποιούν εξ ορισμού όλες τις μεθόδους της Servlet.

Μία Java Servlet του Παγκόσμιου Ιστού επεκτείνει την κλάση `HttpServlet`. Οι πιο συνήθεις τύποι αιτήσεων (request types ή request methods) είναι η GET και η POST. Μία αίτηση GET παίρνει πληροφορίες, όπως, για παράδειγμα αρχεία `.html`, από τον εξυπηρετητή. Μία αίτηση POST στέλνει δεδομένα στον εξυπηρετητή, όπως, για παράδειγμα τα στοιχεία ενός χρήστη από μία φόρμα του προγράμματος περιήγησης. Η κλάση `HttpServlet` περιέχει τις μεθόδους `doGet()` και `doPost()`, για να αποκριθεί στις αιτήσεις GET και POST του πρωτοκόλλου HTTP. Κάθε αίτηση `doGet` ή `doPost` λαμβάνει ένα αντικείμενο που υλοποιεί τη διασύνδεση `HttpServletRequest` και ένα αντικείμενο που υλοποιεί τη διασύνδεση `HttpServletResponse`. Ο διακομιστής Ιστού, μέσα στον οποίο εκτελείται η Servlet, δημιουργεί αυτά τα αντικείμενα και τα περνά στη μέθοδο `service()`. Το αντικείμενο `HttpServletRequest` περιέχει την αίτηση που προέρχεται από τον πελάτη, ενώ το `HttpServletResponse` περιέχει την απάντηση προς τον πελάτη.

[58] [60]

2.16.10 Η Τεχνολογία JavaBeans της Java

Παλαιότερα, την εποχή που επικρατούσε το λειτουργικό σύστημα MS-DOS, για να φτιάξει κανείς ένα πρόγραμμα με μία γλώσσα προγραμματισμού έπρεπε να προγραμματίζει γραμμή-γραμμή και εντολή-εντολή όλα όσα ήθελε να φτιάξει. Με την καθιέρωση των Windows αναπτύχθηκαν γλώσσες όπως η Visual Basic, το Delphi και η Visual C, οι οποίες καθιέρωσαν το λεγόμενο οπτικό προγραμματισμό (visual programming). Ο οπτικός προγραμματισμός δίνει τη δυνατότητα στον προγραμματιστή με πολύ μικρότερη προσπάθεια (σε σχέση με τις παλαιότερες γλώσσες), να δημιουργήσει ισχυρές εφαρμογές που να διαθέτουν πολύ καλό περιβάλλον διασύνδεσης με το χρήστη. Στη Visual Basic, για παράδειγμα, η βάση κάθε εφαρμογής είναι μία φόρμα (ένα ορθογώνιο πλαίσιο) στην οποία ο προγραμματιστής τοποθετεί κουμπιά διαταγών, πλαίσια κειμένου, λίστες, μενού, και μία πληθώρα άλλων χειριστηρίων ή στοιχείων ελέγχου, controls, όπως λέγονται, προκειμένου να φτιάξει το περιβάλλον επικοινωνίας. Ο προγραμματιστής παίρνει με το ποντίκι τα στιγμιότυπα των χειριστηρίων που βρίσκονται έτοιμα στην εργαλειοθήκη και τα τοποθετεί σε όποια θέση της φόρμας θέλει, ρυθμίζοντας το μέγεθός τους, το χρώμα τους και όλες τις άλλες ιδιότητες που έχουν. Η εργασία αυτή (που θυμίζει μάλλον ζωγραφική) δίνει τη δυνατότητα στον προγραμματιστή να φτιάξει το περιβάλλον επικοινωνίας, χωρίς να γράψει κώδικα για αυτό. Ο κώδικας που γράφει σχετίζεται με αυτή καθαυτή την εφαρμογή, καθώς και με τον τρόπο που θα αποκρίνονται στο χρήστη τα συστατικά που τοποθέτησε στη φόρμα. Όλα αυτά τα χειριστήρια της εργαλειοθήκης που τοποθετήθηκαν στη φόρμα αναγνωρίζονται από τη γλώσσα χωρίς κανένα πρόβλημα. Αν και η εργαλειοθήκη περιλαμβάνει ένα σημαντικό αριθμό στοιχείων ελέγχου, ο προγραμματιστής μπορεί να βρει στο Διαδίκτυο, να αγοράσει, ή να φτιάξει ο ίδιος πολύ περισσότερα στοιχεία ελέγχου και να τα εντάξει στην εργαλειοθήκη.

Η αντίστοιχη τεχνολογία της Java η οποία επιτρέπει τη δημιουργία ανάλογων συστατικών, ονομάζεται JavaBeans. Με τη χρήση αυτής της τεχνολογίας ο προγραμματιστής μπορεί να γράψει έναν κόκκο (bean), που είναι ένα κομμάτι επαναχρησιμοποιήσιμου κώδικα (reusable code), το οποίο σε άλλες περιπτώσεις έχει εξωτερική μορφή γραφικών και σε άλλες όχι. Ένα κατάλληλο εργαλείο οπτικής ανάπτυξης που λέγεται περιβάλλον δόμησης εφαρμογών (builder), χρησιμοποιεί τους κόκκους (beans) και με αυτούς δημιουργεί applets και εφαρμογές. Σύμφωνα με τη γνώμη πολλών ειδικών, η τεχνολογία JavaBeans υπερτερεί, διότι οι ανάλογες τεχνολογίες της Microsoft λειτουργούν κυρίως στα Windows ή μόνο στα Windows. Με την τεχνολογία JavaBeans η Sun θέλησε να δημιουργήσει μία αρχιτεκτονική βασισμένη σε συστατικά (component based) για τη δημιουργία, οπτικών ή μη, τμημάτων επαναχρησιμοποιήσιμου κώδικα για την κατασκευή εφαρμογών. Επιδιώκει, επίσης, να ενθαρρύνει τρίτους κατασκευαστές να δημιουργήσουν και να διανείμουν τα δικά τους συστατικά JavaBeans. Η τεχνολογία αυτή για να γίνει αποδεκτή πρέπει να παρέχει δυνατότητες στο χρήστη οι οποίες να του επιτρέπουν την εξέταση και τη ρύθμιση των ιδιοτήτων, των μεθόδων, και των γεγονότων κάθε συστατικού.

Ο κόκκος είναι στην απλούστερη μορφή του μία κλάση ή μία ομάδα κλάσεων της Java. Αυτό που τον κάνει να ξεχωρίζει είναι πως μπορεί να διαθέσει στο περιβάλλον οπτικής δόμησης εφαρμογών (builder) τις ιδιότητές του, τις μεθόδους του, και τα γεγονότα στα οποία ανταποκρίνεται. Η διαδικασία αυτή είναι γνωστή ως ενδοσκόπηση (introspection). Το πρόγραμμα δόμησης έχει τη δυνατότητα να αναλύει και να τροποποιεί τις μεθόδους και τις ιδιότητες του κόκκου. Οι τροποποιήσεις, όμως, αυτές δεν θα είχαν καμιά αξία, αν δε μπορούσαν να αποθηκεύονται και να ανακαλούνται, όποτε χρειάζεται. Η δυνατότητα ενός συστατικού να αποθηκεύει και να ανακτά την κατάστασή του είναι γνωστή ως διατηρησιμότητα (persistence). Η διαδικασία της αποθήκευσης ενός αντικειμένου σε ένα αρχείο γίνεται με

τη μορφή ρεύματος από bytes και ονομάζεται σεριοποίηση (serialization), ενώ η αντίστροφη πορεία της ανασύστασης του αντικειμένου από τις σειρές των bytes του αποθηκευμένου αρχείου λέγεται αποσειριοποίηση (deserialization). Η συσκευασία και η διανομή των κόκκων γίνεται μέσα από αρχεία τύπου JAR, τα οποία είναι ειδικά αρχεία σε μορφή .zip που μπορούν να αποθηκεύουν αρχεία κλάσεων, σειριοποιημένα αρχεία αντικειμένων, εικόνες, και ήχους.

Με βάση τον ορισμό της Sun, ο κόκκος (bean) είναι ένα επαναχρησιμοποιήσιμο κομμάτι κώδικα το οποίο μπορεί να χρησιμοποιηθεί από ένα εργαλείο οπτικής ανάπτυξης για τη δημιουργία μίας εφαρμογής Java.

Ένας κόκκος περιέχει:

- Γεγονότα (events).
- Ιδιότητες (properties).
- Μεθόδους (methods).

Τα γεγονότα είναι αντικείμενα που χρησιμοποιούνται, για να δηλώσουν μία αλλαγή στην κατάσταση ενός αντικειμένου-πηγής του γεγονότος. Το μοντέλο αποστολής γεγονότος (delegation event model) που εξετάσαμε στο Κεφάλαιο 9 παρέχει τους μηχανισμούς για τη σύνδεση της πηγής του γεγονότος και του ακροατή, ο οποίος «αφουγκράζεται» την πηγή. Το μοντέλο περιέχει:

- Πηγές γεγονότων (beans.)
- Αντικείμενα γεγονότων (event objects).
- Ακροατές γεγονότων (event listeners)

Όταν συμβεί ένα γεγονός σε έναν κόκκο, παράγεται ένα αντικείμενο γεγονότος το οποίο ειδοποιεί μέσω της κατάλληλης μεθόδου τον ακροατή που έχει καταχωρηθεί, για να παρακολουθεί το γεγονός.

Οι ιδιότητες που καθορίζουν τα χαρακτηριστικά ενός κόκκου είναι κάτι ανάλογο με τις ιδιότητες ενός τυπικού συστατικού της Java, για παράδειγμα ενός αντικειμένου Button. Ορισμένες από τις ιδιότητες αυτές αντιπροσωπεύουν ορατά χαρακτηριστικά του κόκκου, ενώ άλλες μη ορατά.

Για να δημιουργηθεί ή να ανακτηθεί μία ιδιότητα χρησιμοποιούνται μέθοδοι όπως οι παρακάτω:

- `public void set<Property_Name>(<Type> <Value>)`
- `public <Type> get<Property_Name>()`

Μία ιδιότητα μπορεί να είναι είτε μόνο για εγγραφή (λείπει η μέθοδος get), είτε μόνο για ανάγνωση (λείπει η set), είτε για εγγραφή και ανάγνωση (υπάρχουν και οι δύο μέθοδοι).

Οι μέθοδοι του κόκκου προσφέρονται προς χρήση αφού πρώτα δηλωθούν ως public. Παρόλα αυτά, ο προγραμματιστής μπορεί να καθορίσει ποιες μέθοδοι θα φαίνονται στο εργαλείο οπτικής δόμησης το οποίο χρησιμοποιεί τον κόκκο και ποιες όχι.

[59] [60]

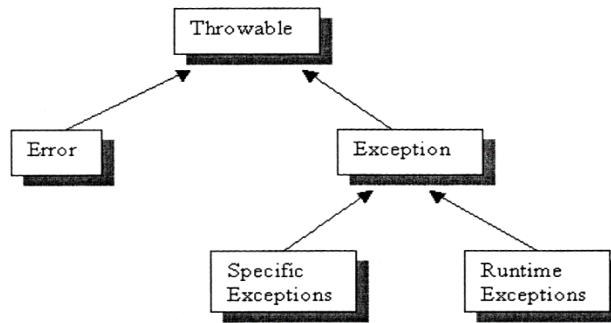
2.16.11 Οι Εξαιρέσεις στην Java

2.16.11.1 Εισαγωγή

Όταν συμβαίνει ένα πρόβλημα στην Java, για παράδειγμα, όταν προσπαθούμε να διαβάσουμε ή να γράψουμε δεδομένα έξω από τα όρια ενός πίνακα ή όταν εισάγουμε λάθος το όνομα ενός αρχείου, τότε το πρόγραμμα γεννά μία εξαίρεση (exception). Με άλλα λόγια η εξαίρεση παράγεται, όταν το σύστημα εκτέλεσης του προγράμματος δε γνωρίζει τι πρέπει να κάνει σε μία απροσδόκητη κατάσταση. Η Sun καθιέρωσε τον όρο «εξαίρεση», γιατί η γλώσσα Java είναι φτιαγμένη έτσι, ώστε τα λάθη σε ένα πρόγραμμα να είναι πολύ περιορισμένα, δηλαδή να αποτελούν την εξαίρεση και όχι τον κανόνα!

Το πλεονέκτημα από τη χρήση μίας εξαίρεσης είναι ότι υπάρχει διαχωρισμός του κώδικα ο οποίος χειρίζεται την εξαίρεση, από τον κώδικα ο οποίος εκτελείται, καθώς όλα γίνονται κανονικά. Επίσης, με τη χρήση συγκεκριμένων εξαιρέσεων, υπάρχει η υποχρέωση ο κώδικας του προγράμματος να περιέχει τμήματα τα οποία «αποκρίνονται» στην αντίστοιχη εξαίρεση, αλλιώς το πρόγραμμα δε μεταγλωττίζεται.

Η εξαίρεση είναι ένα αντικείμενο μίας υποκλάσης της κλάσης `java.lang.Throwable` και δημιουργείται, όταν εμφανίζεται στο πρόγραμμα μία μη κανονική κατάσταση. Η κλάση `Throwable` έχει δύο υποκλάσεις, την `Error` και την `Exception`, όπως φαίνεται στην παρακάτω Εικόνα:



Εικόνα 26: Η κλάση των εξαιρέσεων Throwable της Java

Οι εξαιρέσεις που ορίζονται από την κλάση Error παριστάνουν καταστάσεις σφαλμάτων από τις οποίες δεν υπάρχει δυνατότητα επανόδου. Στην υποκλάση Exception ανήκουν οι ειδικές εξαιρέσεις, καθώς και οι εξαιρέσεις που δημιουργούνται από σφάλματα κατά τη διάρκεια εκτέλεσης του προγράμματος (runtime exceptions). Έχουμε, επίσης, τη δυνατότητα να δημιουργήσουμε και τις δικές μας εξαιρέσεις, επεκτείνοντας την κλάση java.lang.Exception.

Η συγκεκριμένη κλάση παρέχει δύο βασικούς κατασκευαστές:

- I. Exception()
- II. Exception (String message)

Ο πρώτος κατασκευαστής δημιουργεί απλώς ένα αντικείμενο της κλάσης Exception, ενώ ο δεύτερος σε περίπτωση εξαίρεσης, εμφανίζει και ένα κατάλληλο μήνυμα το οποίο έχουμε δημιουργήσει εμείς.

Η πιο συνηθισμένη μέθοδος της κλάσης Exception είναι η μέθοδος:

public String getMessage(), η οποία επιστρέφει το μήνυμα της εξαίρεσης.

Στην Java, κάθε μέθοδος έχει τη δυνατότητα να δηλώσει ότι μπορεί να δημιουργήσει μία εξαίρεση, αφότου συμβούν κάποιες καταστάσεις. Αυτό γίνεται με τη χρήση της δεσμευμένης λέξης throws, η οποία χρησιμοποιείται στη γραμμή δήλωσης της μεθόδου. Ωστόσο, μία εξαίρεση μπορεί να παραχθεί και μέσα από το πρόγραμμα με τη χρήση της εντολής throw. Στο σημείο αυτό, πρέπει να τονίσουμε πως η εντολή throw δεν πρέπει να συγχέεται με τη throws.

[55] [56]

2.16.11.2 Οι Τύποι των Εξαιρέσεων

Υπάρχουν πολλές εξαιρέσεις που προκύπτουν από τον τύπο Throwable, αλλά είναι διασκορπισμένες σε διάφορα πακέτα. Στον πίνακα που ακολουθεί, περιγράφουμε μερικές από τις εξαιρέσεις που συναντάμε συχνότερα. Οι περισσότερες από αυτές τις κλάσεις των εξαιρέσεων είναι υποκλάσεις της RuntimeException και ανήκουν στα πακέτα java.lang και java.util.

Όνομα κλάσης	Κατάσταση που αντιστοιχεί
ArithmeticException	Αδύνατη αριθμητική κατάσταση, όπως, για παράδειγμα, διαίρεση με το μηδέν.
ArrayStoreException	Μη επιτρεπτή προσπάθεια για αποθήκευση ενός δεδομένου σ' έναν πίνακα
EmptyStackException	Μη επιτρεπτή πράξη σε ένα άδειο αντικείμενο στοίβας, όπως, για παράδειγμα, η άντληση (pop) ενός στοιχείου από αυτό.
FileNotFoundException	Προσπάθεια για προσπέλαση ενός μη υπάρχοντος αρχείου.
IllegalArgumentException	Πέρασμα ενός μη επιτρεπτού ορίσματος σε μία μέθοδο.
IndexOutOfBoundsException	Προσπάθεια να χρησιμοποιηθεί δείκτης έξω από τα όρια ενός αντικείμενου.
IOException	Αποτυχημένη προσπάθεια εισόδου/ εξόδου, όπως, για παράδειγμα, αδυναμία ανάγνωσης δεδομένων από ένα αρχείο.
NegativeArraySizeException	Προσπάθεια να ορισθεί ένας πίνακας με αρνητικό μέγεθος.
NumberFormatException	Αποτυχημένη μετατροπή αλφαριθμητικού δεδομένου σε αριθμητικό.

OutOfMemoryException	Έλλειψη επαρκούς χώρου μνήμης για την καταχώριση ενός νέου αντικειμένου.
StringIndexOutOfBoundsException	Προσπαθεία προσπέλασης μίας μη υπάρχουσας θέσης χαρακτήρα ενός αλφαριθμητικού.

Πίνακας 31: Οι πιο βασικές εξαιρέσεις της Java

[55] [56]

2.16.11.3 Ο Χειρισμός των Εξαιρέσεων

Η φιλοσοφία του χειρισμού των εξαιρέσεων (exception handling) βασίζεται στο ότι, όταν μία μέθοδος εντοπίσει ένα σφάλμα παράγει μία εξαίρεση. Αφού παραχθεί μία εξαίρεση, δε σημαίνει ότι θα αντιμετωπισθεί και αυτόματα. Αν υπάρχει κώδικας ο οποίος την αντιμετωπίζει, λέμε ότι η εξαίρεση θα «συλληφθεί» (catch) και θα «υποστεί διαχείριση» (handle). Ο προγραμματιστής ο οποίος θα ασχοληθεί με τις εξαιρέσεις θα χρησιμοποιήσει τρία τμήματα κώδικα, το τμήμα try, το τμήμα catch και το τμήμα finally.

Όταν θέλουμε να συλλάβουμε μία εξαίρεση, πρέπει να δημιουργήσουμε ένα τμήμα try μέσα στο οποίο πρέπει να τοποθετήσουμε τον κώδικα της μεθόδου, ο οποίος ενδέχεται να δημιουργήσει μία εξαίρεση. Το τμήμα αυτό αποτελείται από τη λέξη try και δύο άγκιστρα που περικλείουν τον «επίφοβο» κώδικα, σύμφωνα με τη σύνταξη:

```
try
{
// Code
}
```

Το τμήμα try ακολουθείται από ένα τμήμα catch. Κάθε τμήμα catch προσδιορίζει τον τύπο της εξαίρεσης που συλλαμβάνει και περιέχει κώδικα για το χειρισμό της. Το τμήμα αυτό ορίζεται με τη λέξη catch ακολουθούμενη από μία παρένθεση. Μέσα στην παρένθεση υπάρχει ο τύπος της εξαίρεσης με την οποία ασχολείται το catch. Υπάρχουν, επιπλέον, δύο άγκιστρα που περιέχουν τον κώδικα χειρισμού της εξαίρεσης, σύμφωνα με τη σύνταξη:

```
catch (ExceptionType e)
{
// Handling Code
}
```

Όταν κάποια από τις εντολές που βρίσκεται μέσα στο τμήμα try δημιουργήσει μία εξαίρεση, τότε η ροή του προγράμματος διακόπτεται και το σύστημα ψάχνει ανάμεσα στα τμήματα catch να βρει το τμήμα που χειρίζεται την εξαίρεση. Μόλις δράσει ο κώδικας που υπάρχει στο τμήμα try/catch, συνεχίζεται η εκτέλεση του υπόλοιπου προγράμματος, εκτός κι αν το πρόγραμμα τερματιστεί με κάποια δήλωση μέσα στο τμήμα try/catch.

Αν ένα τμήμα try δημιουργεί πολλούς τύπους εξαιρέσεων, μπορούμε να έχουμε πολλά τμήματα catch μετά από ένα try, έτσι, ώστε το κάθε catch να χειρίζεται και από μία εξαίρεση. Στην περίπτωση αυτή έχει σημασία η σειρά με την οποία είναι τοποθετημένα τα τμήματα catch.

Έτσι στον παρακάτω κώδικα:

```
try
{
// Code
}
catch (Exception e)
{
// Handling Code 1
```

```

}
catch (ArithmeticException e)
{
// Handling Code 2
}

```

το δεύτερο τμήμα catch, που αντιστοιχεί στη δημιουργία μίας εξαίρεσης τύπου ArithmeticException, δεν πρόκειται να εκτελεσθεί ποτέ, γιατί και να συμβεί μία τέτοια εξαίρεση, εκτελείται ο κώδικας που αντιστοιχεί στην εξαίρεση τύπου Exception η οποία είναι υπερκλάση της ArithmeticException. Επομένως, όταν έχουμε πολλά τμήματα catch που αναφέρονται σε εξαιρέσεις οι οποίες ανήκουν στην ίδια ιεραρχία, πρέπει να τοποθετούμε προσεκτικά τα μπλοκ, ώστε στην αρχή να βρίσκονται αυτά με τις χαμηλότερες στην ιεραρχία υποκλάσεις και μετά να προχωρούμε στις υψηλότερες.

Όσον αφορά, τώρα, το τμήμα finally είναι προαιρετικό και τοποθετείται στο τέλος μίας ομάδας από μπλοκ try/catch, σύμφωνα με τη σύνταξη:

```

try
{
// Try Code
}
catch (Exception e)
{
// Handling Code 1
}
catch (Exception2 e)
{
// Handling Code 2
}
finally
{
// Finally Code
}

```

Ο κώδικας που βρίσκεται μέσα στο τμήμα finally εκτελείται οπωσδήποτε μετά τον κώδικα στο try-catch και ανεξάρτητα από το αν θα παρουσιασθούν ή όχι εξαιρέσεις. Η μόνη περίπτωση να μην εκτελεσθεί ο κώδικας του finally είναι αν υπάρχει δήλωση System.exit(O), η οποία τερματίζει το σύστημα, πριν την εκτέλεση του κώδικα αυτού.

[55] [57]

2.16.11.4 Η Δημιουργία των Εξαιρέσεων ενός Χρήστη

Οι εξαιρέσεις είναι και αυτές κλάσεις, όπως όλες οι άλλες μέσα στην ιεραρχία της Java. Παρότι υπάρχουν αρκετές τυποποιημένες εξαιρέσεις οι οποίες καλύπτουν διάφορες περιπτώσεις λαθών, κάποιες φορές χρειάζεται να δημιουργήσουμε τις δικές μας εξαιρέσεις, για να αντιμετωπίσουμε ειδικούς τύπους σφαλμάτων που ενδεχομένως να εμφανισθούν στο πρόγραμμά μας.

Η εξαίρεση που θα δημιουργήσουμε πρέπει να κληρονομεί από κάποια άλλη εξαίρεση της ιεραρχίας της Java. Λογικό είναι η νέα εξαίρεση να είναι παρεμφερής με εκείνη από την οποία κληρονομεί. Αν αυτό δεν μπορεί να γίνει, έχουμε τη δυνατότητα να θεωρήσουμε ότι η εξαίρεση που φτιάχνουμε κληρονομεί από την κλάση Exception, η οποία βρίσκεται στην κορυφή της ιεραρχίας.

[56]

2.16.12 Σύγκριση της Java

Αμέσως παρακάτω παρατίθενται μερικά από τα χαρακτηριστικά της Java που την κάνουν να ξεχωρίζει:

- Είναι σχετικά απλή. Οι δημιουργοί της Java άφησαν έξω από τη γλώσσα «δύσκολα» χαρακτηριστικά που συναντάμε σε άλλες γλώσσες, όπως είναι, για παράδειγμα, οι δείκτες (pointers), οι δομές (structures) και οι ενώσεις (unions) της C.
- Είναι αντικειμενοστραφής (object oriented). Η Java, όπως και η C, χρησιμοποιεί τις κλάσεις (classes) για να οργανώσει τον κώδικα σε λογικές ενότητες. Κατά το χρόνο εκτέλεσης, το εκάστοτε πρόγραμμα δημιουργεί από τις κλάσεις αντικείμενα. Τα αντικείμενα αυτά έχουν δύο συνιστώσες: τα πεδία και τις μεθόδους. Τα πεδία περιγράφουν τι είναι το αντικείμενο, ενώ οι μέθοδοι περιγράφουν τι κάνει το αντικείμενο. Οι κλάσεις μπορούν να κληρονομήσουν ιδιότητες από άλλες κλάσεις. Εκείνο που δεν επιτρέπεται είναι η πολλαπλή κληρονομικότητα (multiple inheritance), όπου μία κλάση έχει τη δυνατότητα να κληρονομήσει πεδία και μεθόδους από περισσότερες από μία άλλες κλάσεις.
- Είναι μεταγλωττιζόμενη (compiled), αλλά μπορεί να χαρακτηριστεί και ως διερμηνευόμενη (interpreted). Η μεταγλώττιση του εκάστοτε πηγαίου προγράμματος (source code) έχει ως αποτέλεσμα την παραγωγή ενός ειδικού κώδικα, ο οποίος ονομάζεται κώδικας byte (bytecode). Ο κώδικας, αυτός, ενώ μοιάζει με τον κώδικα σε γλώσσα μηχανής, μπορεί να εκτελεστεί από οποιοδήποτε λειτουργικό σύστημα (πχ Windows, Unix, Mac OS) που διαθέτει διερμηνευτή της Java. Εδώ εναπόκειται η λύση σχετικά με την τοποθέτηση προγραμμάτων σε Σελίδες του Παγκόσμιου Ιστού. Τα προγράμματα των υπολογιστών είναι κατά κύριο λόγο προσανατολισμένα στο υλικό (hardware) του υπολογιστή και στο λειτουργικό του σύστημα (software). Έτσι, για παράδειγμα, ένα πρόγραμμα γραμμένο για τα Windows δε μπορεί να τρέξει στον Macintosh, ούτε μία εφαρμογή για Macintosh μπορεί να εκτελεστεί σε ένα σταθμό εργασίας Unix. Η Java λύνει αυτό το πρόβλημα δημιουργώντας εφαρμογές με ανεξαρτησία πλατφόρμας (platform independence). Αυτό γίνεται διότι ο μεταγλωττιστής της Java δε δημιουργεί γηγενή κώδικα (native code) προσαρμοσμένο στο συγκεκριμένο τύπο υπολογιστή (όπως κάνουν οι άλλες γλώσσες), αλλά δημιουργεί κώδικα byte, ο οποίος, όπως προειπώθηκε, μοιάζει πολύ με κώδικα γλώσσας μηχανής, με τη διαφορά πως σε όλες τις πλατφόρμες είναι ο ίδιος. Συνεπώς, ένα πρόγραμμα τέτοιου τύπου που δημιουργήθηκε, για παράδειγμα, από έναν υπολογιστή με Windows, μπορεί να τρέξει αυτούσιο σε έναν άλλο υπολογιστή με λειτουργικό σύστημα Unix.
- Είναι ασφαλής: Η Java έχει σχεδιαστεί από την αρχή με τέτοιο τρόπο, ώστε να παρέχει ασφάλεια εκτέλεσης του κώδικα σε δίκτυο. Αυτό είχε ως αποτέλεσμα τον περιορισμό αρκετών από τα χαρακτηριστικά που έχουν η C και η C++. Έτσι δεν υπάρχουν δείκτες ούτε μπορεί να γίνει αυθαίρετη προσπέλαση διευθύνσεων της μνήμης. Τα μέτρα αυτά παρέχουν προστασία από τη δράση των ιών. Η Java χρησιμοποιεί, επίσης, έναν ισχυρό μηχανισμό για τον έλεγχο αναμενόμενων και μη αναμενόμενων σφαλμάτων (exception handling).
- Υποστηρίζει πολυνημάτωση (multithreading). Ένα πρόγραμμα της Java μπορεί να περιλαμβάνει πολλές ξεχωριστές διαδικασίες, οι οποίες εκτελούνται συνεχώς και ανεξάρτητα η μία από την άλλη. Μπορεί, για παράδειγμα, από το πρόγραμμα αυτό να μεταδίδεται μία εικόνα και συγχρόνως ο εκάστοτε χρήστης να εισάγει στοιχεία από το πληκτρολόγιο. Αυτό το ισχυρό χαρακτηριστικό είναι τόσο συνυφασμένο με τη γλώσσα, ώστε καθιστά αδύνατη την σύνδεση της Java με περιβάλλοντα τα οποία δεν υποστηρίζουν πολυνημάτωση, όπως τα Windows 3.1.
- Κάνει συλλογή ακρήστων (garbage collection): Τα προγράμματα της Java κάνουν από μόνα τους συλλογή ακρήστων. Αυτό σημαίνει πως ο εκάστοτε προγραμματιστής δε χρειάζεται να φροντίζει πλέον ο ίδιος για τη διαγραφή ακρήστων δεδομένων από τη μνήμη.
- Μπορεί να δημιουργήσει γρήγορο κώδικα. Το σύστημα του συνδυασμού μεταγλωττιστή (compiler), για τη δημιουργία κώδικα byte ο οποίος μπορεί να τρέχει σε όλες τις πλατφόρμες και διερμηνευτή (interpreter), για τη βήμα-βήμα εκτέλεση του κώδικα byte σε κάθε συγκεκριμένη πλατφόρμα, θεωρείται σχετικά αργό. Για το λόγο αυτό, διάφορες εταιρείες έχουν αναπτύξει ειδικούς μεταγλωττιστές, οι οποίοι μετατρέπουν τον κώδικα byte σε κώδικα γλώσσας μηχανής, ο οποίος ανταγωνίζεται σε ταχύτητα τον κώδικα της C++ ή άλλων γλωσσών. Οι μεταγλωττιστές αυτοί ονομάζονται “just in time compilers” (JIT).
- Παρέχει βιβλιοθήκες κώδικα για διάφορες χρήσεις όπως είναι, για παράδειγμα, η δημιουργία γραφικών, ο χειρισμός αλφαριθμητικών, οι μαθηματικές πράξεις, η προσπέλαση αρχείων στο Διαδίκτυο, ο χειρισμός σχεσιακών βάσεων δεδομένων, η δημιουργία εφαρμογών πελάτη-εξυπηρετητή, η κλήση απομακρυσμένων αντικειμένων.

- Με την Java διευκολύνεται η κατασκευή δυναμικών Ιστοσελίδων (dynamic Web Pages). Μεταξύ άλλων, με τη βοήθεια της Java σε μία Ιστοσελίδα δίνεται η δυνατότητα:
 - ο Να αναπαράγεται ήχος ή βίντεο.
 - ο Να χρησιμοποιούνται διανυσματικά γραφικά, αντί των ψηφιογραφικών (bitmap) εικόνων.
 - ο Να υλοποιείται κίνηση σε πραγματικό χρόνο
 - ο Να δημιουργούνται φόρμες εισαγωγής στοιχείων από τους χρήστες.
 - ο Να φτιάχνονται παιχνίδια και εφαρμογές πραγματικού χρόνου, στα οποία να συμμετέχουν ταυτόχρονα πολλοί παίκτες.

[55] [56] [57]

2.17 Η Java και οι Βάσεις Δεδομένων

2.17.1 Εισαγωγή

Πριν από μερικά χρόνια, για να αντιμετωπίσει η εταιρεία Microsoft το πρόβλημα της επικοινωνίας μεταξύ των διαφορετικών τύπων βάσεων δεδομένων, ανέπτυξε το πρότυπο ODBC, ένα πρότυπο για την προσπέλαση βάσεων δεδομένων. Το ODBC (Open Data Base Connectivity) (απόδοση στα Ελληνικά: Ανοιχτή Συνδεσιμότητα Βάσης Δεδομένων) καθορίζει ένα πρότυπο συμπεριφοράς διαφόρων εφαρμογών στην περίπτωση που αυτές οι εφαρμογές καλούν βάσεις δεδομένων και ένα άλλο πρότυπο συμπεριφοράς των προαναφερόμενων εφαρμογών, όταν μία συγκεκριμένη βάση δεδομένων ανταποκρίνεται σε κλήσεις τύπου ODBC από μία από τις προηγούμενες εφαρμογές.

Η εταιρεία Sun συμπεριέλαβε στη γλώσσα Java τη διασύνδεση JDBC, η οποία καθορίζει τον τρόπο επικοινωνίας μίας εφαρμογής με σχεσιακές βάσεις δεδομένων. Η Java με αυτόν τον τρόπο κατορθώνει να υποστηρίζει τις σχεσιακές βάσεις δεδομένων, που δημιουργούνται από διαφορετικά DBMS, όπως κατορθώνει να υποστηρίζει και τα διαφορετικά λειτουργικά συστήματα

Η διασύνδεση JDBC της γλώσσας Java επιτρέπει στον εκάστοτε προγραμματιστή να γράφει εφαρμογές βάσεων δεδομένων με τη χρήση αποκλειστικά και μόνο της γλώσσας Java. Παρότι η ίδια η Sun δηλώνει ότι η ακολουθία χαρακτήρων JDBC δεν είναι ακρωνύμιο, πολλοί θεωρούν ότι είναι τα αρχικά των λέξεων Java Data Base Connectivity (απόδοση στα Ελληνικά: Συνδεσιμότητα Βάσεων Δεδομένων Java). Όπως και το πρότυπο ODBC της Microsoft, έτσι και το πρότυπο JDBC της Sun, καθορίζει το πώς οι εφαρμογές θα επικοινωνούν με τις βάσεις δεδομένων και ταυτόχρονα το πώς θα κατασκευάζονται οι βάσεις δεδομένων, ώστε να μπορούν να ανταποκρίνονται στις κλήσεις των εφαρμογών. Η διαφορά είναι ότι το ODBC είναι γραμμένο σε γλώσσα C, ενώ το JDBC αποτελείται από κλάσεις και διασυνδέσεις αποκλειστικά γραμμένες σε γλώσσα Java.

Η διασύνδεση JDBC της γλώσσας Java δίνει στον εκάστοτε προγραμματιστή τις εξής δυνατότητες:

- Να συνδέει μία εφαρμογή με μία υπάρχουσα βάση δεδομένων.
- Να συντάσσει SQL εντολές και να τις εκτελεί στην εκάστοτε βάση δεδομένων, προκειμένου να γίνουν διάφορες εργασίες.
- Να λαμβάνει από την εκάστοτε βάση δεδομένων τα αποτελέσματα των εκτελέσεων κατάλληλων σε κάθε περίπτωση SQL εντολών στη συγκεκριμένη βάση και να επεξεργάζεται περαιτέρω τα εν λόγω αποτελέσματα.

[55] [61]

2.17.2 Οι Βάσεις Δεδομένων και οι Σχεσιακές Βάσεις Δεδομένων

Με τον όρο Βάση Δεδομένων (ΒΔ) (Data Base\Database-DB) εννοούμε μία οργανωμένη συλλογή από δεδομένα, τα οποία αναφέρονται σε ένα συγκεκριμένο αντικείμενο (πχ επιχείρηση, οργανισμό, άτομο, γεγονός). Ο τρόπος με τον οποίο είναι οργανωμένη η εν λόγω συλλογή, επιτρέπει να καταχωρούνται εύκολα νέα στοιχεία σε αυτή καθώς και να αντλούνται γρήγορα πληροφορίες από αυτή. Για παράδειγμα, ας θεωρηθεί μία επιχείρηση της οποίας τα έγγραφα τοποθετούνται σε φακέλους με τυχαίο τρόπο και οι φάκελοι τοποθετούνται και αυτοί χωρίς τάξη σε διαφορετικά δωμάτια ή κτίρια. Αν θελήσει κάποιος υπάλληλος να βρει ένα έγγραφο τι θα συμβεί; Απόλυτο χάος... Από την άλλη πλευρά, αν η επιχείρηση οργανωθεί σε τμήματα, κάθε τμήμα έχει μόνο τα έγγραφα που το αφορούν, κάθε έγγραφο

καταχωρηθεί στο σωστό φάκελο και κάθε φάκελος βρίσκεται σε καθορισμένη θέση, είναι πολύ απλό να ανακτηθεί μία πληροφορία ή να αρχειοθετηθεί ένα νέο έγγραφο.

Μία βάση δεδομένων, λοιπόν, επιχειρεί να υλοποιήσει ηλεκτρονικά την αρχειοθέτηση της δεύτερης περίπτωσης του πιο πάνω παραδείγματος. Κάθε πληροφορία που εισάγεται σε μία βάση δεδομένων αποθηκεύεται σε καθορισμένη θέση μέσα σε αυτή. Αν μία βάση δεδομένων είναι καλά οργανωμένη, τότε δίνει τη δυνατότητα να αντληθούν από αυτή σύνθετες πληροφορίες σε πολύ μικρό χρονικό διάστημα. Επίσης, μία ηλεκτρονική βάση δεδομένων είναι δυνατό να προσπελαστεί από μακριά καθώς και να χρησιμοποιείται ταυτόχρονα από πολλούς χρήστες.

Μία βάση δεδομένων δημιουργείται από ένα ειδικό πακέτο λογισμικού που ονομάζεται Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ) (Data Base Management System-DBMS). Το εν λόγω σύστημα, εκτός από το πως δημιουργεί μία βάση δεδομένων, έχει τη δυνατότητα να αποθηκεύει νέα στοιχεία σε αυτή, να εκτελεί σύνθετες εργασίες σε αυτή (πχ ταξινομήσεις ή αναζητήσεις) καθώς, επίσης, να συντηρεί τα δεδομένα της και να τα προστατεύει από λανθασμένους χειρισμούς των χρηστών. Υπάρχουν πάρα πολλά DBMS πακέτα από μεγάλες εταιρείες όπως είναι, για παράδειγμα η IBM, η Microsoft, η Sybase, η Oracle καθώς και από πολλές μικρότερες. Ένα γνωστό σε όλους DBMS είναι η Access, που βρίσκεται μέσα στο πακέτο Office της Microsoft.

Υπάρχουν διάφορα μοντέλα βάσεων δεδομένων, όμως, το πλέον διαδεδομένο είναι το σχεσιακό μοντέλο που είναι αρκετά ευέλικτο και εξαιρετικά δημοφιλές. Μία Σχεσιακή Βάση Δεδομένων (ΣΒΔ) (Relational Database-RDB) περιέχει τα δεδομένα οργανωμένα σε πίνακες. Κάθε πίνακας αποτελείται από γραμμές (rows) ή εγγραφές (records) και στήλες (columns) ή πεδία (fields). Κάθε γραμμή αντιστοιχεί σε μία ξεχωριστή οντότητα (πχ πρόσωπο ή αντικείμενο), ενώ κάθε στήλη προσδιορίζει ένα χαρακτηριστικό της οντότητας αυτής.

Από τα πεδία κάθε εγγραφής ενός πίνακα μίας βάσης δεδομένων ένα (ή συνδυασμός περισσότερων) πρέπει να ταυτοποιεί μοναδικά τη συγκεκριμένη εγγραφή, ώστε αυτή να ξεχωρίζει απόλυτα από τις άλλες εγγραφές του παραπάνω πίνακα. Το πεδίο αυτό ονομάζεται πρωτεύον κλειδί (primary key). Μία πραγματική εφαρμογή αποτελείται συνήθως από πολλούς πίνακες οι οποίοι συνδέονται μεταξύ τους. Η σύνδεση γίνεται με ένα κοινό πεδίο (στήλη). Το πεδίο αυτό στον έναν πίνακα αποτελεί το πρωτεύον κλειδί, ενώ στον άλλον το ξένο κλειδί (foreign key). Στο σχεσιακό περιβάλλον όλοι οι πίνακες πρέπει να είναι κανονικοποιημένοι (normalized) τουλάχιστον σε πρώτο βαθμό κανονικοποίησης, δηλαδή να μην περιέχουν επαναλαμβανόμενες ομάδες δεδομένων.

[55] [63]

2.17.3 Η Γλώσσα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων, SQL

Η γλώσσα SQL (Structured Query Language) (απόδοση στα Ελληνικά: Δομημένη Γλώσσα Ερωτήματος) είναι η γλώσσα που χρησιμοποιείται κυρίως για το χειρισμό μίας σχεσιακής βάσης δεδομένων. Αναπτύχθηκε από την εταιρεία IBM τη δεκαετία μεταξύ 1970 και 1980 και τυποποιήθηκε περί τα τέλη του 1980. Υπάρχουν διάφορες εκδόσεις της SQL. Κάθε εταιρεία-κατασκευαστής βάσεων δεδομένων έχει προσθέσει στην SQL κάποια ειδικότερα χαρακτηριστικά. Η SQL διαφέρει από τις διαδικαστικές γλώσσες στο ότι επικεντρώνεται στον ορισμό και το χειρισμό των δεδομένων και οι εντολές της είναι πολύ κατανοητές μίας και γράφονται σχεδόν σε απλά Αγγλικά. Οι πιο βασικές εντολές της SQL είναι οι επόμενες:

❖ CREATE DATABASE

- Λειτουργία: Δημιουργεί μία νέα βάση δεδομένων.
- Σύνταξη:
`CREATE DATABASE dbname;`
- Παράμετρος (-οι):
 - ο `dbname`: Υποχρεωτική. Το όνομα της εκάστοτε βάσης δεδομένων που πρόκειται να δημιουργηθεί.

❖ CREATE TABLE

- Λειτουργία: Δημιουργεί ένα νέο πίνακα σε μία βάση δεδομένων.
- Σύνταξη:
`CREATE TABLE table_name`
(
`column_name_1 data_type(size),`

```

column_name_2 data_type(size),
column_name_3 data_type(size),
.....
column_name_N data_type(size)
);

```

- Παράμετρος (-οι):
 - ο table_name: Υποχρεωτική. Το όνομα του εκάστοτε πίνακα μίας βάσης δεδομένων που πρόκειται να δημιουργηθεί.
 - ο column_name: Υποχρεωτική. Το όνομα κάθε στήλης του πίνακα table_name.
 - ο data_type: Υποχρεωτική. Ο τύπος των δεδομένων που μπορεί να αποθηκευτεί σε κάθε στήλη του πίνακα table_name. Οι πιο βασικοί τύποι δεδομένων που υποστηρίζει η γλώσσα SQL για τα διάφορα DBMS φαίνονται στον παρακάτω Πίνακα:

Τύπος δεδομένων	Περιγραφή
CHAR	Ομάδα χαρακτήρων σταθερού μήκους με εύρος από 0 ως 255 χαρακτήρες.
DATE	Ημερομηνία στη μορφή (EEEE-MM-HH) (Ε-τος-Μήνας-Ημέρα) με εύρος από '1.000-01-01' ως '9.999-12-31'.
DECIMAL	Μεγάλος αριθμός σταθερής υποδιαστολής.
DOUBLE	Μεγάλος αριθμός κινητής υποδιαστολής.
FLOAT	Μικρός αριθμός κινητής υποδιαστολής.
INT	Ακέραιος αριθμός με εύρος από -2.147.483.648 ως 2.147.483.647 ή από 0 ως 4.294.967.295 για μη προσημασμένους ακεραίους αριθμούς.
TIME	Ώρα στη μορφή ΩΩ:ΛΛ:ΔΔ (Ώρα:Λεπτά:Δευτερόλεπτα) με εύρος από '-838:59:59' ως '838:59:59'.
VARCHAR	Ομάδα χαρακτήρων μεταβλητού μήκους με εύρος από 0 ως 255 χαρακτήρες.

Πίνακας 32: Οι πιο βασικοί τύποι δεδομένων της SQL

- ο size: Προαιρετική. Το μέγιστο μήκος που μπορεί να έχει κάθε στήλη του table_name.
- ❖ DROP TABLE
 - Λειτουργία: Διαγράφει έναν ολόκληρο πίνακα από μία βάση δεδομένων.
 - Σύνταξη:
DROP TABLE table_name;
 - Παράμετρος (-οι):
 - ο table_name: Υποχρεωτική. Το όνομα του εκάστοτε πίνακα που πρόκειται να διαγραφεί από μία βάση δεδομένων.
- ❖ SELECT
 - Λειτουργία: Επιλέγει δεδομένα από μία βάση δεδομένων. Αυτά τα επιλεγμένα κάθε φορά δεδομένα αποθηκεύονται σε έναν πίνακα που κρατάει τα αποτελέσματα επιστροφής της τρέχουσας μεθόδου, ο οποίος καλείται “result-set”.
 - Σύνταξη:
*SELECT column_name_1,column_name_2,column_name_3,...,column_N
FROM table_name;*
 - Παράμετρος (-οι):
 - ο table_name: Υποχρεωτική. Το όνομα του πίνακα μίας βάσης δεδομένων από όπου πρόκειται να επιλεγθούν δεδομένα.

- ο `column_name`: Το όνομα της κάθε στήλης του πίνακα `table_name` που θα επιλεγεί, για να επιστραφούν οι εγγραφές της. Να τονιστεί πως, αν είναι επιθυμητό να επιλεγούν όλες οι στήλες του πίνακα `table_name`, τότε στη σύνταξη της τρέχουσας μεθόδου η έκφραση `column_name_1,column_name_2, column_name_3,...,column_N` είναι προτιμότερο να αντικατασταθεί με έναν αστερίσκο (*).

❖ UPDATE

- Λειτουργία: Ανανεώνει τις τιμές εγγραφών σε ένα πίνακα μίας βάσης δεδομένων.

- Σύνταξη:

UPDATE table_name

SET column_name_1=value_1,column_name_2=value_2,column_name_3=value_3,

...,column_name_N=value_N

(WHERE condition);

- Παράμετρος (-οι):

❖ ALTER TABLE:

- Λειτουργία: Προσθέτει (1^η σύνταξη) ή διαγράφει (2^η σύνταξη) μία ή περισσότερες στήλες σε ή από έναν πίνακα μίας βάσης δεδομένων.

- 1^η Σύνταξη:

ALTER TABLE table_name

ADD column_name data_type

2^η Σύνταξη:

ALTER TABLE table_name

DROP COLUMN column_name

- Παράμετρος (-οι):

- ο `table_name`: Υποχρεωτική. Το όνομα του εκάστοτε πίνακα μίας βάσης δεδομένων που πρόκειται να υποστεί επεξεργασία.

- ο `column_name`: Υποχρεωτική. Το όνομα κάθε στήλης του πίνακα `table_name` που θα προστεθεί ή θα διαγραφεί σε ή από αυτόν.

- ο `data_type`: Υποχρεωτική. Ο τύπος των δεδομένων ο οποίος μπορεί να αποθηκευτεί σε κάθε στήλη που προστίθεται στον πίνακα `table_name`.

❖ INSERT INTO:

- Λειτουργία: Εισάγει εγγραφές σε έναν πίνακα μίας βάσης δεδομένων είτε δίχως να καθορίζει τα ονόματα των στηλών όπου εισέρχονται τα νέα δεδομένα, αλλά μόνο τις τιμές αυτών (1^η σύνταξη), είτε καθορίζοντας τόσο τα ονόματα των στηλών που εισέρχονται τα νέα δεδομένα όσο και τις τιμές αυτών (2^η σύνταξη).

- 1^η Σύνταξη:

INSERT INTO table_name

VALUES (value_1,value_2,value_3,...,value_N);

- 2^η Σύνταξη:

INSERT INTO table_name (column_name_1,column_name_2,column_name_3,...,column_name_N)

VALUES (value_1,value_2,value_3,...,value_N);

- Παράμετρος (-οι):

- ο `table_name`: Υποχρεωτική. Το όνομα του εκάστοτε πίνακα μίας βάσης δεδομένων στον οποίο πρόκειται να εισαχθούν νέες εγγραφές.

- ο `column_name`: Υποχρεωτική. Το όνομα κάθε στήλης του πίνακα `table_name` όπου θα εισαχθούν νέες εγγραφές.

- ο `value`: Υποχρεωτική. Η τιμή που θα καταχωρηθεί στην εκάστοτε νέα εγγραφή που θα εισαχθεί στον πίνακα `table_name`.

❖ DELETE:

- Λειτουργία: Διαγράφει εγγραφές από έναν πίνακα.

- Σύνταξη:

*DELETE FROM table_name
(WHERE condition);*

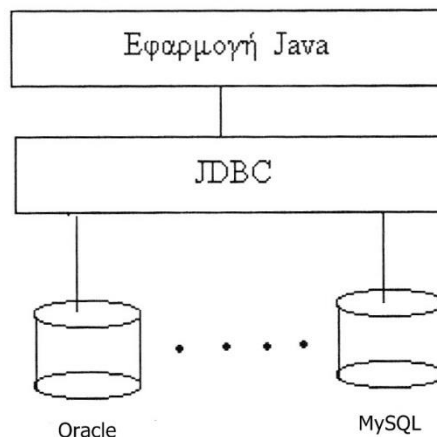
- Παράμετρος (-οι):
 - ο *table_name*: Υποχρεωτική. Το όνομα του εκάστοτε πίνακα από όπου πρόκειται να διαγραφούν εγγραφές.
 - ο *condition*: Προαιρετική. Η συνθήκη που πρέπει να πληρούν οι εγγραφές του πίνακα *table_name*, ώστε να διαγραφούν. Υπάρχει η πιθανότητα η συγκεκριμένη παράμετρος και γενικότερα όλη η έκφραση *WHERE condition* να παραλείπεται. Σε μία τέτοια περίπτωση θα διαγραφούν όλες οι εγγραφές του πίνακα *table_name*.
 - ο *table_name*: Υποχρεωτική. Το όνομα του εκάστοτε πίνακα μίας βάσης δεδομένων του οποίου οι εγγραφές πρόκειται να ανανεωθούν.
 - ο *column*: Το όνομα της κάθε στήλης του πίνακα *table_name* της οποίας οι εγγραφές θα ανανεωθούν.
 - ο *value*: Η νέα τιμή που θα καταχωρηθεί σε κάθε εγγραφή του πίνακα *table_name* που θα ανανεωθεί.
 - ο *condition*: Προαιρετική. Η συνθήκη που πρέπει να πληρούν οι εγγραφές του πίνακα *table_name*, ώστε να ανανεωθούν. Υπάρχει η πιθανότητα η συγκεκριμένη παράμετρος και γενικότερα όλη η έκφραση *WHERE condition* να παραλείπεται. Σε μία τέτοια περίπτωση θα ανανεωθούν όλες οι εγγραφές της στήλης ή των στηλών του πίνακα *table_name* που επιλέχθηκαν για το σκοπό αυτό.

[55] [63] 64]

2.17.4 Η Διασύνδεση JDBC της Γλώσσας Java

2.17.4.1 Εισαγωγή

Ο βασικός στόχος της διασύνδεσης JDBC είναι να επιτρέπει στα προγράμματα της γλώσσας Java να εκτελούν SQL εντολές και να λαμβάνουν τα αποτελέσματα των εντολών αυτών από την κατάλληλη σε κάθε περίπτωση βάση δεδομένων. Όλες οι βάσεις δεδομένων που είναι συμβατές με το πρότυπο ODBC της Microsoft είναι συμβατές και με το πρότυπο JDBC της Sun. Κατά αυτόν τον τρόπο μία Java εφαρμογή αποδεσμεύεται από τη βάση δεδομένων ενός συγκεκριμένου κατασκευαστή. Από την πλευρά του εκάστοτε χρήστη, η προαναφερόμενη εφαρμογή φαίνεται όπως στην παρακάτω Εικόνα:



Εικόνα 27: Ο ρόλος της JDBC σε μία εφαρμογή της Java

Μεταξύ άλλων, η διασύνδεση JDBC της γλώσσας Java περιλαμβάνει τα ακόλουθα πιο βασικά συστατικά:

- Την κλάση *DriverManager*. Συνδέει τον κατάλληλο JDBC οδηγό, που αντιστοιχεί στη βάση δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί η εκάστοτε Java εφαρμογή. Μόλις

αποκατασταθεί η εν λόγω σύνδεση, η DriverManager διεκπεραιώνει την επικοινωνία μεταξύ της προαναφερόμενης εφαρμογής και της προαναφερόμενης βάσης δεδομένων.

- Τη γέφυρα JDBC-ODBC. Επιτρέπει σε μία εφαρμογή συμβατή με το πρότυπο JDBC να επικοινωνεί με μία βάση δεδομένων συμβατή με το πρότυπο ODBC, μεταφέροντας εντολές και αποτελέσματα μέσω του κατάλληλου σε κάθε περίπτωση ODBC οδηγού.

[55] [61]

2.17.4.2 Οι Κλάσεις και οι Διασυνδέσεις της JDBC

Τόσο οι κυριότερες κλάσεις όσο και σημαντικότερες διασυνδέσεις της βιβλιοθήκης JDBC φαίνονται στον μετέπειτα Πίνακα:

Κλάσεις και διασυνδέσεις	Περιγραφή
Connection	Συνδέει μία Java εφαρμογή με μία βάση δεδομένων.
DriverManager	Φορτώνει τους κατάλληλους JDBC οδηγούς μίας βάσης δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί η εκάστοτε Java εφαρμογή.
ResultSet	Επιτυγχάνεται διαχείριση των δεδομένων μίας βάσης δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί η εκάστοτε Java εφαρμογή.
ResultSetMetaData	Περιλαμβάνει σταθερές και μεθόδους που δίνουν πληροφορίες σχετικά με τα αντικείμενα της διασύνδεσης ResultSet της βιβλιοθήκης JDBC.
Statement	Πραγματοποιεί κλήσεις για την εκτέλεση SQL εντολών σε μία βάση δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί η εκάστοτε Java εφαρμογή.
Types	Περιλαμβάνει ως σταθερές όλους τους διαφορετικούς τύπους δεδομένων της τυπικής γλώσσας SQL.

Πίνακας 33: Οι πιο βασικές κλάσεις και διασυνδέσεις της JDBC

Στο σημείο τούτο, αξίζει να μελετηθεί λίγο περισσότερο η διασύνδεση Statement της βιβλιοθήκης JDBC. Η συγκεκριμένη διασύνδεση παρέχει μεθόδους που χρησιμοποιούνται για την επικοινωνία μίας Java εφαρμογής με μία βάση δεδομένων μέσω SQL εντολών. Τέτοιες μέθοδοι είναι οι execute(), executeQuery() και η executeUpdate(). Όσον αφορά τη μέθοδο η executeQuery() εκτελεί μία SQL εντολή σε μία βάση δεδομένων, όπως είναι, για παράδειγμα, η SELECT, και επιστρέφει ένα αντικείμενο της διασύνδεσης ResultSet της βιβλιοθήκης JDBC. Έπειτα, η μέθοδος executeUpdate() εκτελεί μία SQL εντολή σε μία βάση δεδομένων όπως είναι, για παράδειγμα, η INSERT INTO, ή η UPDATE, ή η DELETE, οι οποίες γενικά ενημερώνουν την προαναφερόμενη βάση δεδομένων. Τέλος, η μέθοδος execute() εκτελεί μία SQL εντολή σε μία βάση δεδομένων σε μορφή αλφαριθμητικού και επιστρέφει μία λογική τιμή, ανάλογα αν παρήχθη ή όχι ένα αντικείμενο της ResultSet

Μόλις, τώρα, εκτελείται ένα ερώτημα προς μία βάση δεδομένων, τα αποτελέσματα επιστρέφονται ως ένα αντικείμενο της διασύνδεσης ResultSet της βιβλιοθήκης JDBC. Το αντικείμενο αυτό παρουσιάζει τις τιμές υπό μορφή πίνακα, με γραμμές και στήλες. Η προσπέλαση στα στοιχεία του συγκεκριμένου πίνακα γίνεται ανά μία γραμμή κάθε φορά. Το παραπάνω αντικείμενο περιέχει ένα δείκτη προς την αναφερόμενη κάθε φορά γραμμή. Στην αρχή ο δείκτης δείχνει μία θέση πριν την πρώτη γραμμή του παραπάνω πίνακα. Για να μετακινηθεί ο δείκτης προς την επόμενη γραμμή του εν λόγω πίνακα χρησιμοποιείται η μέθοδος next() της ResultSet. Αν υπάρχει επόμενη γραμμή, τότε η next() επιστρέφει αληθινή τιμή, ενώ αν ο τρέχων πίνακας έχει φθάσει στο τέλος του επιστρέφει ψευδής τιμή. Η ResultSet παρέχει μία σειρά από μεθόδους get() για την επιστροφή των διαφόρων τύπων των δεδομένων κάθε γραμμής του πιο πάνω πίνακα. Μερικές από τις μεθόδους αυτές είναι οι ακόλουθες: getBoolean(), getByte(), getDouble(), getFloat(), getInt(), getObject(), getShort(), getString(), getObject(), getTime(), getTimestamp(). Επιπρόσθετα η getColumnCount() δίνει τον αριθμό των στηλών του προηγούμενου πίνακα, ενώ η洗columnName() επιστρέφει τα ονόματα των στηλών του.

Κλείνοντας, να παρατηρηθεί πως η διασύνδεση `ResultSetMetaData` της βιβλιοθήκης `JDBC` περιέχει σταθερές και μεθόδους που δίνουν πληροφορίες σχετικά με τα αντικείμενα της διασύνδεσης `ResultSet` της βιβλιοθήκης `JDBC` και πως ένα αντικείμενο της `ResultSetMetaData` μπορεί να δημιουργηθεί με τη μέθοδο `getMetaData()` της `ResultSet`.

[55] [61]

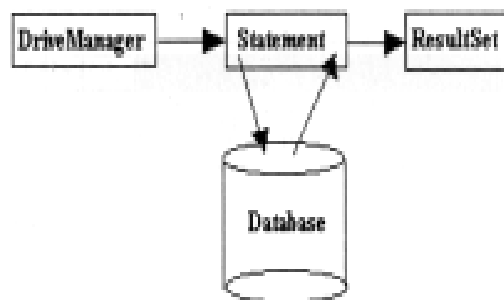
2.17.4.3 Η Σύνδεση της Java με τις Σχεσιακές Βάσεις Δεδομένων

Η όλη διαδικασία, σύνδεσης ενός προγράμματος γραμμένο σε γλώσσα Java με μία σχεσιακή βάση δεδομένων, ξεκινά με την κλάση `DriverManager` της βιβλιοθήκης `JDBC`. Η κλάση αυτή αναλαμβάνει να φορτώσει τους απαραίτητους `JDBC` οδηγούς και να δημιουργήσει τις κατάλληλες διασυνδέσεις με τις αντίστοιχες βάσεις δεδομένων. Η `DriverManager` χρησιμοποιείται, επιπρόσθετα, για να δώσει πληροφορίες σχετικά με τους οδηγούς που είναι φορτωμένοι και χρησιμοποιούνται ήδη από κάποιες βάσεις. Επιπλέον, παρέχει στατικές μεθόδους για τη δημιουργία αντικειμένων που εφαρμόζουν τη διασύνδεση `Connection` της βιβλιοθήκης `JDBC`. Οι μέθοδοι αυτοί χρησιμοποιούνται για τη δημιουργία αντικειμένων κατά ζήτηση. Όταν, δηλαδή, είναι επιθυμητό να δημιουργηθεί μία καινούργια σύνδεση σε έναν `JDBC` οδηγό, ζητείται από την `DriverManager` να δημιουργήσει ένα αντικείμενο για το σκοπό αυτό.

Η διασύνδεση `Connection` της βιβλιοθήκης `JDBC` αντιπροσωπεύει μία ανταλλαγή πληροφοριών με μία συγκεκριμένη βάση δεδομένων. Πρώτα κατασκευάζεται ένα αντικείμενο αυτής της διασύνδεσης για τη σύνδεση με την προαναφερόμενη βάση δεδομένων και στη συνέχεια υλοποιούνται αντικείμενα της διασύνδεσης `Statement` της βιβλιοθήκης `JDBC`, που ορίζουν και εκτελούν συγκεκριμένες `SQL` ερωτήσεις στην τρέχουσα βάση. Ένα αντικείμενο αυτής της διασύνδεσης δημιουργείται με τη μέθοδο `createStatement()` της `Connection`. Αφού δημιουργηθεί το συγκεκριμένο αντικείμενο, εκτελείται με τη μέθοδο `executeQuery()` της `Statement`, η οποία του περνά ένα αλφαριθμητικό με το κείμενο της `SQL` ερώτησης.

Τα αποτελέσματα της παραπάνω διαδικασίας επιστρέφονται με τη μορφή ενός αντικειμένου που υλοποιεί τη διασύνδεση `ResultSet` της βιβλιοθήκης `JDBC`. Το αντικείμενο αυτό εμφανίζει τα αποτελέσματα υπό μορφή πίνακα, με γραμμές και στήλες.

Η συνολική διαδικασία σύνδεσης μίας εφαρμογής γραμμένης σε Java κώδικα με μία σχέση βάση δεδομένων, που περιγράφηκε παραπάνω, απεικονίζεται στην ακόλουθη Εικόνα:



Εικόνα 28: Η διαδικασία της σύνδεσης μίας εφαρμογής της Java με μία σχεσιακή βάση δεδομένων

Πιο αναλυτικά, μία εφαρμογή που κάνει χρήση της βιβλιοθήκης `JDBC`, προκειμένου να ανοίξει και να χρησιμοποιήσει μία σχεσιακή βάση, δεδομένων ακολουθεί τα παρακάτω βήματα:

- I. Εισάγει τις απαραίτητες κλάσεις και διασυνδέσεις οι οποίες βρίσκονται στο πακέτο `java.sql`, της γλώσσας Java σύμφωνα με τη σύνταξη:

```
import java.sql.*;
```

- II. Φορτώνει τον εκάστοτε κατάλληλο `JDBC` οδηγό με τη μέθοδο `forName()` της κλάσης `Class` της γλώσσας Java σύμφωνα με τη σύνταξη:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

- III. Προσδιορίζει μονοσήμαντα την κατάλληλη κάθε φορά πηγή-βάση δεδομένων, δημιουργώντας ένα αλφαριθμητικό που περιέχει την κατάλληλο URL σύμφωνα με τη σύνταξη:

```
String sourceURL=new String("jdbc:odbc:databaseName");
```

Όπου:

- sourceURL: Το όνομα του αλφαριθμητικού που περιέχει τη URL της εκάστοτε πηγής-βάσης δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί μία Java εφαρμογή.
- databaseName: Το όνομα της βάσης δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί μία Java εφαρμογή.

- IV. Δημιουργεί ένα αντικείμενο της διασύνδεσης Connection της βιβλιοθήκης JDBC, το οποίο πραγματοποιεί τη σύνδεση με την κατάλληλη σε κάθε φάση πηγή-βάση δεδομένων και κατασκευάζει ένα περιβάλλον μέσα στο οποίο μπορούν να δημιουργούνται και να εκτελούνται SQL εντολές. Το συγκεκριμένο αντικείμενο δημιουργείται με τη μέθοδο getConnection() της κλάσης DriverManager της βιβλιοθήκης JDBC, σύμφωνα με τη σύνταξη:

```
Connection databaseConnection=DriverManager.getConnection(sourceURL);
```

Όπου databaseConnection το όνομα του αντικείμενου της διασύνδεσης Connection της βιβλιοθήκης JDBC που πραγματοποιεί τη σύνδεση με την κατάλληλη σε κάθε φάση πηγή-βάση δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί μία Java εφαρμογή.

- V. Δημιουργεί ένα αντικείμενο της διασύνδεσης Statement της βιβλιοθήκης JDBC με τη μέθοδο createStatement() της διασύνδεσης Connection της βιβλιοθήκης JDBC σύμφωνα με τη σύνταξη:

```
Statement statement=databaseConnection.createStatement();
```

Όπου statement το όνομα του αντικείμενου της διασύνδεσης Statement της βιβλιοθήκης JDBC που δημιουργείται κατά τη διαδικασία σύνδεσης μίας βάσης δεδομένων με μία Java εφαρμογή.

- VI. Εκτελεί μία SQL ερώτηση στην κατάλληλη σε κάθε φάση πηγή-βάση δεδομένων με τη μέθοδο executeQuery() του αντικειμένου statement της διασύνδεσης Statement της βιβλιοθήκης JDBC, που δημιουργήθηκε στο προηγούμενο βήμα, περνώντας στη συγκεκριμένη μέθοδο ως όρισμα ένα αλφαριθμητικό με το κείμενο της προαναφερόμενης ερώτησης σύμφωνα με τη σύνταξη:

```
ResultSet result=statement.executeQuery(sql);
```

Όπου:

- sql: Το αλφαριθμητικό που περνάει ως όρισμα στη μέθοδο executeQuery() της διασύνδεσης Statement της βιβλιοθήκης JDBC κατά τη φάση εκτέλεσης ενός SQL ερωτήματος σε μία βάση δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί μία Java εφαρμογή.
- result: Το όνομα ενός αντικειμένου της διασύνδεσης ResultSet της βιβλιοθήκης JDBC το οποίο αποθηκεύει σε ένα πίνακα τα αποτελέσματα της απάντησης ενός SQL ερωτήματος που έγινε σε μία βάση δεδομένων με την οποία είναι επιθυμητό να επικοινωνεί μία Java εφαρμογή.

- VII. Παίρνει δεδομένα από το αντικείμενο result της διασύνδεσης ResultSet της βιβλιοθήκης JDBC, που επιστράφηκε στο προηγούμενο βήμα, με την κλήση μίας συνάρτησης η οποία δέχεται ως

όρισμα το συγκεκριμένο αντικείμενο και χρησιμοποιεί διαφορές τεχνικές προγραμματισμού σε συνδυασμό με μεθόδους των διασυνδέσεων Result και ResultMetaData της βιβλιοθήκης JDBC. Η κλήση της προαναφερόμενης συνάρτησης πραγματοποιείται σύμφωνα με τη σύνταξη:

```
displayResults(result);
```

- VIII. Κλείνει το αντικείμενο result της διασύνδεσης ResultSet της βιβλιοθήκης JDBC, που δημιουργήθηκε παραπάνω, σύμφωνα με τη σύνταξη:

```
result.close();
```

- IX. Κλείνει το αντικείμενο statement της διασύνδεσης Statement της βιβλιοθήκης JDBC, που δημιουργήθηκε παραπάνω, σύμφωνα με τη σύνταξη:

```
statement.close();
```

- X. Κλείνει το αντικείμενο databaseConnection της διασύνδεσης Connection της βιβλιοθήκης JDBC, που δημιουργήθηκε παραπάνω, σύμφωνα με τη σύνταξη:

```
databaseConnection.close();
```

[55] [61] [62] [64]

2.17.5 Η Γλώσσα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων, MySQL

Στη σημερινή εποχή, όπου όλο και περισσότερα δεδομένα αποθηκεύονται σε υπολογιστές, η ανάγκη για ταχύτατες και αξιόπιστες βάσεις δεδομένων έχει αυξηθεί δραματικά. Εδώ και χρόνια, μεγάλες εταιρείες όπως η Oracle, η Microsoft, και η IBM παρέχουν υψηλής τεχνολογίας εφαρμογές αποθήκευσης δεδομένων για εργασίες ζωτικής σημασίας. Αυτά τα προγράμματα χρησιμοποιήθηκαν κυρίως από τις 500 πιο επιτυχημένες επιχειρήσεις του περιοδικού Fortune, οι οποίες μπορούν να αντεπεξέλθουν στο υπερβολικό κόστος των εφαρμογών και τις υπερβολικές απαιτήσεις σε προσωπικό. Στο μεταξύ, από την κοινότητα ανοιχτού λογισμικού (open source software community) εμφανίστηκε στην αγορά ένα νέο κύμα μικρών, αξιόπιστων, και φθηνών εφαρμογών βάσεων δεδομένων. Αυτές οι εφαρμογές, από τις οποίες η MySQL και η PostgreSQL αποτελούν τα καλύτερα παραδείγματα, έδωσαν στους απλούς χρήστες και προγραμματιστές μία ανέξοδη, πρακτική επιλογή βάσης δεδομένων.

Η MySQL, ευτυχώς, έχει ξεφύγει από την αρχική μετριότητά της και έχει μετατραπεί σε μία στιβαρή, αξιόπιστη και εύκολη στη διαχείρισή της εφαρμογή βάσεων δεδομένων. Ακόμα πιο εντυπωσιακό είναι το γεγονός ότι η MySQL έχει καταφέρει να διατηρήσει τις ανοιχτού κώδικα ρίζες της, εξακολουθώντας να είναι διαθέσιμη χωρίς κόστος για μερικές χρήσεις. Οι δυνατότητες και το χαμηλό κόστος της MySQL είναι ο λόγος για τον οποίο κολοσσιαίες επιχειρήσεις, όπως είναι, για παράδειγμα η Yahoo ή η NASA τη χρησιμοποιούν μέσα στις εταιρείες τους.

Η εφαρμογή βάσεων δεδομένων MySQL δημιουργήθηκε και υποστηρίζεται από τη MySQL AB, μία επιχείρηση που εδρεύει στη Σουηδία. Πρόκειται για τη δημοφιλέστερη, και μερικοί υποστηρίζουν την καλύτερη, βάση δεδομένων ανοιχτού κώδικα που σημαίνει ότι επιτρέπεται η εκτέλεσή της ή ακόμα και η τροποποίησή της (υπάρχει η δυνατότητα λήψης του ίδιου του πηγαίου κώδικα της συγκεκριμένης βάσης). Η MySQL είναι στην ουσία μία γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων. Τεχνικά, η MySQL είναι μία εφαρμογή που διαχειρίζεται αρχεία τα οποία ονομάζονται βάσεις δεδομένων, αλλά συνήθως θα χρησιμοποιείται ο όρος βάση δεδομένων τόσο για τα αρχεία των δεδομένων όσο και για το ίδιο το πρόγραμμα. Όπως και να έχει, πρόκειται για ένα βιώσιμο, ανταγωνιστικό προϊόν, και αυτό γίνεται όλο και πιο εμφανές, κόντρα στα πανάκριβο προϊόν Oracle της ομώνυμης εταιρείας και του προϊόντος SQL Server της Microsoft.

Στο σημείο αυτό να τονιστεί ότι η MySQL υπακούει στο συντακτικό της τυπικής γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων, SQL. Για το λόγο αυτό χρησιμοποιεί σε ένα γενικό πλαίσιο, τόσο τους τύπους δεδομένων, όσο και τις εντολές της SQL (που αναπτύχθηκαν λεπτομερώς παραπάνω). Επίσης, η συνδεσιμότητά της με ένα πρόγραμμα βασισμένο στη γλώσσα Java επιτυγχάνεται σε γενικές γραμμές με τους κανόνες του προτύπου JDBC (που παρουσιάστηκαν εκτενώς προηγουμένως).

[65]

2.18 Η Τεχνική Προγραμματισμού Διαδικτύου, JSP

2.18.1 Εισαγωγή στις JSP

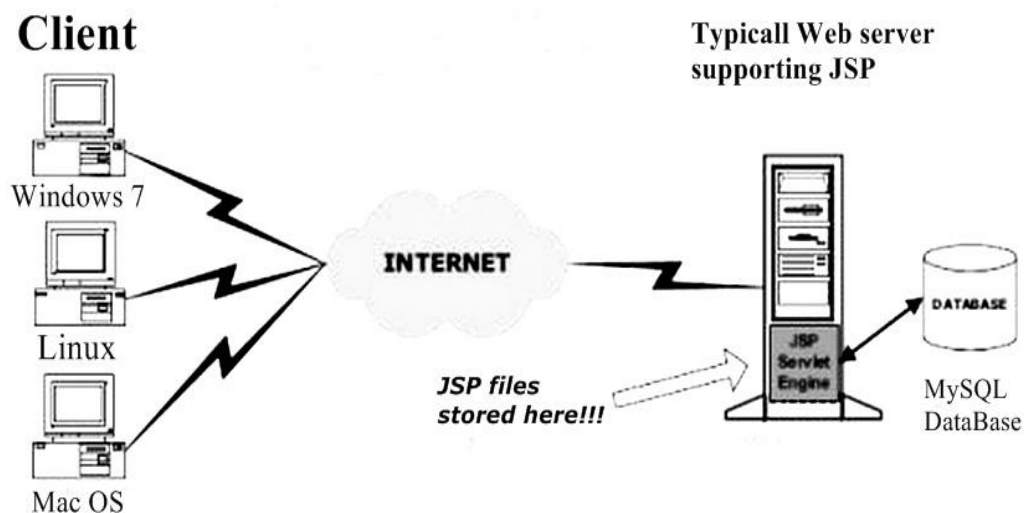
Οι JSP (JavaServer Pages) είναι μία τεχνική προγραμματισμού από την πλευρά του εξυπηρετητή (server-side programming technique) η οποία επιτρέπει τη δημιουργία δυναμικών Ιστοσελίδων. Ουσιαστικά πρόκειται για μία τεχνολογία που εισάγει Java κώδικα σε HTML έγγραφα κάνοντας χρήση ειδικών JSP ετικετών, το σύνολο των οποίων ξεκινούν με τους χαρακτήρες `<%` και κλείνουν με τους `%>`. Αυτές οι JSP ετικέτες, που διατίθενται για διάφορους σκοπούς, όπως είναι, για παράδειγμα, η ανάκτηση πληροφοριών από βάσεις δεδομένων ή άλλες πηγές, η καταγραφή των προτιμήσεων των χρηστών και ο διαμοιρασμός των πληροφοριών μεταξύ των Ιστοσελίδων, συνδυάζονται με JSP μεθόδους, ανεξάρτητες από την εκάστοτε πλατφόρμα λογισμικού (platform independent methods), με σκοπό την κατασκευή Ιστοχώρων στον Παγκόσμιο Ιστό δυναμικού περιεχομένου.

Ένα JSP συστατικό είναι ένας είδος Java Servlet, το οποίο σχεδιάζεται κατάλληλα να έχει πρόσβαση σε όλες τις διασυνδέσεις της γλώσσας Java και φυσικά στη διασύνδεση JDBC, προκειμένου να διαχειρίζεται και να επικοινωνεί με βάσεις δεδομένων καθώς και σε JavaBeans στοιχεία, ώστε να εκπληρώνει το ρόλο μίας διεπαφής ενός χρήστη για μία Java εφαρμογή Διαδικτύου. Οι προγραμματιστές του Διαδικτύου γράφουν τις JSP Ιστοσελίδες ως αρχεία κειμένου, τα οποία συνδυάζουν HTML ή XHTML κώδικα, στοιχεία XML και ενσωματώνουν JSP ενέργειες και εντολές.

[66]

2.18.2 Η Αρχιτεκτονική των JSP

Ένας διακομιστής Ιστού απαιτεί μία JSP μηχανή, δηλαδή έναν «περιέχων» (container), για να φέρει εις πέρας τις JSP Ιστοσελίδες. Στην πράξη ο JSP container συνεργάζεται με έναν εξυπηρετητή Ιστού, προκειμένου να παρέχουν ένα περιβάλλον εκτέλεσης και άλλες απαραίτητες υπηρεσίες που χρειάζονται οι JSP Σελίδες, ώστε να υλοποιηθούν. Η θέση του JSP container και των JSP αρχείων σε μία διαδικτυακή εφαρμογή αποδίδεται στην παρακάτω Εικόνα:

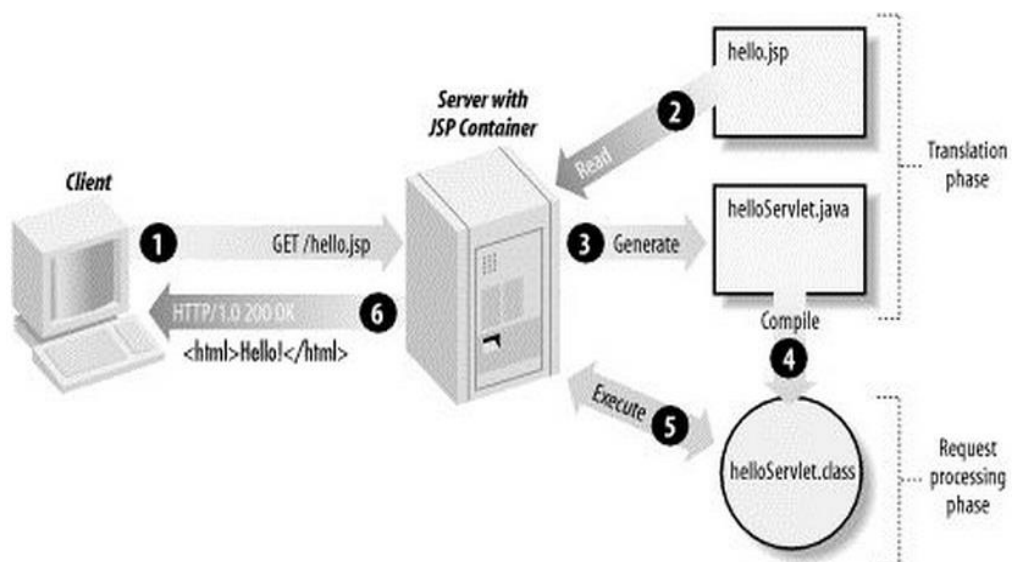


Εικόνα 29: Η θέση του JSP container και των JSP αρχείων σε μία εφαρμογή στο Διαδίκτυο

Ο τρόπος με τον οποίο ένας διακομιστής Ιστού δημιουργεί μία Ιστοσελίδα, από την οποία ο εκάστοτε πελάτης αιτείται περιεχόμενο, με τη χρήση της τεχνικής JSP περιγράφεται στα επόμενα επτά βήματα:

1. Όπως και με μία κανονική Ιστοσελίδα, ο εκάστοτε χρήστης στέλνει μέσω του προγράμματος περιήγησης που χρησιμοποιεί μία HTTP αίτηση σε έναν εξυπηρετητή Ιστού.

- II. Ο διακομιστής αυτός αναγνωρίζει πως η παραπάνω HTTP αίτηση είναι για μία JSP Ιστοσελίδα, διότι η κατάληξη της URL της εν λόγω JSP Σελίδας είναι .jsp, αντί για .html, που είναι σε μία κανονική Σελίδα, και την προωθεί στην JSP μηχανή.
 - III. Η JSP μηχανή που υπάρχει στον παραπάνω διακομιστή φορτώνει την τρέχουσα JSP Ιστοσελίδα από το δίσκο και τη μετατρέπει σε περιεχόμενο τύπου Java Servlet. Η μετατροπή αυτή είναι πολύ απλή, καθώς όλο το πρότυπο κείμενο μετατρέπεται σε εντολές println() και όλα τα JSP στοιχεία της παρούσας JSP Σελίδας μετατρέπονται σε Java κώδικα, ο οποίος υλοποιεί την κατάλληλη δυναμική συμπεριφορά της συγκεκριμένης JSP Σελίδας.
 - IV. Η JSP μηχανή του πιο πάνω διακομιστή μεταγλωττίζει την πρωτότερη Java Servlet που αντιστοιχεί στην προαναφερόμενη JSP Ιστοσελίδα σε μία εκτελέσιμη κλάση και προωθεί το αρχικό αίτημα του παραπάνω χρήστη στη Java Servlet μηχανή που υπάρχει στον εν λόγω διακομιστή.
 - V. Το κομμάτι του τρέχοντος διακομιστή που καλείται Java Servlet μηχανή φορτώνει την προηγούμενη Java Servlet κλάση και την εκτελεί. Κατά τη διάρκεια της εκτέλεσης, η προαναφερόμενη Java Servlet μηχανή παράγει ένα αρχείο εξόδου σε HTML μορφή, το οποίο στη συνέχεια το περνάει στον παρόντα διακομιστή μέσω ενός HTTP μηνύματος απόκρισης.
 - VI. Αυτός ο εξυπηρετητής προωθεί την προαναφερόμενη HTTP απάντηση στο φυλλομετρητή του παραπάνω χρήστη.
- Όλα τα προαναφερόμενα βήματα, σύμφωνα με τα οποία ένας εξυπηρετητής Ιστού υλοποιεί μία JSP Ιστοσελίδα, από όπου ο εκάστοτε πελάτης αιτείται περιεχόμενο, φαίνονται στην ακόλουθη Εικόνα:



Εικόνα 30: Ο τρόπος δημιουργίας Ιστοσελίδων με τη χρήση της JSP

[67] [68]

2.18.3 Ο Κύκλος Ζωής των JSP

Ο καλύτερος τρόπος, για να αντιληφθεί κάποιος τον τρόπο λειτουργίας των JSP Ιστοσελίδων είναι η κατανόηση του κύκλου ζωής που ακολουθούν. Ο συγκεκριμένος κύκλος ζωής αποτυπώνει την όλη διαδικασία που λαμβάνει χώρα από τη δημιουργία ως και την καταστροφή των JSP Σελίδων και φέρει πολλές ομοιότητες με τον κύκλο ζωής των Java Servlets με ένα επιπλέον στάδιο που αντικατοπτρίζει τη μεταγλώττιση μίας JSP Σελίδας σε Java Servlet.

Έτσι, ο κύκλος ζωής μίας JSP Ιστοσελίδας περιλαμβάνει τις ακόλουθες φάσεις:

- I. **Μεταγλώττιση (compilation)**: Όταν ένας περιηγητής αιτείται μία JSP Ιστοσελίδα από έναν εξυπηρετητή Ιστού, η JSP μηχανή ελέγχει αρχικά, αν χρειάζεται να μεταγλωττίσει αυτή τη JSP Σελίδα. Αν η συγκεκριμένη JSP Σελίδα δεν έχει μεταγλωττιστεί ποτέ στο παρελθόν ή αν έχει υποστεί μετασχηματισμούς από την τελευταία φορά που μεταγλωττίστηκε, τότε η JSP μηχανή προχωρά στη μεταγλώττιση της εν λόγω JSP Σελίδας. Η διαδικασία, τώρα, αυτής της μεταγλώττισης γίνεται σε τρία στάδια, τα οποία είναι τα παρακάτω:

- A. Ανάλυση της τρέχουσας JSP Ιστοσελίδας (parsing the JSP).
 - B. Μετατροπή της παρούσας JSP Σελίδας σε Java Servlet (turning the JSP into a Java Servlet).
 - C. Μεταγλώττιση της προαναφερόμενης Java Servlet (compiling the Java Servlet).
- II. Initialization (αρχικοποίηση): Προκείμενου ο JSP container να προβεί στη φόρτωση οποιασδήποτε JSP Ιστοσελίδας επικαλείται τη μέθοδο `jspInit ()` σύμφωνα με την ακόλουθη σύνταξη:

```
public void jspInit()
{
// Initialization code
}
```

Ουσιαστικά η αρχικοποίηση που περιγράφεται με πιο πάνω απόσπασμα JSP κώδικα πραγματοποιείται μόνο μία φορά, με σκοπό την αρχικοποίηση συνδέσεων με βάσεις δεδομένων, ανοίγματος αρχείων, δημιουργίας πινάκων και άλλων παρεμφερών με τις προηγούμενες ενεργειών που οφείλει να πραγματοποιήσει ο JSP container, ώστε να φορτώσει την εκάστοτε JSP Σελίδα.

- III. Execution (εκτέλεση): Η συγκεκριμένη φάση του κύκλου ζωής μίας JSP Ιστοσελίδας περιέχει όλες τις αλληλεπιδράσεις με τις HTTP αιτήσεις των χρηστών μέχρι και την καταστροφή της JSP Σελίδας. Μόλις ένας χρήστης ζητήσει μία JSP Σελίδα από ένα διακομιστή Ιστού και εφόσον γίνουν οι κατάλληλες αρχικοποιήσεις από τον JSP container και η πιο πάνω JSP Σελίδα φορτωθεί, τότε η JSP μηχανή καλεί τη μέθοδο `_jspService()` σύμφωνα με τη σύνταξη:

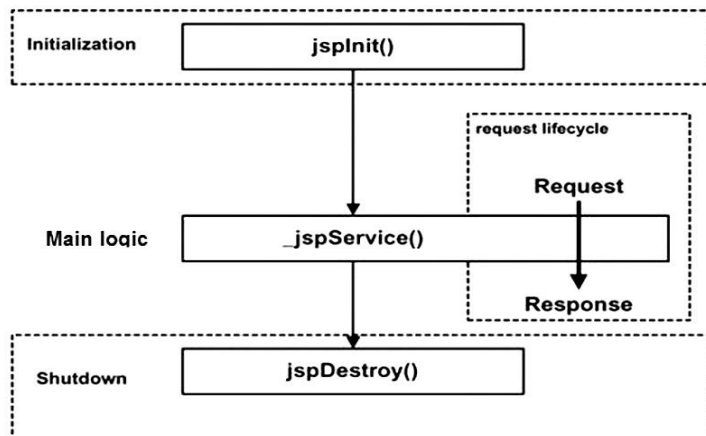
```
void _jspService(HttpServletRequest request, HttpServletResponse response)
{
// Service handling code
}
```

Η τρέχουσα μέθοδος καλείται μία φορά για κάθε HTTP αίτημα ενός χρήστη και είναι υπεύθυνη, για να παράγει την κατάλληλη απάντηση για αυτό το αίτημα. Μάλιστα η συγκεκριμένη μέθοδος δημιουργεί τα μηνύματα HTTP απόκρισης για κάθε μία από τις επτά HTTP μεθόδους GET, HEAD, POST, PUT, DELETE, CONNECT, TRACE.

- IV. Cleanup (καθαρισμός): Σε τούτο το στάδιο μίας JSP Ιστοσελίδας ο JSP container παύει να κάνει χρήση αυτής της JSP Σελίδας. Ο καθαρισμός αυτός συμβαίνει με τη βοήθεια της μεθόδου `jspDestroy()`, η οποία συντάσσεται όπως παρακάτω:

```
public void jspDestroy()
{
// Cleanup code
}
```

Οι τέσσερις προαναφερθείσες φάσεις του κύκλου ζωής μίας JSP Ιστοσελίδας αποτυπώνονται στην παρακάτω Εικόνα:



Εικόνα 31: Οι φάσεις του κύκλου ζωής μίας JSP Ιστοσελίδας

[66]

2.18.4 Η Σύνταξη των JSP

Στη συνέχεια θα παρουσιαστούν τα κύρια γνωρίσματα της σύνταξης που ακολουθείται κατά τη διαδικασία γραψίματος κώδικα για την υλοποίηση των JSP Ιστοσελίδων.

Πρώτον, τα σενάρια εργασίας της τεχνικής JSP (JSP scriptlets): Δύναται να περιέχουν οποιονδήποτε αριθμό μεταβλητών, δηλώσεων, εκφράσεων και ορισμού μεθόδων που συναντώνται στη γλώσσα Java.

Η σύνταξη ενός JSP σεναρίου εργασίας είναι η παρακάτω:

`<% code fragment %>`

Η αντίστοιχη XML σύνταξη της παραπάνω σύνταξης είναι η ακόλουθη:

```

<jsp:scriptlet>
  code fragment
</jsp:scriptlet>
  
```

Στο σημείο αυτό, να τονιστεί ότι οποιοδήποτε κείμενο ή HTML ή XML ετικέτες ή ακόμα και JSP στοιχεία πρέπει να γράφονται εκτός του σεναρίου αυτού.

Δεύτερον, οι δηλώσεις της τεχνικής JSP (JSP declarations): Ορίζουν μία ή περισσότερες μεταβλητές ή μεθόδους που μπορεί αργότερα ο εκάστοτε προγραμματιστής να εκμεταλλευτεί στον κώδικα των JSP Ιστοσελίδων. Σε κάθε περίπτωση μία μεταβλητή ή μέθοδος πρέπει πρώτα να δηλώνεται και μετά να χρησιμοποιείται.

Η σύνταξη μίας JSP δήλωσης είναι η εξής:

`<%! declaration; [declaration;]+ ... %>`

Η αντίστοιχη XML σύνταξη της πιο πάνω σύνταξης είναι η επόμενη:

```

<jsp:declaration>
  code fragment
</jsp:declaration>
  
```

Τρίτον, οι εκφράσεις της τεχνικής JSP (JSP expressions): Πρόκειται για εκφράσεις αντίστοιχες με τις εκφράσεις που απαντώνται σε γλώσσες σεναρίων, οι οποίες καταχωρούνται, εφόσον μετατραπούν σε αλφαριθμητικό, σε όποιο σημείο του JSP κώδικα μίας Ιστοσελίδας εμφανίζονται. Από τη στιγμή που η τιμή μίας JSP έκφρασης μετατρέπεται σε αλφαριθμητικό, δύναται να διατίθεται εντός

μίας γραμμής μίας περιοχής ενός κειμένου ενός JSP κώδικα μίας Σελίδας, ανεξάρτητα αν έχει ενσωματωθεί ή όχι σε μία HTML ετικέτα αυτού του κώδικα. Κάπου εδώ, να δοθεί ιδιαίζουσα σημασία στο γεγονός ότι μία JSP έκφραση μπορεί να είναι οποιαδήποτε έγκυρη έκφραση σύμφωνα με τις προδιαγραφές της γλώσσας Java, αλλά δεν τερματίζεται με τη χρήση του συμβόλου του ελληνικού ερωτηματικού (;), όπως συμβαίνει στην Java.

Η σύνταξη μίας JSP έκφρασης είναι η κατώτερη:

```
<%= expression %>
```

Η αντίστοιχη XML σύνταξη της ανώτερης σύνταξης είναι η παρακάτω:

```
<jsp:expression>  
  expression  
</jsp:expression>
```

Τέταρτον, τα σχόλια της τεχνικής JSP (JSP comments): Επισημαίνουν κείμενο ή προτάσεις εντός του JSP κώδικα μίας Ιστοσελίδας που ο JSP container πρέπει να αγνοήσει. Η εισαγωγή σχολίων στον κώδικα μίας JSP Σελίδας είναι χρήσιμη, όταν είναι επιθυμητός σχολιασμός ή απαιτείται η απόκρυψη κάποιου κομματιού του συγκεκριμένου κώδικα.

Η σύνταξη των JSP σχολίων έχει ως εξής:

```
<%-- comments --%>
```

Πέμπτον, οι κατευθυντήριες οδηγίες της τεχνικής JSP (JSP directives) (που θα μελετηθούν αναλυτικότερα πιο κάτω): Κατέχουν κρίσιμο ρόλο στον κώδικα μίας JSP Ιστοσελίδας, διότι επηρεάζουν τη συνολική δομή της Java Servlet κλάσης που παράγεται κατά την υλοποίηση της προαναφερόμενης JSP Σελίδας.

Έκτον, οι ενέργειες-δράσεις της τεχνικής JSP (JSP actions) (που θα παρουσιαστούν λεπτομερέστερα παρακάτω): Εκμεταλλεύονται δομές προγραμματισμού βασισμένες σε XML σύνταξη, με απώτερο σκοπό και στόχο να ελέγξουν τη συμπεριφορά μίας Java Servlet, η οποία γεννιέται κατά τη φάση ανάπτυξης μίας JSP Ιστοσελίδας. Τέτοιου είδους ενέργειες δίνουν την ικανότητα στους προγραμματιστές JSP Σελίδων να εισάγουν δυναμικά ένα αρχείο σε μία JSP Σελίδα, να προωθούν τον εκάστοτε χρήστη σε μία άλλη JSP Σελίδα διαφορετική από την εκάστοτε τρέχουσα JSP Σελίδα, να επαναχρησιμοποιούν JavaBeans στοιχεία σε μία JSP Σελίδα, να παράγουν HTML κώδικα για τη δημιουργία διαφόρων πρόσθετων για μία JSP Σελίδα και γενικά να εκτελούν μία σειρά παρεμφερών με τις προαναφερόμενες δραστηριότητες.

Έβδομον, τα προκαθορισμένα αντικείμενα της τεχνικής JSP (JSP implicit objects) (που θα αναπτυχθούν εκτενέστερα στη συνέχεια.): Είναι εννέα τύποι εξ ορισμού αντικειμένων που χρησιμοποιεί η JSP. Στην ουσία γίνεται λόγος για εννέα αυτόματα ορισμένες μεταβλητές από την JSP.

Όγδοον, οι τύποι δεδομένων της τεχνικής JSP (JSP literals): Πρόκειται για τους ίδιους τύπους δεδομένων που εκμεταλλεύεται και η γλώσσα Java (και που αναφέρθηκαν πιο πάνω).

Ένατον, οι τελεστές της τεχνικής JSP (JSP operators): Είναι όμοιοι με τους τελεστές που διατίθενται και από τη γλώσσα Java (και οι οποίοι σχολιάστηκαν παραπάνω).

Δέκατον, οι προγραμματιστικές δομές ελέγχου της τεχνικής JSP (JSP programming control structures). Αφορούν τις προγραμματιστικές δομές ελέγχου που προβάλλει και η γλώσσα Java και για τις οποίες έγινε λόγος πρωτότερα.

[66]

2.18.5 Οι Κατευθυντήριες Οδηγίες των JSP

Οι κατευθυντήριες οδηγίες της τεχνικής JSP (JSP directives) παρέχουν κατευθύνσεις και οδηγίες στον JSP container, λέγοντάς του πως να χειριστεί ορισμένες πτυχές της διαδικασίας επεξεργασίας μίας JSP Ιστοσελίδας. Ουσιαστικά μία JSP κατευθυντήρια οδηγία επηρεάζει ολόκληρη τη δομή μίας

Java Servlet κλάσης που δημιουργείται κατά τη φάση υλοποίησης μίας JSP Σελίδας. Επίσης, είναι δυνατό να έχουν έναν αριθμό ιδιοτήτων με τη μορφή ζεύγους «όνομα-τιμή» διαχωριζόμενες μεταξύ τους με κόμμα (.). Έτσι, βασικά συντάσσονται όπως παρακάτω:

```
<%@ directive attribute= "value" %>
```

Υπάρχουν τρεις τύποι JSP κατευθυντήριων οδηγιών, που παρουσιάζονται στον πιο κάτω Πίνακα:

Κατευθυντήρια οδηγία	Περιγραφή
<%@ page ... %>	Καθορίζει ανεξάρτητες ιδιότητες μίας JSP Ιστοσελίδας, όπως είναι, για παράδειγμα, η γλώσσα σεναρίων που χρησιμοποιεί ο κώδικας της παραπάνω JSP Σελίδας, τα κάθε είδους λάθη που προκύπτουν κατά την υλοποίησή της, οι απαιτήσεις της σε ενδιάμεσο χώρο αποθήκευσης και ενδιάμεση μνήμη και άλλες συναφείς.
<% @ include ... %>	Συμπεριλαμβάνει ένα αρχείο κατά τη φάση της μετάφρασης μίας JSP Ιστοσελίδας.
<% @ taglib ... %>	Δηλώνει μία JSP βιβλιοθήκη ετικετών, η οποία εμπεριέχει ειδικές JSP ενέργειες, που χρησιμοποιούνται εντός του κώδικα μίας JSP Ιστοσελίδα.

Πίνακας 34: Οι πιο βασικές κατευθυντήριες οδηγίες της JSP

Στη συνέχεια εξετάζονται λεπτομερέστερα οι JSP κατευθυντήριες οδηγίες του ανωτέρου Πίνακα.

Πρώτον, η κατευθυντήρια οδηγία page της τεχνικής JSP χρησιμοποιείται, με σκοπό να παρέχει οδηγίες στον JSP container που αφορούν την εκάστοτε τρέχουσα JSP Ιστοσελίδα. Η συγκεκριμένη κατευθυντήρια οδηγία δύναται να τοποθετηθεί οπουδήποτε μέσα στον κώδικα μίας JSP Σελίδας, ωστόσο, κατά σύμβαση, τοποθετείται στην κορυφή μίας JSP Σελίδας και έχει την ακόλουθη κύρια σύνταξη:

```
<%@ page attribute="value" %>
```

Η ισοδύναμη, τώρα, XML σύνταξη της παραπάνω κατευθυντήριας οδηγίας είναι η κάτωθι:

```
<jsp:directive.page attribute="value" />
```

Επιπλέον οι ιδιότητες που συνοδεύουν την παρούσα κατευθυντήρια οδηγία φαίνονται στον παρακάτω Πίνακα:

Ιδιότητα	Περιγραφή
autoFlush	Ελέγχει τη συμπεριφορά του ενδιάμεσου χώρου αποθήκευσης και της ενδιάμεσης μνήμης της παραγόμενης Java Servlet κατά τη υλοποίηση μίας JSP Ιστοσελίδας.
buffer	Καθορίζει το μοντέλο του ενδιάμεσου χώρου αποθήκευσης και της ενδιάμεσης μνήμης για την ροή εξόδου μίας JSP Ιστοσελίδας.
contentType	Καθορίζει την κωδικοποίηση των χαρακτήρων μίας JSP Ιστοσελίδας.
errorPage	Καθορίζει τη URL μίας JSP Ιστοσελίδας, η οποία εμφανίζει κάποιο σφάλμα κατά το χρόνο εκτέλεσής της.
extends	Καθορίζει την υπερκλάση που οφείλει να επεκτείνει η παραγόμενη Java Servlet κατά την υλοποίηση μίας JSP Ιστοσελίδας.
import	Καθορίζει μία λίστα από πακέτα ή κλάσεις της γλώσσας Java που θα χρησιμοποιηθούν εντός του κώδικα μίας JSP Ιστοσελίδας.

info	Καθορίζει ένα αλφαριθμητικό το οποίο μπορεί να είναι προσβάσιμο μέσω της μεθόδου <code>getServletInfo()</code> της παραγόμενης Java Servlet κατά την υλοποίηση μίας JSP Ιστοσελίδας.
isELIgnored	Καθορίζει, αν μία JSP έκφραση θα αγνοηθεί ή όχι σε μία JSP Ιστοσελίδα. .
isErrorPage	Υποδεικνύει, αν η URL της εκάστοτε τρέχουσας JSP Ιστοσελίδας, καθορίζεται από την ιδιότητα <code>errorPage</code> μίας άλλης JSP Σελίδας.
isScriptingEnabled	Καθορίζει, αν τεχνικές προγραμματισμού βασισμένες στα σενάρια είναι διαθέσιμες για τη σύνταξη του κώδικα μίας JSP Ιστοσελίδας.
isThreadSafe	Καθορίζει το νηματικό μοντέλο της παραγόμενης Java Servlet κατά την υλοποίηση μίας JSP Ιστοσελίδας.
language	Καθορίζει τη γλώσσα προγραμματισμού που χρησιμοποιείται για τη σύνταξη του κώδικα μίας JSP Ιστοσελίδας.
session	Καθορίζει, αν μία JSP Ιστοσελίδα, συμμετέχει ή όχι σε μία HTTP συνεδρία.

Πίνακας 35: Οι πιο βασικές ιδιότητες της κατευθυντήριας οδηγίας `page` της JSP

Δεύτερον, η κατευθυντήρια οδηγία `include` της τεχνικής JSP, με στόχο να κατευθύνει τον JSP container να συνενώνει το περιεχόμενο άλλων εξωτερικών αρχείων με την σε κάθε περίπτωση τρέχουσα JSP Ιστοσελίδα κατά τη διάρκεια της φάσης μετάφρασης αυτής της JSP Σελίδας. Η συγκεκριμένη κατευθυντήρια οδηγία μπορεί να μπει οπουδήποτε μέσα στον κώδικα μίας JSP Σελίδας και φέρει την παρακάτω βασική σύνταξη

```
<%@ include file="relative url" >
```

Να τονιστεί, εδώ, πως η ιδιότητα `file` που εμπεριέχεται στην προηγούμενη σύνταξη είναι στην πραγματικότητα μία σχετική URL. Αν καθοριστεί ένα αρχείο χωρίς συσχετιζόμενο μονοπάτι, τότε ο JSP μεταγλωττιστής υποθέτει ότι το συγκεκριμένο αρχείο είναι στον κάθε φορά ίδιο κατάλογο με την εκάστοτε τρέχουσα JSP Ιστοσελίδα.

Η ισάξια, τώρα, XML σύνταξη της ανώτερης οδηγίας είναι η παρακάτω:

```
<jsp:directive.include file="relative url" />
```

Τρίτον, η κατευθυντήρια οδηγία `include` της τεχνικής JSP εκμεταλλεύεται, για να δηλώνει ότι η κάθε φορά τρέχουσα JSP Ιστοσελίδα εκμεταλλεύεται μία σειρά από ειδικές ετικέτες που μοιάζουν με HTML ή XML ετικέτες και ορίζονται από τον εκάστοτε χρήστη, ώστε να εξυπηρετούν συγκεκριμένους σκοπούς. Η τρέχουσα κατευθυντήρια οδηγία μπορεί να εισέρθει σε οποιοδήποτε σημείο του κώδικα μίας JSP Σελίδας και έχει την ακόλουθη κύρια σύνταξη

```
<%@ taglib uri="uri" prefix="prefixOfTag" >
```

Σε τούτο το σημείο, να σημειωθεί ότι η ιδιότητα `uri` που περιλαμβάνεται στην πρωτότερη σύνταξη επιλύει μία τοποθεσία στον Παγκόσμιο Ιστό που ο JSP container αντιλαμβάνεται, ενώ η ιδιότητα `prefix` ενημερώνει τον JSP container για τις ετικέτες σήμανσης που ορίζουν εξειδικευμένες JSP ενέργειες.

Κλείνοντας, η ισοδύναμη XML σύνταξη της πιο πάνω οδηγίας είναι η παρακάτω:

```
<jsp:directive.taglib uri="uri" prefix="prefixOfTag" />
```

[66] [67]

2.18.6 Οι Ενέργειες των JSP

Οι ενέργειες-δράσεις που απαντώνται στην τεχνική JSP (JSP actions) χρησιμοποιούν δομές προγραμματισμού βασισμένες σε XML σύνταξη, με σκοπό να ελέγχουν τη συμπεριφορά μίας Java Servlet, η οποία παράγεται κατά τη φάση υλοποίησης μίας JSP Ιστοσελίδας. Τέτοιου είδους ενέργειες δίνουν τη δυνατότητα στους προγραμματιστές JSP Σελίδων να εισάγουν δυναμικά ένα αρχείο σε μία

JSP Σελίδα, να προωθούν το χρήστη σε μία άλλη ξεχωριστή JSP Σελίδα από την εκάστοτε τρέχουσα JSP Σελίδα, να επαναχρησιμοποιούν JavaBeans στοιχεία σε μία JSP Σελίδα, να παράγουν HTML κώδικα για τη δημιουργία πρόσθετων για μία JSP Σελίδα και σε γενικές γραμμές να υλοποιούν μία ακολουθία από σχετικές με τις προαναφερόμενες δράσεις.

Η σύνταξη ενός JSP στοιχείου ενέργειας-δράσης σύμφωνα με το πρότυπο XML είναι η παρακάτω:

```
<jsp:action_name attribute="value"/>
```

Τα JSP στοιχεία δράσεων είναι στην ουσία προκαθορισμένες συναρτήσεις που επιτελούν μία συγκεκριμένη ενέργεια και παρουσιάζονται στον παρακάτω Πίνακα:

Στοιχείο δράσης	Λειτουργία
jsp:attribute	Ορίζει δυναμικά χαρακτηριστικές ιδιότητες προκαθορισμένων XML στοιχείων εντός του κώδικα μίας JSP Ιστοσελίδας.
jsp:body	Ορίζει δυναμικά τη δομή του κώδικα μίας JSP Ιστοσελίδας. με βάση προκαθορισμένα XML στοιχεία.
jsp:element	Ορίζει δυναμικά XML στοιχεία εντός του κώδικα μίας JSP Ιστοσελίδας.
jsp:forward	Προωθεί τον εκάστοτε χρήστη που αιτείται περιεχόμενο από την εκάστοτε τρέχουσα JSP Ιστοσελίδα σε μία άλλη νέα JSP Ιστοσελίδα.
jsp:getProperty	Εισάγει την ιδιότητα ενός JavaBean στοιχείου στον κώδικα της ροής εξόδου μίας JSP Ιστοσελίδας, που αποστέλλεται στον εκάστοτε χρήστη, ο οποίος αιτείται περιεχόμενο από αυτήν την JSP Σελίδα.
jsp:include	Συμπεριλαμβάνει τις επιμέρους JSP Ιστοσελίδες από τις οποίες αποτελείται μία κεντρική JSP Ιστοσελίδα.
jsp:plugin	Δημιουργεί ειδικό κώδικα για ένα πρόγραμμα περιήγησης, ώστε να είναι σε θέση αυτό να παράγει κατάλληλα Java πρόσθετα για μία JSP Ιστοσελίδα.
jsp:setProperty	Καθορίζει την ιδιότητα ενός JavaBean στοιχείου εντός του κώδικα μίας JSP Ιστοσελίδας.
jsp:text	Δημιουργεί το πρότυπο κείμενο μίας JSP Ιστοσελίδας.
jsp:useBean	Βρίσκει ή αρχικοποιεί ένα JavaBean στοιχείο εντός του κώδικα μίας JSP Ιστοσελίδας.

Πίνακας 36: Τα στοιχεία δράσης της JSP

Υπάρχουν δύο χαρακτηριστικά γνωρίσματα που είναι κοινά σε όλα τα JSP στοιχεία δράσης σ: Η ιδιότητα id και η ιδιότητα scope. Στη συνέχεια αναλύονται αυτές οι δύο ιδιότητες:

- **Id:** Προσδιορίζει με μοναδικό τρόπο ένα JSP στοιχείο δράσης και επιτρέπει την αναφορά στο στοιχείο αυτό μέσα στον κώδικα μίας JSP Ιστοσελίδας.
- **Scope:** Προσδιορίζει τον κύκλο ζωής ενός JSP στοιχείου δράσης.

Είναι ευκολονόητο πως οι προαναφερόμενες ιδιότητες id και scope των JSP στοιχείων δράσης είναι άμεσα συσχετιζόμενες, από τη στιγμή που η ιδιότητα scope καθορίζει τη διάρκεια ζωής ενός JSP αντικείμενου που συνδέεται με την ιδιότητα id μέσα στον κώδικα μίας JSP Ιστοσελίδας

Ακολούθως θα μελετηθούν με μεγαλύτερη λεπτομέρεια τα JSP στοιχεία δράσης που παρουσιάστηκαν στον παραπάνω Πίνακα.

Πρώτον, τα JSP στοιχεία δράσης jsp:attribute, jsp:body και jsp:element διατίθενται για το δυναμικό ορισμό των στοιχείων μίας JSP Ιστοσελίδας, των οποίων η δομή βασίζεται στη γλώσσα XML. Ο προσδιορισμός δυναμικός έχει ιδιαίτερη σημασία, διότι σημαίνει ότι τα όποια XML στοιχεία της προαναφερόμενης JSP Σελίδας παράγονται τη στιγμή που εκάστοτε πελάτης ζητάει περιεχόμενο από αυτή τη JSP Σελίδα και όχι στατικά, την ώρα της μεταγλώττισης του κώδικά της.

Δεύτερον, το JSP στοιχείο δράσης jsp:forward τερματίζει τη δράση της εκάστοτε τρέχουσας JSP Ιστοσελίδας και προωθεί το αίτημα του εκάστοτε πελάτη που ζητάει από αυτή περιεχόμενο σε άλλη πηγή πόρων, όπως είναι, για παράδειγμα, μία άλλη JSP Σελίδα ή μία άλλη στατική Σελίδα.

Η σύνταξη του παρόντος JSP στοιχείου δράσης έχει ως εξής:

```
<jsp:forward page="Relative URL" />
```

Η μοναδική ιδιότητα που συνδέεται με το JSP στοιχείο δράσης `jsp:forward` παρατίθεται στον παρακάτω Πίνακα:

Ιδιότητα	Περιγραφή
page	Πρόκειται για τη URL μίας διαφορετικής πηγής πόρων, πέρα από την εκάστοτε τρέχουσα JSP Ιστοσελίδα, εφόσον τερματιστεί η δράση της προαναφερόμενης JSP Σελίδας.

Πίνακας 37: Η ιδιότητα του JSP στοιχείου δράσης `jsp:forward`

Τρίτον, το JSP στοιχείο δράσης `jsp:getProperty` χρησιμοποιείται αρχικά, για να ανακτήσει την τιμή μίας δοσμένης ιδιότητας ενός προκαθορισμένου JavaBean στοιχείου που υπάρχει εντός του κώδικα της εκάστοτε JSP Ιστοσελίδας, η οποία περιέχει το παρόν JSP στοιχείο δράσης και από την οποία ο εκάστοτε πελάτης αιτείται περιεχόμενο.

Η σύνταξη του τρέχοντος JSP στοιχείου δράσης έχει όπως παρακάτω:

```
<jsp:useBean id="myName" ... />
```

```
.....  
<jsp:getProperty name="myName" property="someProperty" ... />
```

Οι πιο κύριες ιδιότητες που σχετίζονται με το JSP στοιχείο δράσης `jsp:getProperty` φαίνονται στον παρακάτω Πίνακα:

Ιδιότητα	Περιγραφή
name	Πρόκειται για το όνομα ενός JavaBean στοιχείου εντός του κώδικα μίας JSP Ιστοσελίδας. Ωστόσο, το προαναφερόμενο JavaBean στοιχείο πρέπει πρωτίτερα να έχει οριστεί.
property	Πρόκειται για την ιδιότητα ενός JavaBean στοιχείου εντός του κώδικα μίας JSP Ιστοσελίδας.

Πίνακας 38: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης `jsp:getProperty`

Τέταρτον, το JSP στοιχείο δράσης `jsp:include` χρησιμοποιείται, με σκοπό να εισάγει τις επιμέρους JSP Ιστοσελίδες της εκάστοτε κύριας JSP Ιστοσελίδας, που περιλαμβάνει το παρόν JSP στοιχείο δράσης μέσα στον κώδικά της και από την οποία ο εκάστοτε πελάτης αιτείται περιεχόμενο.

Η σύνταξη του συγκεκριμένου JSP στοιχείου δράσης έχει όπως ακολούθως:

```
<jsp:include page="relative URL" flush="true" />
```

Σε αντίθεση με την JSP κατευθυντήρια οδηγία `include`, που εισάγει επιμέρους JSP Ιστοσελίδες στην εκάστοτε τρέχουσα κεντρική JSP Ιστοσελίδα, τη στιγμή που η τελευταία μεταφράζεται σε Java Servlet, ανεξάρτητα από το πόσο ο εκάστοτε πελάτης αιτείται από αυτή περιεχόμενο, το JSP στοιχείο δράσης `jsp:include` εισάγει επιμέρους JSP Ιστοσελίδες στην εκάστοτε τρέχουσα κεντρική JSP Ιστοσελίδα, ακριβώς τη χρονική στιγμή που ο εκάστοτε πελάτης ζητάει από αυτή περιεχόμενο.

Οι πιο αντιπροσωπευτικές ιδιότητες που είναι αλληλένδετες με το JSP στοιχείο δράσης `jsp:include` φαίνονται στον παρακάτω Πίνακα:

Ιδιότητα	Περιγραφή
page	Πρόκειται για τη URL κάθε επιμέρους JSP Ιστοσελίδας που αποτελεί ένα δομικό λίθο μίας κεντρικής JSP Ιστοσελίδας.

flush	Η λογική ιδιότητα που καθορίζει, αν καθεμία από επιμέρους JSP Ιστοσελίδες που συνιστά διαθρωτικό συστατικό μίας κύρια JSP Ιστοσελίδας, έχει διαθέσιμο ενδιάμεσο αποθηκευτικό χώρο και ενδιάμεση μνήμη, προτού καθεμία από τις παραπάνω μεμονωμένες JSP Ιστοσελίδες δομήσει την προαναφερόμενη βασική JSP Ιστοσελίδα.
-------	--

Πίνακας 39: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης jsp:include

Πέμπτον, το JSP στοιχείο δράσης jsp:plugin διατίθεται, με στόχο να εισάγει Java πρόσθετα σε μία JSP Ιστοσελίδα, εντός του κώδικα της οποίας βρίσκεται το συγκεκριμένο JSP στοιχείο δράσης και από όπου ο εκάστοτε χρήστης ζητάει περιεχόμενο. Επίσης, το τρέχον JSP στοιχείο δράσης καθορίζει τον τύπο του κατάλληλου σε κάθε περίπτωση προγράμματος περιήγησης στο Διαδίκτυο και εισάγει τις HTML ετικέτες <object> και <embed> σε οποιοδήποτε σημείο του κώδικα της πρωτύτης JSP Σελίδας κρίνεται απαραίτητο. Αν το απαιτούμενο κάθε φορά πρόσθετο δεν είναι αναγκαίο στην εκάστοτε τρέχουσα JSP Σελίδα, τότε ο εκάστοτε φυλλομετρητής το κατεβάζει, ώστε να το έχει διαθέσιμο, όταν χρειαστεί να το εκτελέσει. Τα προαναφερόμενα πρόσθετα είναι δυνατό να είναι είτε JavaBeans στοιχεία είτε Java Applets.

Ένα παράδειγμα σύνταξης αυτού του JSP στοιχείου δράσης είναι το ακόλουθο:

```
<jsp:plugin type="applet" codebase="dirname" code="MyApplet.class"width="60" height="80">
  <jsp:param name="fontcolor" value="red" />
  <jsp:param name="background" value="black" />

  <jsp:fallback>
    // Unable to initialize Java Plugin
  </jsp:fallback>

</jsp:plugin>
```

Όσον αφορά, τώρα, το JSP στοιχείο jsp:param χρησιμοποιείται, προκειμένου να στέλνει παραμέτρους σε ένα JavaBean στοιχείο που περιέχεται μέσα στον κώδικα της εκάστοτε JSP Ιστοσελίδας, που περιλαμβάνει το παραπάνω JSP στοιχείο δράσης ή σε μία Java Applet, ενώ το JSP στοιχείο jsp:fallback χρησιμοποιείται να καθορίσει ένα αλφαριθμητικό που περιέχει ένα μήνυμα σφάλματος, το οποίο στέλνεται στον εκάστοτε χρήστη που ζητάει από την προαναφερόμενη JSP Σελίδα περιεχόμενο, εφόσον το σε κάθε φάση διαθέσιμο Java πρόσθετο αποτύχει να επιτελέσει τη λειτουργία για την οποία προορίζεται.

Επιπρόσθετα, το JSP στοιχείο δράσης jsp:plugin έχει ποικίλες χαρακτηριστικές ιδιότητες που ανταποκρίνονται σε κοινότητες HTML ετικέτες, οι οποίες είναι εκμεταλλεύσιμες, για να διαμορφώσουν διάφορα συστατικά της γλώσσας Java που είναι τοποθετημένα εντός του κώδικα της εκάστοτε JSP Ιστοσελίδας που περιλαμβάνει το παραπάνω JSP στοιχείο δράσης.

Έκτον, το JSP στοιχείο δράσης jsp:setProperty είναι εκμεταλλεύσιμο, ώστε να καθορίζει τις ιδιότητες των προκαθορισμένων JavaBean στοιχείων που υφίστανται στην εκάστοτε JSP Ιστοσελίδα, μέσα στον κώδικα της οποίας περικλείεται το παρόν JSP στοιχείο δράσης και από την οποία ο εκάστοτε χρήστης ζητάει περιεχόμενο.

Υπάρχουν δύο βασικοί τρόποι χρήσης του συγκεκριμένου JSP στοιχείου δράσης.

Πρώτον, έξω από το σώμα του JSP στοιχείου δράσης jsp:useBean σύμφωνα με τη σύνταξη:

```
<jsp:useBean id="myName" ... />
.....
<jsp:setProperty name="myName" property="someProperty" ... />
```

Στην περίπτωση αυτή το JSP στοιχείο δράσης jsp:setProperty εκτελείται ανεξάρτητα από την αρχικοποίηση ενός νέου JavaBean στοιχείου ή την ανακάλυψη ενός υπάρχοντος.

Δεύτερον, εντός του σώματος του JSP στοιχείου δράσης jsp:useBean κατά τη σύνταξη:

```
<jsp:useBean id="myName" ... >
```

```
.....  
<jsp:setProperty name="myName" property="someProperty" ... />  
</jsp:useBean>
```

Στη συγκεκριμένη περίπτωση το JSP στοιχείο δράσης `jsp:setProperty` εκτελείται μόνο αν αρχικοποιηθεί ένα καινούργιο `JavaBean` στοιχείο και όχι αν βρεθεί ένα υπάρχον.

Στον πιο κάτω Πίνακα παρουσιάζονται οι πιο χαρακτηριστικές ιδιότητες που αφορούν το JSP στοιχείο δράσης `jsp:setProperty`:

Ιδιότητα	Περιγραφή
name	Πρόκειται για το όνομα ενός <code>JavaBean</code> στοιχείου του οποίου θα καθοριστεί μία ιδιότητα εντός του κώδικα μίας JSP Ιστοσελίδας. Ωστόσο, το προαναφερόμενο <code>JavaBean</code> στοιχείο πρέπει πρωτίτερα να έχει οριστεί.
property	Πρόκειται για την ιδιότητα ενός <code>JavaBean</code> στοιχείου η οποία θα καθοριστεί εντός του κώδικα μίας JSP Ιστοσελίδας.
value	Πρόκειται για την τιμή η οποία θα καταχωρηθεί στην ιδιότητα ενός <code>JavaBean</code> στοιχείου εντός του κώδικα μίας JSP Ιστοσελίδας. Αν, τώρα, η προαναφερόμενη τιμή είναι <code>null</code> ή δε βρεθεί, τότε το JSP στοιχείο δράσης <code>jsp:getProperty</code> παραλείπεται.
param	Πρόκειται για μία παράμετρο με βάση την οποία θα ανακτηθεί η τιμή μίας ιδιότητας ενός <code>JavaBean</code> στοιχείου εντός του κώδικα μίας JSP Ιστοσελίδας. Ωστόσο, δεν επιτρέπεται ταυτόχρονη χρήση των ιδιοτήτων <code>value</code> και <code>param</code> , παρόλο είναι δυνατή η παράλληλη παράκαμψή τους.

Πίνακας 40: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης `jsp:setProperty`

Έβδομον, το JSP στοιχείο δράσης `jsp:text` χρησιμοποιείται, στοχεύοντας στη συγγραφή πρότυπου κείμενου σε μία JSP Ιστοσελίδα, εντός του κώδικα της οποίας περικλείεται το παρόν JSP στοιχείο δράσης και από την οποία ο εκάστοτε χρήστης ζητάει περιεχόμενο.

Η σύνταξη του τρέχοντος JSP στοιχείου δράσης είναι η πιο κάτω:

```
<jsp:text>Template data</jsp:text>
```

Το πρότυπο αυτό κείμενο δε δύναται να περιλαμβάνει άλλα στοιχεία, πέρα από κείμενο και JSP εκφράσεις.

Όγδοον, το JSP στοιχείο δράσης `jsp:useBean` είναι αρκετά πολύπλευρο. Αρχικά ψάχνει να βρει ένα υπάρχον `JavaBean` στοιχείο με συγκεκριμένες τις ιδιότητές του `id` και `scope`. Αν δε βρεθεί ένα αντικείμενο που να ανταποκρίνεται στα πιο πάνω κριτήρια αναζήτησης, τότε το εν λόγω JSP στοιχείο δράσης προσπαθεί να δημιουργήσει ένα τέτοιο `JavaBean` στοιχείο.

Ο απλούστερος τρόπος για να φορτωθεί ένα `JavaBean` στοιχείο είναι ο παρακάτω:

```
<jsp:useBean id="name" class="package.class" />
```

Από τη στιγμή, τώρα, που φορτωθεί ένα `JavaBean` στοιχείο, γίνεται εκμετάλλευση των JSP στοιχείων δράσης `jsp:setProperty` και `jsp:getProperty`, προκειμένου να τροποποιηθούν και να ανακτηθούν οι ιδιότητες του προαναφερόμενου στοιχείου.

Στο μετέπειτα Πίνακα καταγράφονται οι βασικότερες ιδιότητες που είναι συνδεδεμένες με το JSP στοιχείο δράσης `jsp:useBean`:

Ιδιότητα	Περιγραφή
package	Ορίζει το όνομα του πλήρους πακέτου που περιέχει το εκάστοτε <code>JavaBean</code> στοιχείο που απαντάται μέσα στον κώδικα μίας JSP Ιστοσελίδας, από την οποία ο εκάστοτε πελάτης αιτείται περιεχόμενο.

type	Ορίζει τον τύπο της μεταβλητής ο οποίος θα αναφερθεί στο εκάστοτε JavaBean στοιχείο που απαντάται μέσα στον κώδικα μιας JSP Σελίδα από την οποία ο εκάστοτε πελάτης αιτείται περιεχόμενο.
beanName	Αποδίδει το όνομα του εκάστοτε Java κόκκου που απαντάται σε μία JSP Σελίδα από την οποία ο εκάστοτε πελάτης αιτείται περιεχόμενο, όπως αυτό καθορίζεται από τη μέθοδο instantiate() της κλάσης java.beans.Beans

Πίνακας 41: Οι πιο βασικές ιδιότητες του JSP στοιχείου δράσης jsp:useBean

[66] [68]

2.18.7 Τα Προκαθορισμένα Αντικείμενα των JSP

Τα προκαθορισμένα αντικείμενα της τεχνικής JSP (JSP implicit objects) είναι αντικείμενα της γλώσσας Java, τα οποία ο JSP container διαθέτει στους προγραμματιστές των JSP Ιστοσελίδων με απευθείας κλήση, χωρίς προγενέστερη ρητή δήλωσή τους εντός του κώδικα αυτών των JSP Σελίδων.

Η τεχνική JSP υποστηρίζει εννέα τύπους προκαθορισμένων αντικειμένων, που ταξινομούνται στον ακόλουθο Πίνακα:

Αντικείμενο	Περιγραφή
application	Είναι το αντικείμενο ServletContext της γλώσσας Java που καταγράφει τον κύκλο ζωής μιας JSP Ιστοσελίδας.
config	Είναι το αντικείμενο ServletConfig της γλώσσας Java που σχετίζεται με την αρχική παραμετροποίηση μιας Java Servlet, η οποία παράγεται κατά την υλοποίηση μιας JSP Ιστοσελίδας.
Exception	Είναι το αντικείμενο Exception της γλώσσας Java που επιτρέπει τα δεδομένα μιας εξαίρεσης που απαντάται εντός του κώδικα μιας JSP Ιστοσελίδας να είναι προσπελάσιμα από προκαθορισμένες JSP Ιστοσελίδες.
out	Είναι το αντικείμενο PrintWriter της γλώσσας Java που χρησιμοποιείται, για να αποστέλλεται έξοδος στον εκάστοτε πελάτη, ο οποίος αιτείται περιεχόμενο από μία JSP Ιστοσελίδα.
page	Είναι απλώς ένα συνώνυμο του αντικειμένου this της γλώσσας Java και χρησιμοποιείται, για να καλεί τις μεθόδους που ορίζονται από μία Java Servlet, η οποία παράγεται κατά την υλοποίηση μιας JSP Ιστοσελίδας.
pageContext	Είναι το προκαθορισμένο αντικείμενο της γλώσσας Java που ενσωματώνει τη χρήση εξειδικευμένων εργαλείων, όσον αφορά την κατασκευή μιας JSP Ιστοσελίδας από την πλευρά του εκάστοτε εξυπηρετητή Ιστού όπως είναι, για παράδειγμα, οι υψηλής απόδοσης JSPWriters.
request	Είναι το αντικείμενο HttpServletRequest της γλώσσας Java που σχετίζεται με ένα HTTP μήνυμα αίτησης του εκάστοτε πελάτη, ο οποίος αιτείται περιεχόμενο από μία JSP Ιστοσελίδα.
response	Είναι το αντικείμενο HttpServletResponse της γλώσσας Java που σχετίζεται με το HTTP μήνυμα απόκρισης του εκάστοτε εξυπηρετητή Ιστού, από τον οποίο αιτείται περιεχόμενο ο εκάστοτε πελάτης μιας JSP Ιστοσελίδας.
session	Είναι το αντικείμενο HttpSession της γλώσσας Java που σχετίζεται με τη συνεδρία που δημιουργείται για ένα HTTP μήνυμα αίτησης του εκάστοτε πελάτη, ο οποίος αιτείται περιεχόμενο από μία JSP Ιστοσελίδα.

Πίνακας 42: Τα προκαθορισμένα αντικείμενα της JSP

Στη συνέχεια θα γίνει μία εκτενέστερη αναφορά στα αντικείμενα της τεχνικής JSP του παραπάνω Πίνακα.

Πρώτον, το προκαθορισμένο αντικείμενο application της τεχνικής JSP είναι ένα στιγμιότυπο της διασύνδεσης javax.servlet.ServletContext. της γλώσσας Java. Το τρέχον αντικείμενο αναπαριστά

μία JSP Ιστοσελίδα καθόλη τη διάρκεια του κύκλου ζωής της. Έτσι, το συγκεκριμένο αντικείμενο δημιουργείται, όταν μία JSP Σελίδα αρχικοποιείται με τη μέθοδο `jspInit()` και καταργείται, μόλις η JSP Σελίδα καταστρέφεται με τη μέθοδο `jspDestroy()`.

Δεύτερον, το προκαθορισμένο αντικείμενο `config` της τεχνικής JSP είναι ένα στιγμιότυπο της διασύνδεσης `javax.servlet.ServletConfig` της γλώσσας Java. Το παρόν αντικείμενο δίνει τη δυνατότητα στους JSP προγραμματιστές να έχουν πρόσβαση στις παραμέτρους αρχικοποίησης της Java Servlet που παράγεται κατά την υλοποίηση μίας JSP Ιστοσελίδας.

Τρίτον, το προκαθορισμένο αντικείμενο `Exception` είναι ένα στιγμιότυπο της κλάσης `java.lang.Exception` της γλώσσας Java. Το εν λόγω αντικείμενο περιέχει τα κατάλληλα δεδομένα, ώστε μία JSP Ιστοσελίδα να είναι ικανή να ανταποκριθεί σε πιθανά σφάλματα που μπορεί να προκύψουν κατά την δημιουργία της.

Τέταρτον, το προκαθορισμένο αντικείμενο `out` της τεχνικής JSP είναι ένα στιγμιότυπο της κλάσης `javax.servlet.jsp.JspWriter` της γλώσσας Java. Το συγκεκριμένο αντικείμενο χρησιμοποιείται, με στόχο να στέλνει περιεχόμενο-κείμενο στον εκάστοτε χρήστη που το αιτείται από μία JSP Ιστοσελίδα. Η αρχικοποίηση του αντικειμένου `JspWriter` δεν είναι ίδια σε όλες τις περιπτώσεις, αλλά εξαρτάται από τη δυνατότητα κάθε JSP Σελίδας για ενδιάμεσο χώρο αποθήκευσης και ενδιάμεση μνήμη. Το αντικείμενο αυτό, τώρα, περιλαμβάνει ως επί το πλείστον τις ίδιες μεθόδους με την κλάση `java.io.PrintWriter`, οι κυριότερες από τις οποίες φαίνονται στον πιο κάτω Πίνακα:

Μέθοδος	Λειτουργία
<code>flush()</code>	Κατακλύζει τη ροή εξόδου μίας JSP Ιστοσελίδας με το κατάλληλο περιεχόμενο που αιτείται ο εκάστοτε χρήστης από αυτή τη JSP Σελίδα.
<code>print(dataType dt)</code>	Τυπώνει την τιμή μίας μεταβλητής που περιέχεται στον κώδικα μίας JSP Ιστοσελίδας και η οποία περνάει ως όρισμα στη μέθοδο.
<code>println(dataType dt)</code>	Τυπώνει την τιμή μίας μεταβλητής που περιέχεται στον κώδικα μίας JSP Ιστοσελίδας και η οποία περνάει ως όρισμα στη μέθοδο και μετακινεί το δρομέα στην αμέσως επόμενη γραμμή εντός του κώδικα της προαναφερόμενης JSP Σελίδας .

Πίνακας 43: Οι πιο βασικές μέθοδοι του αντικειμένου `JspWriter` της JSP

Πέμπτον, το προκαθορισμένο αντικείμενο `page` της τεχνικής JSP είναι στην πραγματικότητα ένα συνώνυμο του αντικειμένου `this`...

Έκτον, το προκαθορισμένο αντικείμενο `pageContext` είναι ένα στιγμιότυπο της κλάσης `javax.servlet.jsp.PageContext` της γλώσσας Java. Το συγκεκριμένο αντικείμενο διατίθεται, για να αναπαριστά ολόκληρο το περιεχόμενο μίας JSP Ιστοσελίδας παρακάμπτοντας διάφορες τεχνικές λεπτομέρειες. Επιπρόσθετα, το πρωτότερο αντικείμενο αποθηκεύει αναφορές σχετικές με τα αντικείμενα των HTTP μηνυμάτων αίτησης και απόκρισης που δημιουργούνται, από τη στιγμή ένας πελάτης ζητάει περιεχόμενο από μία JSP Σελίδα. Έτσι, παίζει σπουδαίο ρόλο στη διαμόρφωση των αντικειμένων `ServletConfig` `ServletContext`, `PrintWriter` της γλώσσας Java και εμπεριέχει πληροφορίες για τις κατευθυντήριες οδηγίες που πιθανόν να υπάρχουν σε μία JSP Σελίδα.

Από την πλευρά της, τώρα, η κλάση `PageContext` υποστηρίζει αρκετά πεδία, συμπεριλαμβανομένων των `page_scope`, `request_scope`, `session_scope`, και `application_scope` και πάνω από 40 μεθόδους, οι μισές από τις οποίες κληρονομούνται από την κλάση `javax.servlet.jsp.JspContext`. Μία από τις πιο σημαντικές μεθόδους της συγκεκριμένης κλάσης είναι η `removeAttribute()`, η οποία συντάσσεται ως εξής:

```
pageContext.removeAttribute("attrName",scopeName);
```

Όπου:

- `attrName`: Υποχρεωτική. Η χαρακτηριστική ιδιότητα που θα αφαιρεθεί από τα περιεχόμενα μίας JSP Σελίδας.

- `scopeName`: Προαιρετική. Το πεδίο μίας JSP Σελίδας, από το οποίο θα αφαιρεθεί η `attrName`. Αν η παράμετρος αυτή παραλείπεται, τότε η μέθοδος αφαιρεί την `attrName` από όλα τα πεδία της JSP Σελίδας.

Έβδομον, το προκαθορισμένο αντικείμενο `HttpServletRequest` είναι ένα στιγμιότυπο του αντικειμένου `javax.servlet.http.HttpServletRequest`. Κάθε φορά που ο εκάστοτε πελάτης αιτείται περιεχόμενο μέσω του φυλλομετρητή του από ένα διακομιστή μίας JSP Σελίδας, υλοποιείται ένα νέο αντικείμενο `HttpServletRequest`, που αναπαριστά αυτή την αίτηση. Το συγκεκριμένο αντικείμενο παρέχει μεθόδους, προκειμένου να ανακτώνται πληροφορίες από τις κεφαλίδες των εκάστοτε HTTP μηνυμάτων αίτησης, οι οποίες έχουν να κάνουν κυρίως με δεδομένα φορμών ή cookies. Οι σημαντικότερες μέθοδοι αυτού του αντικειμένου θα αναπτυχθούν λεπτομερώς στη συνέχεια.

Όγδοον, το προκαθορισμένο αντικείμενο `HttpServletResponse` είναι ένα στιγμιότυπο του αντικειμένου `javax.servlet.http.HttpServletResponse`. Για κάθε αντικείμενο `HttpServletRequest`, που δημιουργείται, όπως μόλις προαναφέρθηκε, αφότου ο εκάστοτε πελάτης στέλνει ένα HTTP μήνυμα αίτησης σε ένα διακομιστή Ιστοσελίδων, δημιουργείται και ένα HTTP μήνυμα απάντησης, το οποίο αναπαριστάται από ένα αντικείμενο `HttpServletResponse`. Μέσω του αντικειμένου αυτού οι JSP προγραμματιστές είναι σε θέση να προσθέτουν στις JSP νέα cookies ή σημάνσεις ημερομηνιών, HTTP κωδικούς κατάστασης και πολλές άλλες χρήσιμες πληροφορίες. Οι πιο βασικές μέθοδοι αυτού του αντικειμένου θα παρουσιαστούν εκτενώς στη συνέχεια.

Ένατον, το προκαθορισμένο αντικείμενο `HttpSession` είναι ένα στιγμιότυπο του αντικειμένου `javax.servlet.http.HttpSession` και συμπεριφέρεται, όπως ακριβώς συμπεριφέρεται ένα αντικείμενο συνόδου στις Java Servlets, έχοντας ως κύρια λειτουργία του να ανιχνεύει μία σύνοδο που δημιουργείται από έναν πελάτη, εφόσον αυτός αποστέλλει μία αίτηση.

[66] [67] [68]

2.18.8 Οι Φόρμες των JSP

Για τις φόρμες που τυχόν απαντώνται στις JSP Ιστοσελίδες ισχύει ότι και για τις φόρμες που απαντώνται σε HTML έγγραφα στα οποία γίνεται χρήση της γλώσσας JavaScript (και οι οποίες περιγράφηκαν παραπάνω). Ωστόσο, στο σημείο τούτο, αξίζει να τονιστεί πως οι JSP Σελίδες χειρίζονται αυτόματα τα δεδομένα των φορμών, που τυχόν συναντώνται σε αυτές, με τη χρήση των παρακάτω σημαντικότερων JSP μεθόδων:

- `getParameter()`: Καλείται, για να ληφθεί η τιμή μίας παραμέτρου μίας φόρμας μίας JSP Ιστοσελίδας.
- `getParameterValues()`: Καλείται, όταν μία παράμετρος μίας φόρμας εμφανίζεται πάνω από μία φορά σε μία JSP Ιστοσελίδα και επιστρέφει πολλαπλές τιμές, όπως, για παράδειγμα, συμβαίνει με ένα πλαίσιο ελέγχου.
- `getParameterNames()`: Καλείται, όταν είναι επιθυμητό να ληφθεί μία πλήρη λίστα όλων των παραμέτρων που υπάρχουν σε μία φόρμα μίας JSP Ιστοσελίδας.
- `getInputStream()`: Καλείται, προκειμένου να διαβαστούν οι ροές εισόδου δυαδικών δεδομένων που καταχωρεί ο εκάστοτε χρήστης σε μία φόρμα μίας JSP Ιστοσελίδας.

[66]

2.18.9 Σύγκριση των JSP

Η τεχνική JSP συχνά εξυπηρετεί τον ίδιο σκοπό με τα προγράμματα που υλοποιούνται με τη χρήση σεναρίων CGI. Ωστόσο, η JSP προσφέρει μερικά πλεονεκτήματα σε σχέση με τα CGI, τα οποία είναι τα εξής:

- Η απόδοση των διαδικτυακών εφαρμογών είναι σαφώς καλύτερη με τη χρήση της τεχνικής JSP, διότι η JSP επιτρέπει την ενσωμάτωση δυναμικών στοιχείων σε HTML έγγραφα, δίχως τη χρήση ξεχωριστών αρχείων, όπως συμβαίνει με τα CGI.
- Οι JSP Ιστοσελίδες μεταγλωττίζονται πάντα προτού επεξεργαστούν από τον σε κάθε περίπτωση διακομιστή Ιστού, σε αντίθεση με τα σεναρία CGI και τη γλώσσα προγραμματισμού PERL, που απαιτούν από τον υφιστάμενο κάθε φορά εξυπηρετητή να φορτώσει το κατάλληλο σε κάθε περίπτωση σεναριο, όποτε ο εκάστοτε χρήστης αιτείται περιεχόμενο από την εκάστοτε Σελίδα.
- Η τεχνική JSP είναι δομημένη στην κορυφή των Java Servlets APIs, οπότε, όπως και οι Java

Servlets, η JSP έχει, επίσης, πρόσβαση σε όλα τις ισχυρές APIs της γλώσσας Java, όπως είναι, για παράδειγμα η JDBC, κάτι που φυσικά δε συμβαίνει με τα σενάρια CGI.

- Οι JSP μπορούν να χρησιμοποιηθούν σε συνδυασμό με τις Java Servlets που χειρίζονται το “business logic” κομμάτι μίας εφαρμογής του Διαδικτύου, το οποίο υποστηρίζεται από τις μηχανές-πρότυπα των Java Servlets, φαινόμενο που δεν απαντάται στα σενάρια CGI.
- Η τεχνική JSP αποτελεί αναπόσπαστο κομμάτι της Java Platform Enterprise Edition (Java EE), μίας πλήρως ολοκληρωμένης πλατφόρμα λογισμικού για πανίσχυρες εφαρμογές Διαδικτύου και όχι μόνο. Αυτό συνεπάγεται πως η JSP μπορεί να διαδραματίσει ιδιαίζουσας σημασίας ρόλο από τις πιο απλές ως τις πιο σύνθετες και απαιτητικές εφαρμογές του Παγκόσμιου Ιστού.

Επιπρόσθετα, πέρα από την υπεροχή της JSP έναντι των σεναρίων CGI, η συγκεκριμένη τεχνική υπερτερεί και σε σχέση με άλλες σύγχρονες τεχνολογίες προγραμματισμού Διαδικτύου, όπως αναλύεται παρακάτω:

- Σε σχέση με τις Active Server Pages (ASP): Τα πλεονεκτήματα της τεχνικής JSP σε αυτήν την περίπτωση είναι τα ακόλουθα δύο:
 - I. Το δυναμικό κομμάτι των JSP Ιστοσελίδων είναι γραμμένο σε Java κώδικα, οπότε είναι πιο ισχυρό και ευκολότερο στη χρήση από το αντίστοιχο των ASP Ιστοσελίδων, που είναι γραμμένο σε Visual Basic κώδικα.
 - II. Το δυναμικό κομμάτι των JSP Ιστοσελίδων είναι φορητό και σε άλλα λειτουργικά συστήματα, εκτός των Microsoft Windows, και σε άλλους διακομιστές Ιστού που δεν είναι της Microsoft, κάτι που δε συμβαίνει με τις ASP Ιστοσελίδες.
- Σε σχέση με τις «Καθαρές» (Pure) Java Servlets: Είναι πιο βολικό να γράψει και να τροποποιήσει κάποιος τυπικό HTML κώδικα σε σύγκριση με ένα κώδικα αποτελούμενο από πληθώρα δηλώσεων και εντολών, όπως είναι αυτός που απαντάται στις Java Servlets.
- Σε σχέση με τη διερμηνευόμενη από την πλευρά του εκάστοτε διακομιστή Ιστού γλώσσα σεναρίων Server-Side Includes (SSI): Η SSI στοχεύει στην υλοποίηση απλών εφαρμογών και όχι στην ανάπτυξη «πραγματικών» εφαρμογών Διαδικτύου, οι οποίες έχουν σύνθετες απαιτήσεις, όπως είναι, για παράδειγμα, ή χρησιμοποίηση δεδομένων φορμών, η επικοινωνία με βάσεις δεδομένων και άλλες παρόμοιες με τις προηγούμενες δράσεις, που απαντώνται κατά κόρον στις JSP Ιστοσελίδες.
- Σε σχέση με τη γλώσσα JavaScript: Η JavaScript είναι ικανή να παράγει δυναμικό HTML, κώδικα μόνο όσον αφορά την πλευρά του εκάστοτε χρήστη, αλλά σχεδόν αδυνατεί να αλληλεπιδρά με τον υφιστάμενο σε κάθε περίπτωση εξυπηρετητή Ιστού, προκειμένου να διεκπεραιώνει σύνθετα καθήκοντα, όπως είναι, για παράδειγμα, η πρόσβαση σε βάσεις δεδομένων, η επεξεργασία εικόνων και άλλες όμοιες με τις προαναφερόμενες ενέργειες, οι οποίες, όμως, συνιστούν κύριο γνώρισμα της JSP τεχνικής.
- Σε σχέση με τη γλώσσα HTML: Η τυπική HTML είναι στατική και δεν εμφανίζει δυναμικό χαρακτήρα, σε αντίθεση με την τεχνική JSP που υποστηρίζει τη δημιουργία δυναμικών Ιστοσελίδων.

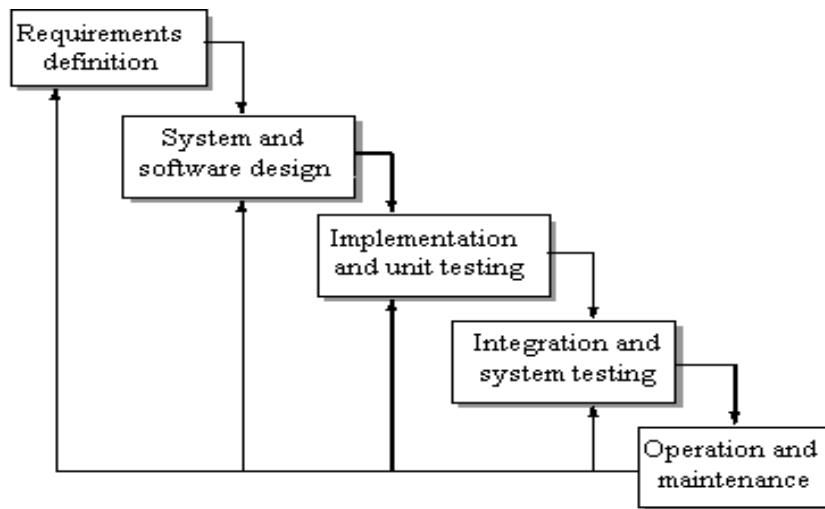
[66] [67] [68]

3 Η Παρουσίαση του Πρακτικού Κομματιού της Πτυχιακής Εργασίας

Στο κεφάλαιο τούτο θα γίνει πρακτική εφαρμογή του θεωρητικού υποβάθρου που αναπτύχθηκε λεπτομερέστατα στο προηγούμενο κεφάλαιο, με απώτερο σκοπό και στόχο την ορθή ανάπτυξη-σχεδιασμό-υλοποίηση του συστήματος λογισμικού το οποίο πραγματεύεται η παρούσα πτυχιακή διατριβή.

3.1 Το Μοντέλο του Καταρράκτη

Ο κύκλος ζωής της διαδικτυακής εφαρμογής η οποία αναπτύσσεται στην τρέχουσα αναφορά ακολουθεί το κλασικό Μοντέλο του Καταρράκτη (Waterfall Model), το οποίο απεικονίζεται στην πιο κάτω Εικόνα:



Εικόνα 32: Το διάγραμμα του Μοντέλου του Καταρράκτη

Όπως φαίνεται και από την παραπάνω Εικόνα το Μοντέλο του Καταρράκτη περιλαμβάνει τις ακόλουθες φάσεις:

- I. Ανάλυση και Καθορισμός Απαιτήσεων Συστήματος (System Requirements Analysis and Specification): Μέσω συμβουλευτικής διαδικασίας με τους χρήστες ενός συστήματος λογισμικού προσδιορίζονται οι υπηρεσίες που θα παρέχει το εν λόγω σύστημα, οι περιορισμοί και οι στόχοι του.
- II. Σχεδιασμός Συστήματος και Λογισμικού (System and Software Design): Η διαδικασία σχεδιασμού ενός συστήματος λογισμικού χωρίζει τις απαιτήσεις σε απαιτήσεις υλικού και απαιτήσεις λογισμικού. Εγκαθιδρύει μία γενική αρχιτεκτονική συστήματος. Ο σχεδιασμός του εκάστοτε λογισμικού περιλαμβάνει την αναγνώριση και την περιγραφή των σημαντικών μερών ενός συστήματος λογισμικού και των σχέσεων μεταξύ τους.
- III. Υλοποίηση Συστήματος και Έλεγχος Υποομάδων (System Implementation and Modules Testing): Κατά τη διάρκεια αυτού του σταδίου, ο σχεδιασμός ενός συστήματος λογισμικού γίνεται αντιληπτός ως ένα σύνολο προγραμμάτων ή ενοτήτων προγραμμάτων. Ο έλεγχος των επιμέρους ενοτήτων προγραμμάτων ενός συστήματος λογισμικού περιλαμβάνει την επαλή-θευση του γεγονότος ότι κάθε τέτοια ενότητα εκπληρώνει τις προδιαγραφές της.
- IV. Ενσωμάτωση και Έλεγχος Συστήματος (Integration and System Testing): Οι ανεξάρτητες ενότητες των προγραμμάτων ενός συστήματος λογισμικού συγχωνεύονται και ελέγχονται ως ένα ολοκληρωμένο πρόγραμμα, για να διασφαλιστεί ότι έχουν εκπληρωθεί οι απαιτήσεις του προαναφερόμενου συστήματος. Μετά τη ενσωμάτωση των υπομονάδων ενός συστήματος λογισμικού, ακολουθεί ο έλεγχος αυτού του συστήματος και τελικά το εν λόγω σύστημα παραδίδεται στον εκάστοτε πελάτη.

- V. Λειτουργία και Συντήρηση Συστήματος (Operation and System Maintenance): Κανονικά, αλλά όχι απαραίτητα, αυτή είναι η πιο χρονοβόρα φάση του κύκλου ζωής ενός συστήματος λογισμικού, κατά την οποία ένα τέτοιο σύστημα εγκαθίσταται και μπαίνει σε χρήση στην πράξη. Η συντήρηση περιλαμβάνει τη διόρθωση λαθών ενός συστήματος λογισμικού τα οποία δεν έγιναν αντιληπτά σε προηγούμενα στάδια του κύκλου ζωής του, βελτιώνοντας την εφαρμογή των διαφόρων ενοτήτων του εν λόγω συστήματος και τις υπηρεσίες που αυτό προσφέρει, αφού ανακαλύπτονται νέες απαιτήσεις.

Στο σημείο αυτό, είναι χρήσιμο να παρατηρηθεί ότι σύμφωνα με τις βασικές αρχές του Μοντέλου του Καταρράκτη το αποτέλεσμα κάθε σταδίου του εν λόγω Μοντέλου είναι ένα ή περισσότερα έγγραφα που πρέπει να εγκριθούν. Έτσι, η κάθε φορά επόμενη φάση του τρέχοντος Μοντέλου δεν μπορεί να ξεκινήσει, αν η εκάστοτε προηγούμενη φάση του δεν έχει ολοκληρωθεί. Στην πράξη αυτά τα στάδια υπερκαλύπτονται και ανταλλάσσουν πληροφορίες μεταξύ τους. Η ανάπτυξη, λοιπόν, ενός συστήματος λογισμικού δεν είναι ένα απλό γραμμικό μοντέλο, αλλά περιλαμβάνει μία ακολουθία επαναλήψεων των δραστηριοτήτων εξέλιξης αυτού του συστήματος. Λόγω, όμως, του κόστους της κατασκευής και της έγκρισης εγγράφων, οι προαναφερόμενες επαναλήψεις στοιχίζουν αρκετά και απαιτούν σημαντική επαναληπτική εργασία. Ως εκ τούτου, μετά από ένα μικρό αριθμό τέτοιων επαναλήψεων, είναι φυσιολογικό να παγώνουν προσωρινά τμήματα της πορείας υλοποίησης του εν λόγω συστήματος, όπως είναι, για παράδειγμα, ο καθορισμός προδιαγραφών του παρόντος συστήματος, και να συνεχίζεται η εκπόνηση των ανωτέρω τμημάτων σε επόμενα στάδια της παραπάνω διαδικασίας. Πιθανά προβλήματα, λοιπόν, κατά την κατασκευή μίας εφαρμογής αφήνονται για μεταγενέστερη επίλυση ή αγνοούνται. Ωστόσο, το πρόωρο πάγωμα κομματιών ανάτασης ενός συστήματος λογισμικού μπορεί να οδηγήσει σε ένα άσχημα δομημένο σύστημα το οποίο δεν θα ανταποκρίνεται επακριβώς και κατά γράμμα στις απαιτήσεις του εκάστοτε πελάτη. Κατά τη διάρκεια του τελευταίου σταδίου της ενέργειας υλοποίησης ενός συστήματος λογισμικού σύμφωνα με το Μοντέλο του Καταρράκτη, το εν λόγω σύστημα μπαίνει σε λειτουργία. Λάθη και παραλείψεις στις αυθεντικές απαιτήσεις του συγκεκριμένου συστήματος ανακαλύπτονται. Προγραμματιστικά και σχεδιαστικά λάθη ενοποιούνται και αναγνωρίζεται η ανάγκη για νέα λειτουργικότητα. Τέλος, αξίζει να τονιστεί πως οποιοδήποτε σύστημα λογισμικού, οφείλει να εξελίσσεται, προκειμένου να παραμένει χρήσιμο. Το να γίνουν όμως οι απαραίτητες αλλαγές (συντήρηση λογισμικού) μπορεί να σημαίνει την επανάληψη κάποιων ή όλων των ενεργειών εξέλιξης ενός τέτοιου συστήματος. Επειδή, ωστόσο, το Μοντέλο του Καταρράκτη ανταποκρίνεται δύσκολα στις αλλαγές των απαιτήσεων του εκάστοτε πελάτη (αφού χωρίζει το έργο σε σχετικά δύσκαμπτα στάδια), θα πρέπει στην αρχή της μεθόδου χτισίματος ενός συστήματος λογισμικού να καθοριστούν σαφώς και να κατανοηθούν πλήρως οι απαιτήσεις αυτού του συστήματος. Έτσι, πιθανά λάθη και όποιες παραλείψεις δύναται να απαντηθούν, κατά τη διάρκεια της προηγούμενης διαδικασίας, τείνουν να ελαχιστοποιηθούν.

[69] [70] [71]

3.2 Η Ανάλυση και ο Καθορισμός των Απαιτήσεων του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Σε τούτη την ενότητα του τρέχοντος κεφαλαίου θα παρουσιαστούν οι απαιτήσεις- περιορισμοί του συστήματος λογισμικού το οποίο πραγματεύεται η παρούσα πτυχιακή διατριβή. Η εφαρμογή Διαδικτύου, λοιπόν, που αναπτύσσεται στην τρέχουσα πτυχιακή αφορά ένα λογισμικό για την παρακολούθηση ασθενών, τόσο δημόσιων όσο και ιδιωτικών, οφθαλμιατρικών κλινικών συμβεβλημένων με τα Ασφαλιστικά Ταμεία του τομέα Υγείας μίας χώρας.

Αρχικά, πρέπει να προσδιοριστούν οι χρήστες του πληροφοριακού συστήματος που μελετάται στην παρούσα εργασία, δηλαδή οι οντότητες\πρόσωπα που αλληλεπιδρούν με αυτό το σύστημα. Οι χρήστες, λοιπόν, του προαναφερόμενου συστήματος χωρίζονται σε δύο μεγάλες κατηγορίες, οι οποίες είναι οι ακόλουθες:

- I. Οι πρωτεύοντες\άμεσοι χρήστες του συστήματος λογισμικού το οποίο υλοποιείται στα πλαίσια του παρόντος έργου: Στην κατηγορία αυτοί ανήκουν οι μετέπειτα τρεις επιμέρους κατηγορίες οντοτήτων\προσώπων του εν λόγω λογισμικού:

- A. Οι διαχειριστές της διαδικτυακής εφαρμογής που αναπτύσσεται στο παρόν έγγραφο, οι οποίοι θα έχουν τη δυνατότητα να εισέρχονται σε αυτή την εφαρμογή και αποτελούν οντότητες του τομέα Υγείας μίας χώρας.

- B. Οι ιατροί, οι οποίοι θα έχουν τη δυνατότητα να εισέρχονται στη διαδικτυακή εφαρμογή που αναπτύσσεται στο παρόν έγγραφο και είναι συμβεβλημένοι με ένα ή περισσότερους Ασφαλιστικούς Φορείς του τομέα Υγείας μίας χώρας. Αυτοί οι ιατροί θα είναι σε θέση να εξετάζουν ασθενείς, που είναι ασφαλισμένοι σε έναν από τους προηγούμενους Φορείς και παρουσιάζουν οφθαλμολογικά προβλήματα. Αν, μάλιστα, τα προβλήματα αυτά χρήζουν ειδικής ιατρικής φροντίδας, τότε οι εν λόγω ιατροί θα παραπέμπουν τους συγκεκριμένους ασθενείς για περαιτέρω εξετάσεις στα συνεργαζόμενα με τους προαναφερόμενα Φορείς, είτε κρατικά είτε ιδιωτικά, νοσοκομεία-οφθαλμολογικές κλινικές.
- C. Η γραμματεία του εκάστοτε, είτε δημοσίου είτε ιδιωτικού, νοσοκομείου-οφθαλμολογικής κλινικής, η οποία θα έχει τη δυνατότητα να εισέρχεται στη διαδικτυακή εφαρμογή που αναπτύσσεται στο παρόν έγγραφο. Στο προαναφερόμενο νοσοκομείο, που είναι συμβεβλημένο με ένα ή περισσότερα Ασφαλιστικά Ταμεία του Συστήματος Υγείας μίας χώρας, θα παραπέμπεται ένας ασθενής, ο οποίος φέρει οφθαλμιατρικά προβλήματα, τα οποία χρήζουν ειδικής ιατρικής φροντίδας, και είναι ασφαλισμένος σε κάποιο από τα πιο πάνω Ταμεία, αυστηρά κατόπιν σύστασης ενός ιατρού, ο οποίος σε καμία περίπτωση δεν ανήκει στο προσωπικό του ανώτερου νοσοκομείου και συνεργάζεται άμεσα με ένα ή περισσότερα από τα εν λόγω Ταμεία.
- II. Οι δευτερευόντες/έμμεσοι χρήστες του συστήματος λογισμικού το οποίο υλοποιείται στα πλαίσια του παρόντος εγγράφου: Στην κατηγορία αυτοί ανήκουν οι τεχνικοί, που στην ουσία γίνεται λόγος για τους προγραμματιστές, του εν λόγω λογισμικού.

Στο σημείο τούτο να τονιστεί πως στο πρόγραμμα που αποτελεί το αντικείμενο μελέτης της τρέχουσας αναφοράς, οι μεν άμεσοι χρήστες του συγκεκριμένου προγράμματος θα έχουν την ευχέρεια να εισέρχονται σε αυτό, καταχωρώντας στη φόρμα εισόδου του τα προσωπικά τους στοιχεία στη μορφή όνομα χρήστη-κωδικός και εφόσον στη συνέχεια αυτά τα στοιχεία επαληθευτούν μέσω της σύνδεσης του εν λόγω προγράμματος με μία βάση δεδομένων, όπου θα προϋπάρχουν καταχωρημένες κάποιες κατάλληλες εγγραφές, οι δε έμμεσοι χρήστες-τεχνικοί του προαναφερόμενου προγράμματος θα είναι ικανοί να επεμβαίνουν άμεσα στον κώδικά της, δίχως να χρειάζεται περαιτέρω ταυτοποίησή τους.

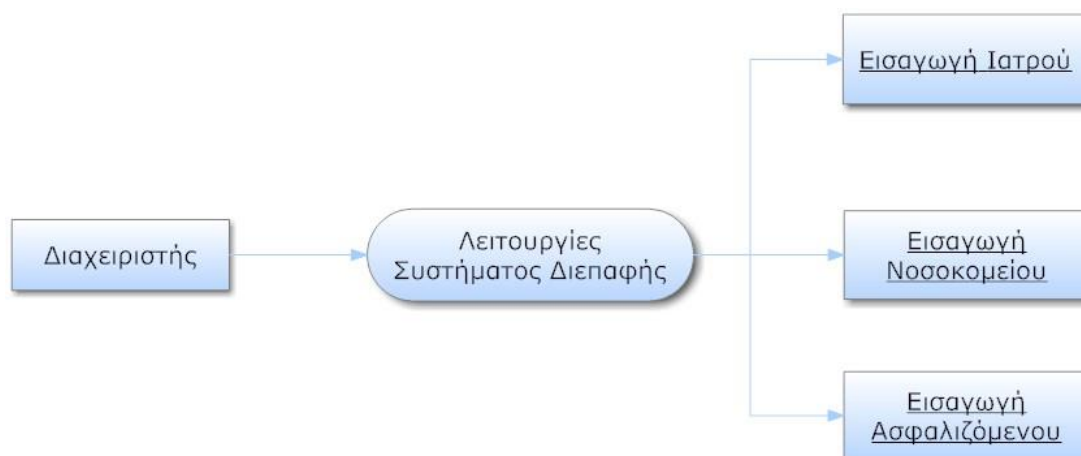
Μετά την είσοδο του στο πληροφοριακό σύστημα που λαμβάνει χώρα στην τρέχουσα πτυχιακή, κάθε ένας από τους πιο άνω, είτε άμεσους είτε έμμεσους, χρήστες θα είναι σε θέση να εκτελεί συγκεκριμένες ενέργειες, οι οποίες αναπτύσσονται πιο κάτω:

- Ο εκάστοτε διαχειριστής, που θα μπορεί να αλληλεπιδρά με το σύστημα λογισμικού το οποίο μελετάται στην παρούσα εργασία, θα δύναται να εκτελεί τις ακόλουθες ενέργειες:
 - ο Να εγγράφει έναν ιατρό στην εφαρμογή η οποία εξετάζεται στην εν λόγω εργασία.
 - ο Να εγγράφει ένα νοσοκομείο στην εφαρμογή η οποία εξετάζεται στην εν λόγω εργασία.
 - ο Να εγγράφει έναν ασθενή στην εφαρμογή η οποία εξετάζεται στην εν λόγω εργασία.
- Ο εκάστοτε ιατρός, που θα μπορεί να αλληλεπιδρά με το σύστημα λογισμικού το οποίο μελετάται στην παρούσα εργασία, θα δύναται να εκτελεί τις ακόλουθες ενέργειες:
 - ο Να συντάσσει ένα παραπεμπτικό για έναν ασθενή, κάθε φορά που κάτι τέτοιο κρίνεται αναγκαίο, προκειμένου ο συγκεκριμένος ασθενής να επισκέπτεται ένα νοσοκομείο-οφθαλμιατρική κλινική για τη διενέργεια πιο εξειδικευμένων εξετάσεων.
 - ο Να ανατρέχει στα παραπεμπτικά που αφορούν ένα συγκεκριμένο ασθενή και τα οποία έχει συντάξει είτε ο ίδιος είτε οποιοσδήποτε άλλος ιατρός συμβεβλημένος με το Ασφαλιστικό Ταμείο αυτού του ασθενούς, ώστε να έχει μια πλήρη εικόνα όλων των διαγνώσεων που έχουν γίνει στο παρελθόν και σχετίζονται με τον εν λόγω ασθενή.
 - ο Να ανατρέχει στο ιστορικό ενός ασθενούς, με σκοπό να έχει μία όσο γίνεται πιο ξεκάθαρη εικόνα της υγείας αυτού του ασθενούς.
 - ο Να ανατρέχει στο ιστορικό του ίδιου του προαναφερόμενου, για να έχει την ικανότητα να λαμβάνει πληροφορίες σχετικά με μία διάγνωση που έχει πραγματώσει αυτός ο ιατρός σε παρελθοντικό χρόνο.
- Η γραμματεία του εκάστοτε νοσοκομείου-οφθαλμολογικής κλινικής, που θα μπορεί να αλληλεπιδρά με το σύστημα λογισμικού το οποίο μελετάται στην παρούσα αναφορά, θα δύναται να εκτελεί τις ακόλουθες ενέργειες:
 - ο Να ορίζει ημερομηνία εξέτασης για όσους ασθενείς έχουν παραπεμφθεί στο προαναφερόμενο νοσοκομείο, προκειμένου να εξεταστούν λεπτομερέστερα.

- ο Να καταχωρεί τα αποτελέσματα των εξετάσεων που πραγματοποίησε ο εκάστοτε ασθενής εντός του προαναφερόμενου νοσοκομείου στον ιατρικό φάκελο αυτού του ασθενούς.
- ο Να ανατρέχει στο ιστορικό ενός ασθενούς, με σκοπό να έχει μία όσο είναι εφικτό πιο διαυγή εικόνα της υγείας αυτού του ασθενούς.
- ο Να ανατρέχει στο ιστορικό του ίδιου του προαναφερόμενου νοσοκομείου, ώστε να έχει τη δυνατότητα κάθε στιγμή να λαμβάνει πληροφορίες όσον αφορά τις εξετάσεις που έχουν λάβει χώρα σε αυτό το νοσοκομείο.
- Ο εκάστοτε τεχνικός, που προγραμματίσει το σύστημα λογισμικού το οποίο μελετάται στην παρούσα αναφορά, θα πρέπει να είναι σε θέση να εκτελεί τις ακόλουθες ενέργειες:
 - ο Να συντηρεί και να αναβαθμίζει τον κώδικα της εφαρμογής η οποία εξετάζεται στην εν λόγω εργασία.
 - ο Να ανανεώνει την άδεια του κάθε φορά τελικού χρήστη της εφαρμογής η οποία εξετάζεται στην εν λόγω εργασία.

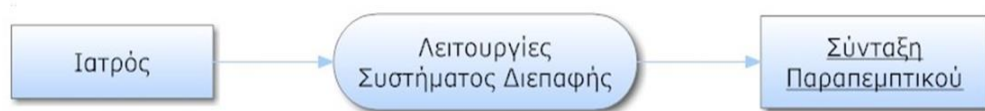
Κάπου εδώ κρίνεται επιτακτική η ανάγκη να γίνουν πιο ξεκάθαρες οι πιο βασικές απαιτήσεις του πληροφοριακού συστήματος το οποίο αναλύεται στο τρέχον έργο, όσον αφορά τους άμεσους χρήστες αυτού του συστήματος. Οι κυριότερες, απαιτήσεις, λοιπόν, αναφορικά με καθέναν από τους άμεσους χρήστες του προαναφερόμενου συστήματος καθώς και τα διαγράμματα που αποδίδουν σχηματικά τις εν λόγω απαιτήσεις, βοηθώντας με αυτόν τον τρόπο στην πληρέστερη κατανόησή τους, αποτυπώνονται παρακάτω:

- Ο εκάστοτε διαχειριστής, που θα είναι ικανός να πραγματοποιεί είσοδο στο λογισμικό το οποίο περιγράφεται στο παρόν έγγραφο, θα έχει ως πρώτη αρμοδιότητά του την καταχώρηση των απαραίτητων, αναφορικά με τη λειτουργία του προαναφερόμενου λογισμικού, εγγραφών σε μία βάση δεδομένων, η οποία θα επικοινωνεί με το εν λόγω λογισμικό. Οι εγγραφές αυτές θα αφορούν είτε ένα νέο ιατρό, είτε ένα νέο νοσοκομείο, είτε ένα νέο ασθενή, που σε κάθε περίπτωση θα είναι αναγκαίο να καταχωρηθούν με όσο γίνεται επαρκέστερη περιγραφή στην πρωτότερη βάση. Οι πιο βασικές απαιτήσεις, λοιπόν, του λογισμικού το οποίο μελετάται στην παρούσα εργασία, όσον αφορά τους χρήστες-διαχειριστές του συγκεκριμένου συστήματος, απεικονίζονται στο διάγραμμα το οποίο φαίνεται στην παρακάτω Εικόνα:



Εικόνα 33: Το διάγραμμα της ανάλυσης των πιο βασικών απαιτήσεων του λογισμικού, όσον αφορά ένα χρήστη-διαχειριστή

- Ο εκάστοτε ιατρός, που θα είναι ικανός να πραγματοποιεί είσοδο στο λογισμικό το οποίο περιγράφεται στο παρόν έγγραφο, θα έχει ως πρώτη αρμοδιότητά του τη σύνταξη παραπεμπτικού για τον εκάστοτε ασθενή που θα τον επισκέπτεται και θα χρήζει περαιτέρω εξετάσεων σε κάποια οφθαλμιατρική κλινική. Οι πιο βασικές απαιτήσεις, λοιπόν, του λογισμικού το οποίο μελετάται στην παρούσα εργασία, όσον αφορά τους χρήστες-διαχειριστές του συγκεκριμένου συστήματος, απεικονίζονται στο διάγραμμα το οποίο φαίνεται στην παρακάτω Εικόνα:



Εικόνα 34: Το διάγραμμα της ανάλυσης των πιο βασικών απαιτήσεων του λογισμικού, όσον αφορά ένα χρήστη-ιατρό

- Η εκάστοτε γραμματεία ενός νοσοκομείου, που θα είναι ικανή να πραγματοποιεί είσοδο στο λογισμικό το οποίο περιγράφεται στο παρόν έγγραφο, θα έχει ως πρώτη αρμοδιότητά της, αρχικά, τον ορισμό ημερομηνία εξέτασης για τα παραπεμπτικά ασθενών τα οποία έχουν αποσταλεί από κάποιον ιατρό στην εν λόγω οφθαλμιατρική κλινική. Στη συνέχεια, αφού εξεταστεί ο εκάστοτε ασθενής από έναν ιατρό του προαναφερόμενου νοσοκομείου (που σε κάθε περίπτωση είναι διαφορετικός από τον ιατρό που τον παρέπεμψε) και βγει το τελικό πόρισμα αυτής της εξέτασης, η γραμματεία της εν λόγω κλινικής θα οφείλει να καταχωρεί τα αποτελέσματα της συγκεκριμένης εξέτασης στον ιατρικό φάκελο του πιο πάνω ασθενούς. Οι πιο βασικές απαιτήσεις, λοιπόν, του λογισμικού το οποίο μελετάται στην παρούσα εργασία, όσον αφορά τους χρήστες-νοσοκομεία του συγκεκριμένου συστήματος, απεικονίζονται στο διάγραμμα το οποίο φαίνεται στην παρακάτω Εικόνα:



Εικόνα 35: Το διάγραμμα της ανάλυσης των πιο βασικών απαιτήσεων του λογισμικού, όσον αφορά ένα χρήστη-νοσοκομείο

Συμπερασματικά, το σύστημα που αναλύεται και μελετάται στην εν λόγω πτυχιακή διατριβή έχει ως κύρια απαίτηση της τη διαδικασία διεξαγωγής και ολοκλήρωσης οφθαλμιατρικών εξετάσεων ασθενών σύμφωνα με τα ακόλουθα καίρια βήματα:

- Ένας ασθενής, που είναι ασφαλισμένος σε έναν από τους Ασφαλιστικούς Φορείς του τομέα Υγείας μίας χώρας και έχει κάποιο πρόβλημα με τα μάτια του, επισκέπτεται έναν ιατρό συμβεβλημένο με το Ασφαλιστικό Ταμείο του παραπάνω ασθενούς, ώστε να επιτευχθεί διάγνωση και θεραπεία της πάθησης αυτού του ασθενούς.
- Ο προαναφερόμενος ιατρός, αφού εξετάσει τον πιο πάνω ασθενή και κρίνει σκόπιμο πως πρέπει να παραπεμφθεί σε κάποια, είτε δημόσια είτε ιδιωτική, οφθαλμιατρική κλινική συνεργαζόμενη με το Ταμείο του προαναφερόμενου ασθενούς, συντάσσει ένα παραπεμπτικό και το αποστέλλει στην αρμόζουσα κάθε φορά κλινική.
- Η γραμματεία της προαναφερόμενης κλινικής λαμβάνει το ανώτερο παραπεμπτικό και ορίζει ημερομηνία εξέτασης για το συγκεκριμένο ασθενή.
- Ο εν λόγω ασθενής εξετάζεται από έναν οφθαλμίατρο της ανώτερης κλινικής (ξεχωριστό σε κάθε περίπτωση από τον ιατρό που συνέταξε το παραπεμπτικό του ασθενούς) και το τελικό πόρισμα της συγκεκριμένης εξέτασης αποστέλλεται στη γραμματεία του πιο άνω νοσοκομείου, με σκοπό να καταχωρηθεί στον ιατρικό φάκελο του τρέχοντος ασθενούς.

Συνεπώς, η κεντρική ιδέα του συστήματος λογισμικού το οποίο πραγματεύεται η παρούσα αναφορά αντικατοπτρίζεται στο διάγραμμα το οποίο δείχνεται στη μετέπειτα Εικόνα:



Εικόνα 36: Το διάγραμμα της κεντρικής ιδέας του λογισμικού

[71]

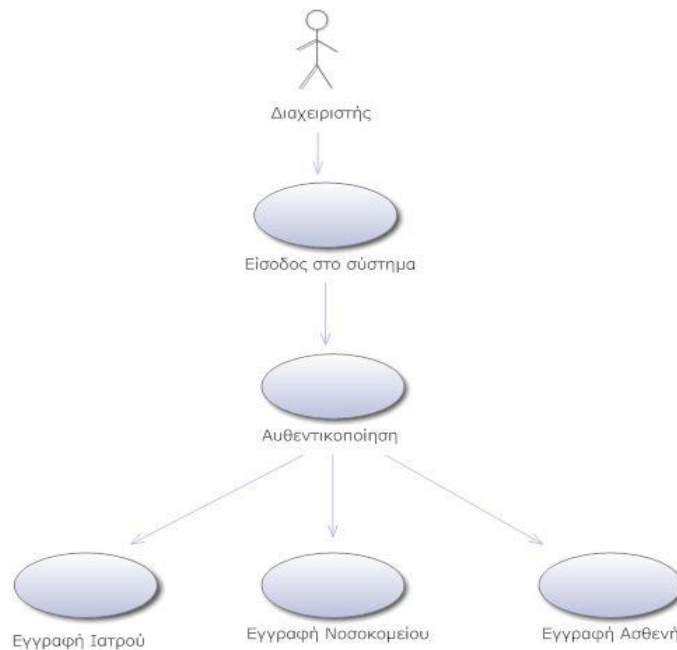
3.3 Ο Σχεδιασμός του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Στην παρούσα ενότητα του εν λόγω κεφαλαίου θα γίνει μία προσπάθεια να προσεγγιστεί με όσο είναι εφικτό με πιο αποσαφηνιστικό τρόπο ο σχεδιασμός του συστήματος λογισμικού που αποτελεί το αντικείμενο μελέτης της τρέχουσας εργασίας. Η επίτευξη ενός τέτοιου σκοπού θα γίνει με την παρουσίαση των κυριότερων UML διαγραμμάτων του προαναφερόμενου λογισμικού, η υλοποίηση των οποίων θα στηριχτεί στις απαιτήσεις του συγκεκριμένου λογισμικού, οι οποίες περιγράφηκαν διεξοδικά στην αμέσως ανώτερη ενότητα του τρέχοντος κεφαλαίου. Επίσης, να τονιστεί ότι των η κατασκευή των προαναφερόμενων UML διαγραμμάτων θα πραγματοποιηθεί με χρήση του προγράμματος “SmartDraw 2016 Enterprise Edition”.

3.3.1 Τα UML Διαγράμματα Περιπτώσεων Χρήσης του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

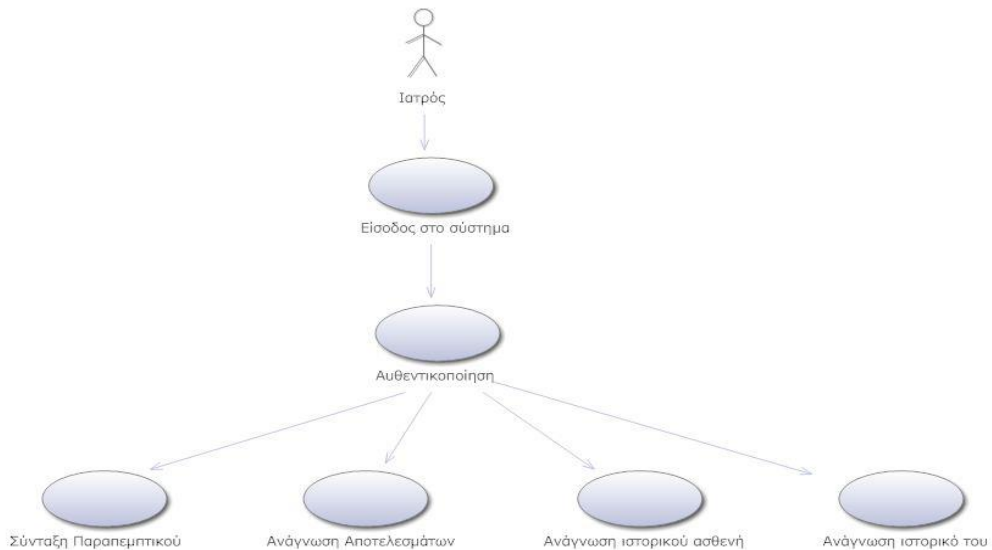
Τα UML διαγράμματα περιπτώσεων χρήσης είναι ίσως από τα πιο σημαντικά διαγράμματα της γλώσσας UML. Τα εν λόγω διαγράμματα μοντελοποιούν το πλαίσιο λειτουργίας ενός συστήματος λογισμικού και τις προδιαγραφές του. Έτσι, τα UML διαγράμματα περιπτώσεων χρήσης που καθορίζουν το μοντέλο λειτουργίας του συστήματος λογισμικού που παρουσιάζεται στην τρέχουσα εργασία είναι τα εξής:

- Το UML διάγραμμα που αναπαριστά τις περιπτώσεις χρήσης του εκάστοτε διαχειριστή, ο οποίος θα είναι σε θέση να πραγματοποιεί είσοδο στη διαδικτυακή εφαρμογή η οποία αναλύεται στο παρόν έγγραφο, και το οποίο αποδίδεται στην παρακάτω Εικόνα:



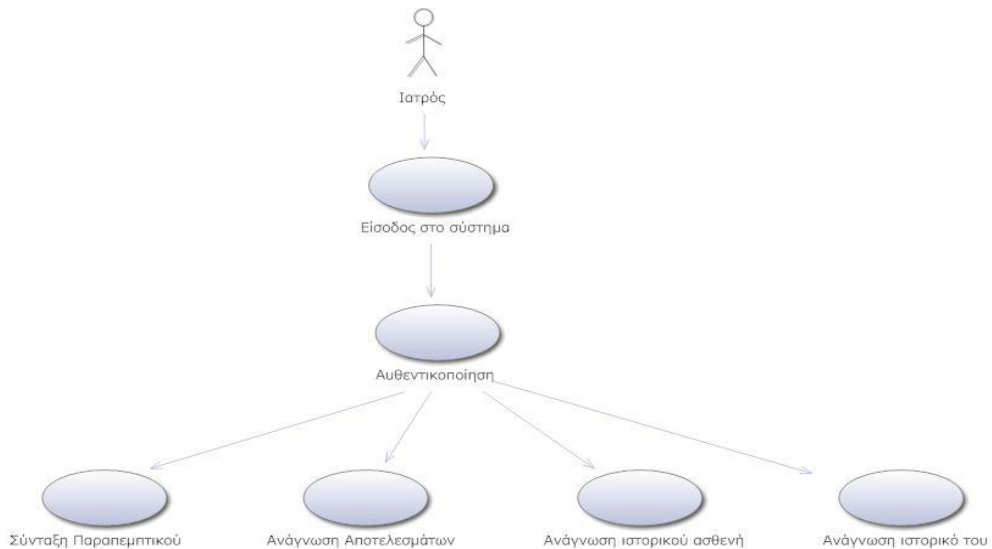
Εικόνα 37: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-διαχειριστή

- Το UML διάγραμμα που αναπαριστά τις περιπτώσεις χρήσης του εκάστοτε ιατρού, ο οποίος θα είναι σε θέση να πραγματοποιεί είσοδο στη διαδικτυακή εφαρμογή η οποία αναλύεται στο παρόν έγγραφο, και το οποίο αποδίδεται στην παρακάτω Εικόνα:



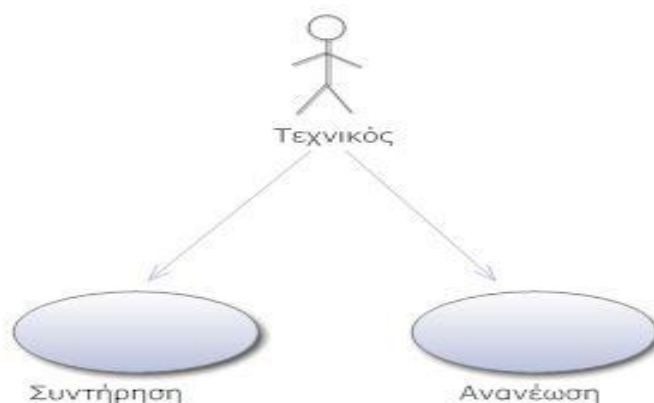
Εικόνα 38: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-ιατρό

- Το UML διάγραμμα που αναπαριστά τις περιπτώσεις χρήσης του εκάστοτε νοσοκομείου, το οποίο θα είναι σε θέση να πραγματοποιεί είσοδο στο πληροφοριακό σύστημα στη διαδικτυακή εφαρμογή η οποία αναλύεται στο παρόν έγγραφο, και το οποίο αποδίδεται στην παρακάτω Εικόνα:



Εικόνα 39: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-νοσοκομείο

- Το UML διάγραμμα που αναπαριστά τις περιπτώσεις χρήσης του εκάστοτε τεχνικού-προγραμματιστή, ο οποίος θα είναι σε θέση να επεμβαίνει στον κώδικα της διαδικτυακής εφαρμογής η οποία αναλύεται στο παρόν έγγραφο, και το οποίο αποδίδεται στην παρακάτω Εικόνα:

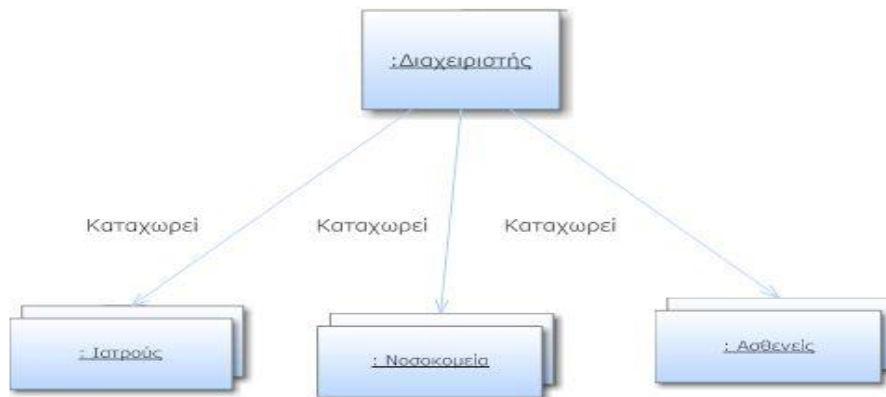


Εικόνα 40: Το UML διάγραμμα περιπτώσεων χρήσης του λογισμικού, όσον αφορά ένα χρήστη-τεχνικό
[71] [72] [73] [75] [76]

3.3.2 Τα UML Διαγράμματα Αντικειμένων του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

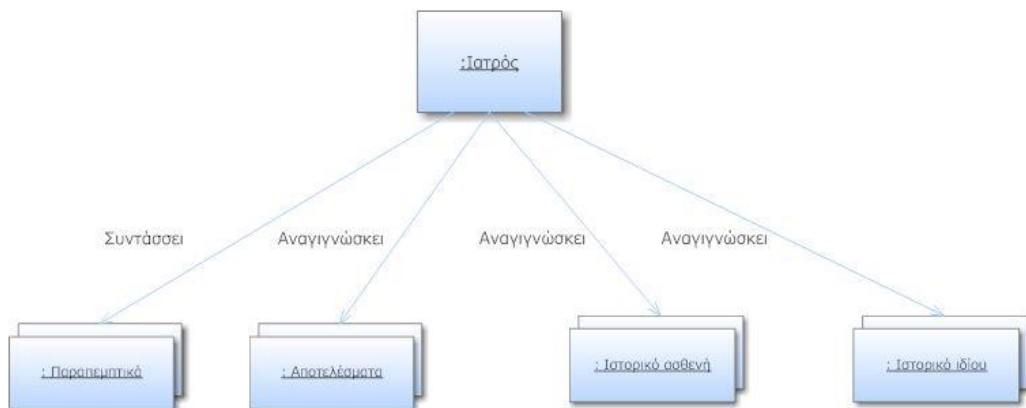
Τα διαγράμματα αντικειμένων της γλώσσας UML εκφράζουν την στατική δομή ενός πληροφοριακού συστήματος με όρους αντικειμένων και τις μεταξύ τους σχέσεις. Άρα, τα UML διαγράμματα αντικειμένων του πληροφοριακού συστήματος που υλοποιείται στην τρέχουσα αναφορά είναι τα κατωτέρω:

- Το UML διάγραμμα αντικειμένων που αφορά τον εκάστοτε διαχειριστή, ο οποίος θα δύναται να εισέρχεται στο λογισμικό που μελετάται στο τρέχον έργο, και το οποίο αναπαρίσταται στην ακόλουθη Εικόνα:



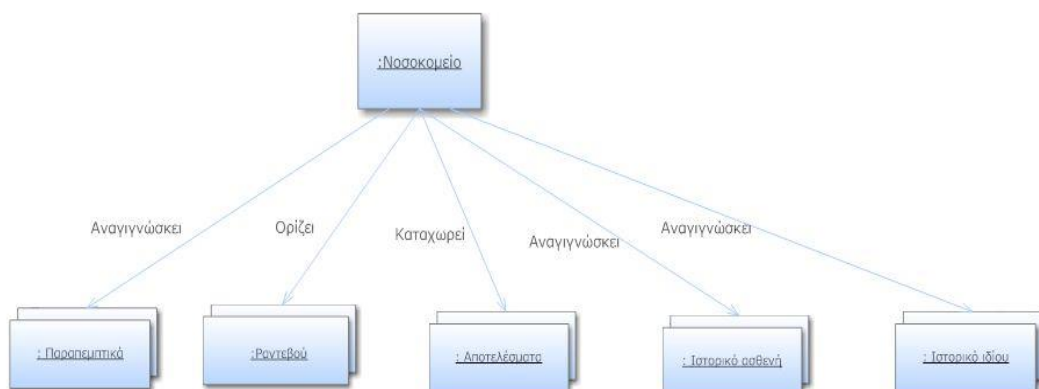
Εικόνα 41: Το UML διάγραμμα αντικειμένων του λογισμικού, όσον αφορά ένα χρήστη-διαχειριστή

- Το UML διάγραμμα αντικειμένων που αφορά τον εκάστοτε ιατρό, ο οποίος θα δύναται να εισέρχεται στο λογισμικό που μελετάται στο τρέχον έργο, και το οποίο αναπαρίσταται στην ακόλουθη Εικόνα:



Εικόνα 42: Το UML διάγραμμα αντικειμένων του λογισμικού, όσον αφορά ένα χρήστη-ιατρό

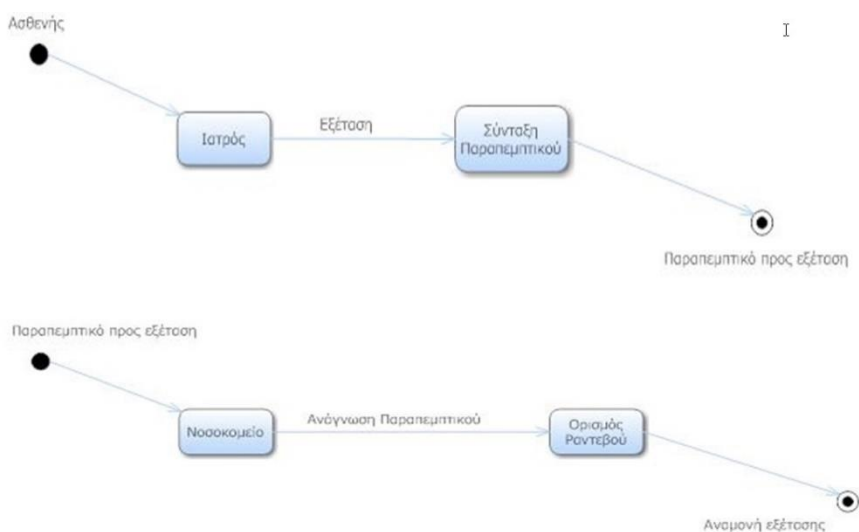
- Το UML διάγραμμα αντικειμένων που αφορά το εκάστοτε νοσοκομείο, το οποίο θα δύναται να εισέρχεται στο λογισμικό που μελετάται στο τρέχον έργο, και το οποίο αναπαρίσταται στην ακόλουθη Εικόνα:



Εικόνα 43: Το UML διάγραμμα αντικειμένων του λογισμικού, όσον αφορά ένα χρήστη-νοσοκομείο [72] [74] [75] [76]

3.3.3 Το UML Διάγραμμα Καταστάσεων του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Ένα διάγραμμα καταστάσεων της γλώσσας UML εξετάζει τα αντικείμενα ενός συστήματος λογισμικού και περιγράφει τη ροή του ελέγχου από μία κατάσταση σε μία άλλη. Επομένως, το UML διάγραμμα καταστάσεων του συστήματος λογισμικού που πραγματεύεται η τρέχουσα πτυχιακή παρουσιάζεται στη μετέπειτα Εικόνα:

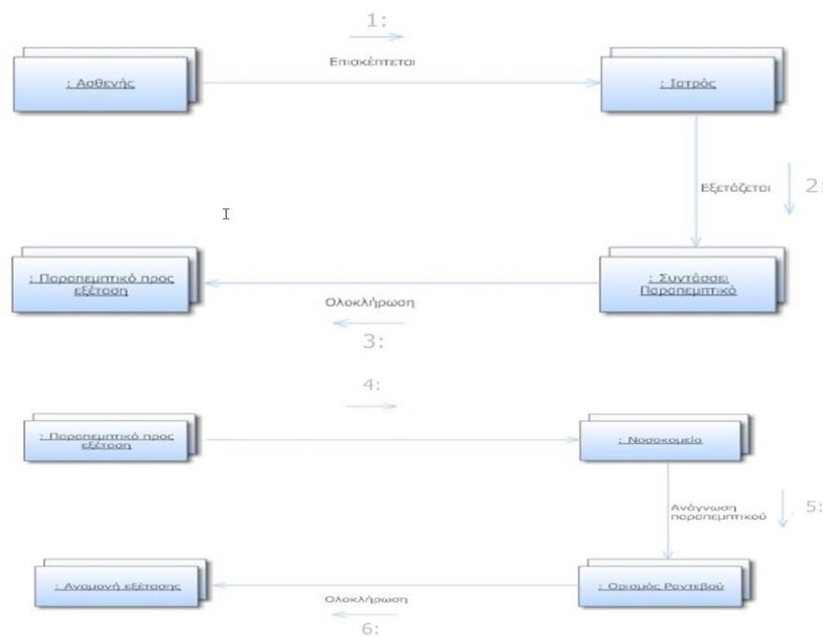


Εικόνα 44: Το UML διάγραμμα καταστάσεων του λογισμικού

[72] [74] [75] [76]

3.3.4 Το UML Διάγραμμα Συνεργασίας του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Ένα διάγραμμα συνεργασίας της γλώσσας UML αναπαριστά την αλληλεπίδραση των αντικειμένων ενός πληροφοριακού συστήματος, μέσω της ανταλλαγής μηνυμάτων, εξάροντας τη χρονική αλληλουχία αυτών των μηνυμάτων. Συνεπώς, το UML διάγραμμα συνεργασίας του πληροφοριακού συστήματος που πραγματεύεται η τρέχουσα πτυχιακή περιγράφεται στην πιο κάτω Εικόνα:



Εικόνα 45: Το UML διάγραμμα συνεργασίας του λογισμικού

[72] [74] [75] [76]

3.4 Ο Τρόπος Λειτουργίας του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Στην ενότητα τούτη θα παρουσιαστεί βήμα προς βήμα ο τρόπος λειτουργίας της εφαρμογής Διαδικτύου την οποία αναλύει η τρέχουσα διπλωματική εργασία, παραθέτοντας αναλυτικά, από την αρχή ως το τέλος, τις δυνατότητες τις οποίες αυτό η εφαρμογή δύναται να προσφέρει. Η παρουσίαση κάθε υποομάδας του λογισμικού το οποίο συνιστά το επίκεντρο αυτής της εργασίας, θα συνοδεύεται από κατάλληλη τεκμηρίωση και αντίστοιχα στιγμιότυπα οθόνης τα οποία θα προκύπτουν από την εκτέλεση του ανάλογου κώδικα καθεμιάς από τις προαναφερόμενες υποομάδες. Έτσι, θα επιτευχθεί η πλήρη αποσαφήνιση κάθε υποενότητας του συστήματος λογισμικού το οποίο μελετάται στο τρέχον έργο, με απώτερο σκοπό και στόχο την πλήρη κατανόηση της λειτουργίας του εν λόγω λογισμικού, ώστε να εξασφαλιστεί η απρόσκοπτη χρήση του και να εντοπιστούν τυχόν ατέλειες του, οι οποίες είναι δυνατό να διορθωθούν σε μετέπειτα αναβαθμίσεις του, διευκολύνοντας με αυτό κατά αυτόν τον τρόπο τη μελλοντική συντήρησή του.

3.4.1 Η Είσοδος ενός Διαχειριστή στο Σύστημα Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Το αρχικό παράθυρο της εφαρμογής η οποία αναπτύσσεται στα πλαίσια τούτης της πτυχιακής διατριβής φαίνεται στην παρακάτω Εικόνα:

Εθνικό Σύστημα Υγείας (ΕΣΥ) - Οφθαλμιατρικές Εξετάσεις

Ιατρός	Νοσοκομείο	Διαχειριστής
<p>ΟΝΟΜΑ ΧΡΗΣΤΗ</p> <input style="width: 80%;" type="text"/> <p>ΚΩΔΙΚΟΣ</p> <input style="width: 80%;" type="text"/> <p style="text-align: center; margin-top: 10px;">ΕΙΣΟΔΟΣ</p>	<p>ΟΝΟΜΑ ΧΡΗΣΤΗ</p> <input style="width: 80%;" type="text"/> <p>ΚΩΔΙΚΟΣ</p> <input style="width: 80%;" type="text"/> <p style="text-align: center; margin-top: 10px;">ΕΙΣΟΔΟΣ</p>	<p>ΟΝΟΜΑ ΧΡΗΣΤΗ</p> <input style="width: 80%;" type="text"/> <p>ΚΩΔΙΚΟΣ</p> <input style="width: 80%;" type="text"/> <p style="text-align: center; margin-top: 10px;">ΕΙΣΟΔΟΣ</p>

Εικόνα 46: Το αρχικό παράθυρο του λογισμικού

Σύμφωνα με την πιο πάνω Εικόνα το πληροφοριακό σύστημα το οποίο υλοποιείται στα πλαίσια της παρούσας αναφοράς δίνει τη δυνατότητα σε τρεις κατηγορίες κύριων χρηστών να πραγματοποιούν είσοδο σε αυτό το σύστημα, οι οποίες είναι οι ακόλουθες:

- Ο εκάστοτε διαχειριστής του λογισμικού το οποίο πραγματεύεται το τρέχον έργο.
- Ο εκάστοτε ιατρός, που δύναται να αλληλεπιδρά με το λογισμικό το οποίο πραγματεύεται το τρέχον έργο.
- Η γραμματεία του εκάστοτε νοσοκομείου, η οποία δύναται να αλληλεπιδρά με το λογισμικό το οποίο πραγματεύεται το τρέχον έργο.

Η είσοδος, τώρα, καθενός από τους παραπάνω πρωτεύοντες χρήστες στην εφαρμογή η οποία παρουσιάζεται στην παρούσα αναφορά είναι διαβαθμισμένη, δηλαδή απαιτεί την καταχώρηση των προσωπικών δεδομένων των πιο άνω χρηστών στη φόρμα εισόδου της συγκεκριμένης εφαρμογής κατά το σχήμα όνομα χρήστη-κωδικός. Στη συνέχεια, τα στοιχεία αυτά θα επαληθευτούν με βάση τις υφιστάμενες σε κάθε περίπτωση αρμόζουσες εγγραφές, οι οποίες θα υπάρχουν ήδη στην κατάλληλη σε κάθε περίπτωση βάση δεδομένων, με την οποία θα επικοινωνεί η συγκεκριμένη εφαρμογή, και έτσι θα πραγματοποιείται επιτυχής ή ανεπιτυχής είσοδος των πιο πάνω χρηστών στην προαναφερόμενη εφαρμογή.

Βέβαια, η πρώτη-πρώτη είσοδος στο πρόγραμμα το οποίο αποτελεί το αντικείμενο μελέτης του παρόντος εγγράφου θα πρέπει να γίνει αυστηρά από τον εκάστοτε διαχειριστή του συγκεκριμένου προγράμματος, προκειμένου αυτός ο διαχειριστής να καταχωρήσει τα απαραίτητα κάθε φορά δεδομένα προς επεξεργασία. Συνεπώς, ο εν λόγω διαχειριστής, προκειμένου να εισέρθει για πρώτη φορά στο τρέχον λογισμικό, θα καταχωρήσει το ορθό όνομα χρήστη και το σωστό κωδικό στα αντίστοιχα πεδία της φόρμας εισόδου διαχειριστή, η οποία περιέχεται στο αρχικό παράθυρο του παρόντος λογισμικού και κατόπιν θα πατήσει το κουμπί «Είσοδος», το οποίο βρίσκεται στο κάτω μέρος της πρωτύτερης φόρμας.

Υστερα, θα ακολουθήσει ταυτοποίηση του εκάστοτε διαχειριστή, ο οποίος επιχειρεί να εισέρθει στην εφαρμογή η οποία αναπτύσσεται στην τρέχουσα πτυχιακή. Έτσι, αν δεν επαληθευτούν τα στοιχεία αυτού του διαχειριστή, τότε θα αναδυθεί το παράθυρο προειδοποίησης το οποίο απεικονίζεται στην κατώτερη Εικόνα:

Το όνομα χρήστη ή ο κωδικός που πληκτρολογήσατε δε βρέθηκε!!!

Δοκιμάστε ξανά

Εικόνα 47: Το παράθυρο προειδοποίησης της ανεπιτυχούς εισόδου στο λογισμικό ενός διαχειριστή

Το παράθυρο που αποδίδεται στην πιο πάνω Εικόνα προειδοποιεί τον εκάστοτε διαχειριστή, ο οποίος προσπαθεί να πραγματοποιήσει είσοδο στην εφαρμογή η οποία μελετάται στο παρόν έργο, ότι εισήγαγε στα πεδία της φόρμας εισόδου διαχειριστή, η οποία υπάρχει στο αρχικό παράθυρο της εν λόγω εφαρμογής, είτε λαθεμένο όνομα χρήστη, είτε λαθεμένο κωδικό, είτε λαθεμένα και τα δύο αυτά στοιχεία. Επίσης, δίνει τη δυνατότητα στον πιο άνω διαχειριστή να καταχωρήσει τα προσωπικά του δεδομένα εισόδου στην προαναφερόμενη φόρμα με ορθό τρόπο, κάνοντας κλικ στο κουμπί «Δοκιμάστε ξανά» του ανώτερου παραθύρου.

Στην περίπτωση, όμως, που τα στοιχεία του εκάστοτε διαχειριστή, ο οποίος επιδιώκει να εισέρθει στη διαδικτυακή εφαρμογή η οποία ορίζεται στο παρόν έγγραφο, επαληθευτούν, τότε θα προκύψει το παράθυρο το οποίο φαίνεται στην ύστερη Εικόνα:



Εικόνα 48: Το παράθυρο της διαχείρισης πληροφοριών ενός διαχειριστή

Το παράθυρο που δείχνεται στην παραπάνω Εικόνα δίνει την ικανότητα στον εκάστοτε διαχειριστή, ο οποίος αποτελεί τον τρέχον του διαδικτυακού συστήματος το οποίο υλοποιείται στην τρέχουσα διατριβή, να διαχειρίζεται με κατάλληλο τρόπο τις πληροφορίες και να είναι σε θέση να εκτελεί μία από τις ακόλουθες ενέργειες:

- Να εγγράφει έναν ιατρό στο σύστημα το οποίο εξετάζεται στην παρούσα διατριβή, πατώντας το κουμπί «Εγγραφή Ιατρού».
- Να εγγράφει ένα νοσοκομείο στο σύστημα το οποίο εξετάζεται στην παρούσα διατριβή, πατώντας το κουμπί «Εγγραφή Νοσοκομείου».
- Να εγγράφει έναν ασθενή στο σύστημα το οποίο εξετάζεται στην παρούσα διατριβή, πατώντας το κουμπί «Εγγραφή Ασθενούς».
- Να αποσυνδέεται από το σύστημα το οποίο εξετάζεται στην παρούσα διατριβή, πατώντας το κουμπί «ΑΠΟΣΥΝΔΕΣΗ».

Συνεπώς, στην περίπτωση που ο εκάστοτε διαχειριστής, ο οποίος συνιστά τον παρών χρήστη του λογισμικού Διαδικτύου το οποίο παρουσιάζεται στην τρέχουσα αναφορά, θα είναι αναγκαίο να εγγράψει έναν ιατρό στο εν λόγω λογισμικό, τότε θα προβάλλει το παράθυρο το οποίο αναπαρίσταται στην κατοπινή Εικόνα:

Φόρμα Εγγραφής Ιατρού

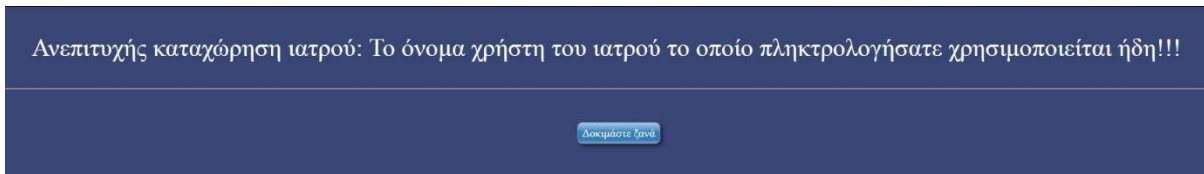
Όνομα Χρήστη	<input type="text"/>	[Πληκτρολογήστε το όνομα χρήστη του ιατρού, το οποίο πρέπει να περιέχει από 5 έως 16 μη λευκούς χαρακτήρες...]
Κωδικός	<input type="text"/>	[Πληκτρολογήστε τον κωδικό του ιατρού, ο οποίος πρέπει να περιέχει από 6 έως 14 μη λευκούς χαρακτήρες...]
Επαλήθευση Κωδικού	<input type="text"/>	[Πληκτρολογήστε ξανά τον κωδικό του ιατρού...]
Όνομα	<input type="text"/>	[Πληκτρολογήστε το μικρό όνομα του ιατρού, το οποίο πρέπει να περιέχει από 2 έως 25 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών ονομάτων!!!)]
Επώνυμο	<input type="text"/>	[Πληκτρολογήστε το επίθετο του ιατρού, το οποίο πρέπει να περιέχει από 2 έως 35 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών επωνύμων!!!)]
Φύλο	<input type="radio"/> <input type="radio"/>	[Επιλέξτε το φύλο του ιατρού: Άρρεν ή Θήλυ...]
Ειδικότητα	<input type="text"/>	[Πληκτρολογήστε την ειδικότητα του ιατρού, η οποία πρέπει να περιέχει από 2 έως 100 αλφαβητικούς χαρακτήρες ή λευκούς χαρακτήρες (εκτός από τον πρώτο χαρακτήρα) ή ένα ή περισσότερα από τα σύμβολα "-", " ", "(", ")", " " ...]
Κινητό	<input type="text"/>	[Πληκτρολογήστε το κινητό τηλέφωνο του ιατρού, το οποίο πρέπει να περιέχει ακριβώς 10 ψηφία...]

Εικόνα 49: Το παράθυρο της φόρμας εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή

Το παράθυρο που αποδίδεται στην πιο άνω Εικόνα καθιστά ικανό τον εκάστοτε διαχειριστή, ο οποίος είναι ο παρών χρήστης του διαδικτυακού συστήματος λογισμικού το οποίο λαμβάνει χώρα στην παρούσα διπλωματική, να εγγράψει έναν ιατρό στο εν λόγω λογισμικό, συμπληρώνοντας τα πεδία της φόρμας εγγραφής ιατρού η οποία φαίνεται στην πιο πάνω Εικόνα και υποβάλλοντας κατόπιν τη συγκεκριμένη φόρμα στο διακομιστή Ιστού του συγκεκριμένου συστήματος, με το πάτημα του κουμπιού «Υποβολή». Στο σημείο αυτό να τονιστεί ότι ο προαναφερόμενος διαχειριστής, κατά τη διαδικασία συμπλήρωσης των πεδίων της προηγούμενης φόρμας, θα οφείλει να τηρεί τους κανόνες οι οποίοι αναφέρονται στα δεξιά καθενός από τα αυτά πεδία. Επίσης, δεν θα πρέπει σε καμιά περίπτωση να καταχωρήσει στο πεδίο «Όνομα Χρήστη» ή στο πεδίο «Κινητό» της προαναφερόμενης φόρμας τιμές οι οποίες υπάρχουν ήδη στα αντίστοιχα πεδία της βάσης δεδομένων με την οποία επικοινωνεί το εν λόγω λογισμικό, διότι το όνομα χρήστη και το κινητό τηλέφωνο ενός ιατρού ταυτοποιεί μοναδικά κάθε ιατρό, ο οποίος πρόκειται να εγγραφεί στο παραπάνω σύστημα λογισμικού.

Έτσι, αν ο εκάστοτε διαχειριστής, ο οποίος χρησιμοποιεί την εφαρμογή την οποία πραγματεύεται το τρέχον έγγραφο, κατά την ενέργεια συμπλήρωσης της φόρμας εγγραφής ιατρού της συγκεκριμένης εφαρμογής, καταχωρήσει στο πεδίο «Όνομα Χρήστη» της προαναφερόμενης φόρμας ένα όνομα χρήστη ιατρού το οποίο υπάρχει ήδη στο ανάλογο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί

η εν λόγω εφαρμογή, τότε, πιέζοντας ο παρών διαχειριστής το κουμπί «Υποβολή» της προηγούμενης φόρμας θα εμφανιστεί το παράθυρο προειδοποίησης το οποίο φαίνεται στην παρακάτω Εικόνα:



Εικόνα 50: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου ονόματος χρήστη του ιατρού

Το παράθυρο που φαίνεται στην πιο πάνω Εικόνα ειδοποιεί τον εκάστοτε διαχειριστή, ο οποίος είναι ο τρέχων χρήστης του συστήματος το οποίο προδιαγράφεται στην παρούσα εργασία, ότι κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής ιατρού του προαναφερόμενου συστήματος εισήγαγε στο πεδίο «Όνομα Χρήστη» της εν λόγω φόρμας ένα όνομα χρήστη ιατρού το οποίο προϋπάρχει στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία διασυνδέεται το συγκεκριμένο λογισμικό. Ταυτόχρονα, δίνει την ευκαιρία στον τρέχων διαχειριστή να καταχωρήσει στο πεδίο «Όνομα Χρήστη» της παραπάνω φόρμας ένα όνομα χρήστη ιατρού το οποίο δεν απαντάται στην προηγούμενη βάση, πατώντας το κουμπί «Δοκιμάστε ξανά».

Αμα, τώρα, ο εκάστοτε διαχειριστής, ο οποίος είναι ο παρών χρήστης του λογισμικού το οποίο ορίζεται στην παρούσα αναφορά, κατά την πράξη γεμίματος της φόρμας εγγραφής ιατρού του εν λόγω λογισμικού, γράψει στο πεδίο «Κινητό» της προαναφερόμενης φόρμας ένα κινητό τηλέφωνο ιατρού το οποίο είναι καταχωρημένο ήδη στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία συνδέεται το συγκεκριμένο λογισμικό, τότε, «κλικάροντας» ο τρέχων διαχειριστής το κουμπί «Υποβολή» της προγενέστερης φόρμας, θα αναδυθεί το παράθυρο προειδοποίησης το οποίο αναπαρίσταται στη μεταγενέστερη Εικόνα:



Εικόνα 51: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου κινητού τηλεφώνου του ιατρού

Το παράθυρο που δείχνεται στην πιο άνω Εικόνα ενημερώνει τον εκάστοτε διαχειριστή, ο οποίος είναι ο παρών λειτουργός του συστήματος το οποίο προδιαγράφεται στην τρέχουσα μελέτη, πως κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής ιατρού του προαναφερόμενου συστήματος καταχώρησε στο πεδίο «Κινητό» της εν λόγω φόρμας ένα κινητό τηλέφωνο ιατρού το οποίο είναι εγγεγραμμένο ήδη στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία έχει επικοινωνία το συγκεκριμένο λογισμικό. Παράλληλα, δίνει τη δυνατότητα στον προηγούμενο διαχειριστή να καταχωρήσει στο πεδίο «Κινητό» της προαναφερόμενης φόρμας ένα κινητό τηλέφωνο ιατρού το οποίο δεν υπάρχει στην πρωτότερη βάση, κάνοντας κλικ στο κουμπί «Δοκιμάστε ξανά» του ανώτερου παραθύρου.

Επιπρόσθετα, εφόσον, ο εκάστοτε διαχειριστής, ο οποίος μπαίνει στο σύστημα λογισμικού το οποίο αναπτύσσεται στην παρούσα εργασία, κατά την πράξη συμπλήρωσης της φόρμας εγγραφής ιατρού του προαναφερόμενου συστήματος, εισάγει τόσο στο πεδίο «Όνομα Χρήστη» όσο και στο πεδίο «Κινητό» της προαναφερόμενης φόρμας τιμές οι οποίες έχουν ήδη εισαχθεί στα ανάλογα πεδία της βάσης δεδομένων με την οποία επικοινωνεί το εν λόγω λογισμικό, τότε, πατώντας ο συγκεκριμένος διαχειριστής το κουμπί «Υποβολή» της πρωτότερης φόρμας, θα φανεί το παράθυρο προειδοποίησης το οποίο απεικονίζεται στην επόμενη Εικόνα:

Ανεπιτυχής καταχώρηση ιατρού: Τόσο το όνομα χρήστη του ιατρού όσο και το κινητό του τηλέφωνο τα οποία πληκτρολογήσατε χρησιμοποιούνται ήδη!!!

Δοκιμάστε ξανά

Εικόνα 52: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενων ονόματος χρήστη και κινητού τηλεφώνου του ιατρού

Το παράθυρο που δείχνεται στην παραπάνω Εικόνα προειδοποιεί τον εκάστοτε διαχειριστή, ο οποίος χειρίζεται το πληροφοριακό σύστημα Διαδικτύου το οποίο αναλύεται στην παρούσα εργασία, πως, κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής ιατρού αυτού του συστήματος, καταχώρησε τόσο στο πεδίο «Όνομα Χρήστη» όσο και στο πεδίο «Κινητό» της προαναφερόμενης φόρμας τιμές οι οποίες υφίστανται από πριν στα αντίστοιχα πεδία της βάσης δεδομένων με την οποία είναι συνδεδεμένο το εν λόγω λογισμικό. Την ίδια στιγμή, δίνει τη δυνατότητα στον προαναφερόμενο διαχειριστή να καταχωρήσει στα προηγούμενα πεδία της προαναφερόμενης φόρμας τιμές που δεν είναι εγγεγραμμένες στην προγενέστερη βάση, πιέζοντας το κουμπί «Δοκιμάστε ξανά» του πιο άνω παραθύρου..

Στην περίπτωση, βέβαια, που ο εκάστοτε διαχειριστής, ο οποίος πραγματώνει είσοδο στην εφαρμογή η οποία λαμβάνει χώρα στο τρέχον έργο, συμπληρώσει με σωστό τρόπο όλα τα πεδία της φόρμας εγγραφής ιατρού της εν λόγω εφαρμογής, τότε, πατώντας ο συγκεκριμένος διαχειριστής το κουμπί «Υποβολή» της προαναφερόμενης φόρμας, θα προκύψει το παράθυρο επιβεβαίωσης το οποίο αντικατοπτρίζεται στην ακόλουθη Εικόνα:



Εικόνα 53: Το παράθυρο επιβεβαίωσης της επιτυχούς εγγραφής ενός ιατρού στο λογισμικό από ένα διαχειριστή

Το παράθυρο που απεικονίζεται στην ανώτερη Εικόνα πληροφορεί τον εκάστοτε διαχειριστή, ο οποίος συνιστά τον τρέχων χρήστη του λογισμικού το οποίο οριοθετείται σε τούτη τη διπλωματική, πως συμπλήρωσε έγκυρα τα πεδία της φόρμας εγγραφής ιατρού αυτού του λογισμικού και συνεπώς καταχώρησε με επιτυχία ένα νέο ιατρό στη βάση δεδομένων η οποία επικοινωνεί με το εν λόγω σύστημα. Ακόμα, παρέχει στον παραπάνω διαχειριστή την ευχέρεια να εγγράψει εκ νέου είτε έναν ιατρό, είτε ένα νοσοκομείο, είτε έναν ασθενή στο προαναφερόμενο λογισμικό, αφότου πατώντας ο εν λόγω διαχειριστής το κουμπί «Εντάξει» του παραπάνω παραθύρου, του εμφανίζεται το παράθυρο διαχείρισης πληροφοριών του παρόντος λογισμικού.

Στην περίπτωση, τώρα, που ο εκάστοτε διαχειριστής, ο οποίος συνιστά τον τρέχων χρήστη του συστήματος Διαδικτύου το οποίο υλοποιείται στην παρούσα αναφορά, θα πρέπει να εγγράψει ένα νοσοκομείο στο εν λόγω σύστημα, τότε θα αναδυθεί το παράθυρο το οποίο φαίνεται στην επόμενη Εικόνα:

Φόρμα Εγγραφής Νοσοκομείου

Όνομα χρήστη		[Πληκτρολογήστε το όνομα χρήστη του νοσοκομείου, το οποίο πρέπει να περιέχει από 5 έως 16 μη λευκούς χαρακτήρες...]
Κωδικός		[Πληκτρολογήστε τον κωδικό του νοσοκομείου, ο οποίος πρέπει να περιέχει από 6 έως 14 μη λευκούς χαρακτήρες...]
Επαλήθευση Κωδικού		[Πληκτρολογήστε ξανά τον κωδικό του νοσοκομείου...]
Όνομασία		[Πληκτρολογήστε το όνομα του νοσοκομείου, το οποίο πρέπει να περιέχει από 2 έως 100 αλφαβητικούς ή αριθμητικούς ή λευκούς χαρακτήρες (εκτός από τον πρώτο χαρακτήρα) ή ένα ή περισσότερα από τα σύμβολα "-", " ", ".", "(", ")", " " ή διπλά εισαγωγικά...]
Διεύθυνση		[Πληκτρολογήστε την πλήρη διεύθυνση του νοσοκομείου, η οποία πρέπει να περιέχει από 10 έως και 120 αλφαβητικούς ή αριθμητικούς ή λευκούς χαρακτήρες (εκτός από τον 1ο χαρακτήρα) ή ένα ή περισσότερα από τα σύμβολα "-", " ", ".", "(", ")", " " ...]

Εικόνα 54: Το παράθυρο της φόρμας εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή

Το παράθυρο που δείχνεται στην πιο άνω Εικόνα δίνει την ευχέρεια στον εκάστοτε διαχειριστή, ο οποίος είναι ο τρέχων χρήστης του πληροφοριακού συστήματος το οποίο καθορίζεται στην παρούσα εργασία, να εγγράψει έναν ιατρό στο εν λόγω λογισμικό, συμπληρώνοντας τα πεδία της φόρμας εγγραφής νοσοκομείου η οποία απεικονίζεται στην πιο πάνω Εικόνα και υποβάλλοντας ύστερα την εν λόγω φόρμα στο διακομιστή Ιστού του ανώτερου συστήματος, πατώντας το κουμπί «Υποβολή». Εδώ, να παρατηρηθεί ότι ο προαναφερόμενος διαχειριστής, κατά την ενέργεια συμπλήρωσης των πεδίων της πρωτύτης φόρμας, θα χρειαστεί να τηρεί τις συστάσεις οι οποίες σημειώνονται στα δεξιά καθενός από τα αυτά πεδία. Επιπρόσθετα, δεν θα πρέπει σε καμιά περίπτωση να καταχωρήσει στο πεδίο «Όνομα Χρήστη» ή στο πεδίο «Όνομασία» της προαναφερόμενης φόρμας τιμές οι οποίες υφίστανται ήδη στα αντίστοιχα πεδία της βάσης δεδομένων με την οποία διασυνδέεται το εν λόγω σύστημα, διότι το όνομα χρήστη και το κινητό τηλέφωνο ενός ιατρού ταυτοποιεί ξεχωριστά κάθε ιατρό, ο οποίος πρόκειται να εγγραφεί στο παραπάνω σύστημα.

Συνεπώς, αν ο εκάστοτε διαχειριστής, ο οποίος χειρίζεται την εφαρμογή η οποία μελετάται στο παρόν έργο, κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής νοσοκομείου της εν λόγω εφαρμογής, καταχωρήσει στο πεδίο «Όνομα Χρήστη» της προαναφερόμενης φόρμας ένα όνομα χρήστη νοσοκομείου το οποίο είναι ήδη γραμμένο στο ανάλογο πεδίο της βάσης δεδομένων με την οποία συνδέεται η τρέχουσα εφαρμογή, τότε, «κλικάροντας» ο παραπάνω διαχειριστής το κουμπί «Υποβολή» της προγενέστερης φόρμας θα αναδυθεί το παράθυρο προειδοποίησης το οποίο δείχνεται στην κατοπινή Εικόνα:

Ανεπιτυχής καταχώρηση νοσοκομείου: Το όνομα χρήστη του νοσοκομείου που πληκτρολογήσατε χρησιμοποιείται ήδη!!!

Δοκιμάστε ξανά

Εικόνα 55: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου ονόματος χρήστη του νοσοκομείου

Το παράθυρο που αντικατοπτρίζεται στην πιο πάνω Εικόνα γνωστοποιεί στον εκάστοτε διαχειριστή, ο οποίος χρησιμοποιεί το σύστημα το οποίο μελετάται στην τρέχουσα αναφορά, ότι κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής νοσοκομείου του προαναφερόμενου συστήματος εισήγαγε στο πεδίο «Όνομα Χρήστη» της εν λόγω φόρμας ένα όνομα χρήστη ιατρού το οποίο είναι ήδη καταχωρημένο στο ανάλογο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί το συγκεκριμένο σύστημα. Παράλληλα, δίνει την ικανότητα στον τρέχων διαχειριστή να καταχωρήσει στο πεδίο «Όνομα Χρήστη» της ανώτερης φόρμας ένα όνομα χρήστη νοσοκομείου το οποίο δεν συναντάται στην προηγούμενη βάση, πατώντας το κουμπί «Δοκιμάστε ξανά» του παραπάνω παραθύρου.

Άμα, τώρα, ο εκάστοτε διαχειριστής, ο οποίος μπαίνει στο λογισμικό το οποίο λαμβάνει χώρα στην παρούσα αναφορά, κατά την ενέργεια συμπλήρωσης της φόρμας εγγραφής νοσοκομείου αυτού του λογισμικού, γράψει στο πεδίο «Όνομασία» της προαναφερόμενης φόρμας ένα όνομα νοσοκομείου το οποίο είναι εγγεγραμμένο ήδη στο ανάλογο πεδίο της βάσης δεδομένων με την οποία είναι συνδεδεμένο το συγκεκριμένο λογισμικό, τότε, πιέζοντας ο παρών διαχειριστής το κουμπί «Υποβολή» της πρωτύτης φόρμας, θα προβάλει το παράθυρο προειδοποίησης το οποίο απεικονίζεται στην επόμενη Εικόνα:

Ανεπιτυχής καταχώρηση νοσοκομείου: Η ονομασία του νοσοκομείου την οποία πληκτρολογήσατε χρησιμοποιείται ήδη!!!

Δοκιμάστε ξανά

Εικόνα 56: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενης ονομασίας του νοσοκομείου

Το παράθυρο που δείχνεται στην πιο άνω Εικόνα ενημερώνει τον εκάστοτε διαχειριστή, ο οποίος είναι ο παρών χρήστης της διαδικτυακής εφαρμογής η οποία προδιαγράφεται στην τρέχουσα διατριβή, ότι, κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής νοσοκομείου της εν λόγω εφαρμογής καταχώρησε στο πεδίο «Όνομασία» αυτής της φόρμας ένα όνομα νοσοκομείου το οποίο είναι εγγεγραμμένο ήδη στο ανάλογο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί η ανώτερη εφαρμογή. Συνάμα, προσφέρει τη δυνατότητα στον εν λόγω διαχειριστή να καταχωρήσει στο πεδίο «Όνομασία» της προαναφερόμενης φόρμας ένα όνομα νοσοκομείου το οποίο δεν υπάρχει στην παραπάνω βάση, κάνοντας κλικ στο κουμπί «Δοκιμάστε ξανά» του πιο πάνω παραθύρου..

Επιπλέον, εφόσον, ο εκάστοτε διαχειριστής, ο οποίος εισέρχεται στο σύστημα το οποίο είναι η θεματολογία της παρούσας αναφοράς, κατά την ενέργεια συμπλήρωσης της φόρμας εγγραφής νοσοκομείου του προαναφερόμενου συστήματος, εισάγει τόσο στο πεδίο «Όνομα Χρήστη» όσο και στο πεδίο «Όνομασία» της προαναφερόμενης φόρμας τιμές οι οποίες έχουν ήδη καταχωρηθεί στα ανάλογα πεδία της βάσης δεδομένων με την οποία αλληλεπιδρά το εν λόγω σύστημα, τότε, πιέζοντας ο συγκεκριμένος διαχειριστής το κουμπί «Υποβολή» της πιο άνω φόρμας, θα φανεί το παράθυρο προειδοποίησης το οποίο αναπαρίσταται στην επόμενη Εικόνα:

Ανεπιτυχής καταχώρηση νοσοκομείου: Τόσο το όνομα χρήστη του νοσοκομείου όσο και η ονομασία του που πληκτρολογήσατε χρησιμοποιούνται ήδη!!!

Δοκιμάστε ξανά

Εικόνα 57: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενων ονόματος χρήστη και ονομασίας του νοσοκομείου

Το παράθυρο που εμφανίζεται στην παραπάνω Εικόνα ειδοποιεί τον εκάστοτε διαχειριστή, ο οποίος είναι ο λειτουργός του διαδικτυακού πληροφοριακού συστήματος το οποίο εξετάζεται στην παρούσα πτυχιακή, ότι, κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής νοσοκομείου αυτού του συστήματος, καταχώρησε τόσο στο πεδίο «Όνομα Χρήστη» όσο και στο πεδίο «Όνομασία» της προαναφερόμενης φόρμας τιμές οι οποίες προϋπάρχουν στα αντίστοιχα πεδία της βάσης δεδομένων με την οποία είναι σε επαφή το συγκεκριμένο λογισμικό. Την ίδια στιγμή, δίνει τη δυνατότητα στον προαναφερόμενο διαχειριστή να καταχωρήσει στα πρωτότερα πεδία της προαναφερόμενης φόρμας τιμές που δεν είναι εγγεγραμμένες στην παραπάνω βάση, πατώντας το κουμπί «Δοκιμάστε ξανά» του ανώτερου παραθύρου.

Εφόσον, βέβαια, ο εκάστοτε διαχειριστής, ο οποίος πραγματώνει είσοδο στην εφαρμογή η οποία αναπτύσσεται στο τρέχον έγγραφο, συμπληρώσει με έγκυρο τρόπο όλα τα πεδία της φόρμας εγγραφής νοσοκομείου της εν λόγω εφαρμογής, τότε, «κλικάροντας» ο συγκεκριμένος διαχειριστής το κουμπί «Υποβολή» της παραπάνω φόρμας, θα εμφανιστεί το παράθυρο επιβεβαίωσης το οποίο αποδίδεται στη μεταγενέστερη Εικόνα:

Επιτυχής καταχώρηση νοσοκομείου!!!

Εντάξει

Εικόνα 58: Το παράθυρο επιβεβαίωσης της επιτυχούς εγγραφής ενός νοσοκομείου στο λογισμικό από ένα διαχειριστή

Το παράθυρο που απεικονίζεται στην παραπάνω Εικόνα επιβεβαιώνει στον εκάστοτε διαχειριστή, ο οποίος αποτελεί τον τρέχων χρήστη του λογισμικού το οποίο είναι το αντικείμενο της τρέχουσας πτυχιακής εργασίας, πως συμπλήρωσε ορθά τα πεδία της φόρμας εγγραφής νοσοκομείου αυτού του λογισμικού και άρα καταχώρησε επιτυχημένα ένα καινούργιο νοσοκομείο στη βάση δεδομένων η οποία επικοινωνεί με αυτό το λογισμικό. Ακόμα, παρέχει στον παραπάνω διαχειριστή την ευχέρεια να εγγράψει εκ νέου είτε έναν ιατρό, είτε ένα νοσοκομείο, είτε έναν ασθενή στο προαναφερόμενο λογισμικό, διότι πιέζοντας ο εν λόγω διαχειριστής το κουμπί «Εντάξει» του ανώτερου παραθύρου, προκύπτει το παράθυρο διαχείρισης πληροφοριών του παρόντος λογισμικού.

Κλείνοντας τις ενέργειες στις οποίες είναι δυνατό να προβεί ο εκάστοτε διαχειριστής, ο οποίος μπαίνει στο διαδικτυακό λογισμικό το οποίο πραγματεύεται η τρέχουσα διπλωματική, θα γίνει μνεία στην περίπτωση κατά την οποία ο προαναφερόμενος διαχειριστής θα οφείλει να εγγράψει έναν ασθενή

στο εν λόγω λογισμικό. Σε μία τέτοια περίπτωση, λοιπόν, θα προβάλει το παράθυρο το οποίο δείχνεται στην ύστερη Εικόνα:

Φόρμα Εγγραφής Ασθενούς

ΑΜΚΑ	<input type="text"/>	[Πληκτρολογήστε τον Αριθμό Μητρώου Κοινωνικής Ασφάλισης (ΑΜΚΑ) του ασθενούς, ο οποίος πρέπει να περιέχει ακριβώς 11 ψηφία...]
Ασφάλεια	<input type="text"/>	[Πληκτρολογήστε το Ασφαλιστικό ταμείο του ασθενούς, το οποίο πρέπει να περιέχει από 2 έως 8 αποκλειστικά αλφαβητικούς χαρακτήρες...]
Όνομα	<input type="text"/>	[Πληκτρολογήστε το μικρό όνομα του ασθενούς, το οποίο πρέπει να περιέχει από 2 έως 25 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών ονομάτων!!!)]
Επώνυμο	<input type="text"/>	[Πληκτρολογήστε το επίθετο του ασθενούς, το οποίο πρέπει να περιέχει από 2 έως 35 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών επωνύμων!!!)]
Φύλο	<input type="radio"/> <input type="radio"/>	[Επιλέξτε το φύλο του ασθενούς: Άρρεν ή Θήλυ...]
Τηλέφωνο	<input type="text"/>	[Πληκτρολογήστε το σταθερό ή το κινητό τηλέφωνο του ασθενούς, το οποίο πρέπει να περιέχει ακριβώς 10 ψηφία...]
E-Mail	<input type="text"/>	[Πληκτρολογήστε μία έγκυρη διεύθυνση ηλεκτρονικού ταχυδρομείου (e-mail) του ασθενούς...]

Εικόνα 59: Το παράθυρο της φόρμας εγγραφής ενός ασθενούς στο λογισμικό από ένα διαχειριστή

Το παράθυρο που φαίνεται στην ανώτερη Εικόνα δίνει την ευκαιρία στον εκάστοτε διαχειριστή, ο οποίος είναι ο παρών χρήστης του συστήματος το οποίο ορίζεται στην παρούσα αναφορά, να εγγράψει έναν ασθενή στο εν λόγω λογισμικό, γεμίζοντας τα πεδία της φόρμας εγγραφής ασθενούς η οποία αναπαρίσταται στην πιο άνω Εικόνα και υποβάλλοντας στη συνέχεια την εν λόγω φόρμα στο διακομιστή Ιστού του προαναφερόμενου συστήματος, «κλικάροντας» το κουμπί «Υποβολή». Σε αυτό το σημείο, να παρατηρηθεί πως ο παρών διαχειριστής, κατά τη διαδικασία συμπλήρωσης των πεδίων της προγενέστερης φόρμας, θα πρέπει να συμμορφώνεται με τις προτροπές οι οποίες καταγράφονται στα δεξιά καθενός από τα αυτά πεδία. Ακόμα, δεν θα πρέπει σε ουδεμία περίπτωση να καταχωρήσει στο πεδίο «ΑΜΚΑ» της προαναφερόμενης φόρμας τιμές οι οποίες είναι ήδη γραμμένες στα αντίστοιχα πεδία της βάσης δεδομένων με την οποία επικοινωνεί το εν λόγω σύστημα, επειδή το ΑΜΚΑ ενός ασθενούς δίνει μία μοναδική ταυτότητα σε κάθε ασθενή, ο οποίος πρόκειται να εισαχθεί στο ανώτερο σύστημα.

Συνεπώς, αφότου ο εκάστοτε διαχειριστής, ο οποίος χειρίζεται την εφαρμογή η οποία παίρνει σάρκα και οστά στο παρόν έργο, κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής ασθενούς της εν λόγω εφαρμογής, καταχωρήσει στο πεδίο «ΑΜΚΑ» αυτής της φόρμας έναν ΑΜΚΑ ασθενούς ο οποίος είναι ήδη εγγεγραμμένος στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί η εν λόγω εφαρμογή, τότε, πιέζοντας ο παραπάνω διαχειριστής το κουμπί «Υποβολή» της προηγούμενης φόρμας θα προβάλει το παράθυρο προειδοποίησης το οποίο φαίνεται στην κατώτερη Εικόνα:

Ανεπιτυχής καταχώρηση ασθενούς: Ο ΑΜΚΑ του ασθενούς που πληκτρολογήσατε χρησιμοποιείται ήδη!!!

Δοκιμάστε ξανά

Εικόνα 60: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός ασθενούς στο λογισμικό από ένα διαχειριστή, λόγω ήδη χρησιμοποιούμενου ΑΜΚΑ του ασθενούς

Το παράθυρο που αντικατοπτρίζεται στην πιο πάνω Εικόνα προειδοποιεί τον εκάστοτε διαχειριστή, ο οποίος πραγματοποιεί είσοδο στο διαδικτυακό λογισμικό το οποίο συνιστά το θέμα της παρούσας εργασίας, πως κατά την πράξη συμπλήρωσης της φόρμας εγγραφής ασθενούς αυτού του λογισμικού εισήγαγε στο πεδίο «ΑΜΚΑ» της εν λόγω φόρμας έναν ΑΜΚΑ ασθενούς ο οποίος είναι ήδη καταχωρημένο στο ανάλογο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί το συγκεκριμένο λογισμικό. Την ίδια στιγμή, καθιστά ικανό στον τρέχον διαχειριστή να καταχωρήσει στο πεδίο «ΑΜΚΑ» της ανώτερης φόρμας έναν ΑΜΚΑ ασθενούς το οποίο δεν υπάρχει στην πρωτότερη βάση, πατώντας το κουμπί «Δοκιμάστε ξανά» του παραπάνω παραθύρου.

Εφόσον, φυσικά, ο εκάστοτε διαχειριστής, ο οποίος εισέρχεται στο σύστημα το οποίο αναλύεται στο τρέχον έγγραφο, συμπληρώσει με έγκυρο τρόπο όλα τα πεδία της φόρμας εγγραφής ασθενούς της προαναφερόμενης εφαρμογής, τότε, κάνοντας κλικ ο συγκεκριμένος διαχειριστής το κουμπί «Υποβολή» της ανώτερης φόρμας, θα αναδυθεί το παράθυρο επιβεβαίωσης το οποίο δείχνεται στη μεταγενέστερη Εικόνα:

Επιτυχής καταχώρηση ασθενούς!!!

Εντάξει

Εικόνα 61: Το παράθυρο επιβεβαίωσης της επιτυχούς εγγραφής ενός ασθενούς στο λογισμικό από ένα διαχειριστή

Το παράθυρο που δίδεται στην παραπάνω Εικόνα επιβεβαιώνει στον εκάστοτε διαχειριστή, ο οποίος αποτελεί τον τρέχον χρήστη του λογισμικού το οποίο είναι το αντικείμενο της παρούσας διπλωματικής αναφοράς, ότι συμπλήρωσε ορθά τα πεδία της φόρμας εγγραφής νοσοκομείου αυτού του λογισμικού και επομένως καταχώρησε με επιτυχία έναν νέο ασθενή στη βάση δεδομένων η οποία αλληλεπιδρά με αυτό το λογισμικό. Ακόμα, παρέχει στον προαναφερόμενο διαχειριστή την ευχέρεια να εγγράψει εκ νέου είτε έναν ιατρό, είτε ένα νοσοκομείο, είτε έναν ασθενή στο εν λόγω λογισμικό, διότι πατώντας ο εν λόγω διαχειριστής το κουμπί «Εντάξει» του πιο άνω παραθύρου, αναδύεται το παράθυρο διαχείρισης πληροφοριών του τρέχοντος λογισμικού.

Κάπου εδώ να αναφερθεί ότι στην κατώτερη περιοχή καθεμίας από τις φόρμες εγγραφής είτε ιατρού, είτε νοσοκομείου, είτε ασθενούς που απαντώνται στο σύστημα το οποίο καθορίζεται στην τρέχουσα αναφορά υπάρχουν, εκτός από το κουμπί «Υποβολή», η λειτουργία του οποίου αναπτύχθηκε πιο πάνω, και τα κουμπιά «Επαναφορά» και «Εξοδος», ο ρόλος των οποίων διαχωρίζεται ως εξής:

- Το μεν κουμπί «Επαναφορά» δίνει την ευκαιρία στον εκάστοτε διαχειριστή του λογισμικού, ο οποίος αποτελεί τον τρέχων χρήστη του λογισμικού το οποίο αναπτύσσεται στην παρούσα εργασία, εφόσον αυτός διαχειριστής πατήσει στο εν λόγω κουμπί, να σβήσει τα περιεχόμενα όλων των πεδίων καθεμίας από τις φόρμες εγγραφής ιατρού είτε νοσοκομείου είτε ασθενούς που συναντώνται στο προαναφερόμενο λογισμικό
- Το δε κουμπί «Εξοδος» παρέχει τη δυνατότητα στον εκάστοτε διαχειριστή, ο οποίος συνιστά τον παρών χρήστη της εφαρμογής η οποία υλοποιείται στην τρέχουσα αναφορά, να εγγράψει από την αρχή είτε έναν ιατρό, είτε ένα νοσοκομείο, είτε έναν ασθενή στο προαναφερόμενη εφαρμογή, διότι κάνοντας κλικ ο προαναφερόμενος διαχειριστής στο συγκεκριμένο κουμπί του εμφανίζεται το παράθυρο διαχείρισης πληροφοριών. (Το κουμπί «Εξοδος», δηλαδή, εξυπηρετεί ακριβώς τον ίδιο σκοπό που εξυπηρετεί και το κουμπί «Εντάξει» των παραθύρων επιβεβαίωσης που μελετήθηκαν παραπάνω.)

3.4.2 Η Είσοδος ενός Ιατρού στο Σύστημα Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Εκτός από τον εκάστοτε διαχειριστή του πληροφοριακού συστήματος το οποίο μελετάται στην τρέχουσα πτυχιακή διατριβή, θα δύναται να εισέρθει σε αυτό το σύστημα και ο εκάστοτε ιατρός ο οποίος είναι συμβεβλημένος με ένα ή περισσότερα Ασφαλιστικά Ταμεία του τομέα Υγείας μίας χώρας. Όπως ο εκάστοτε διαχειριστής, έτσι και ο εκάστοτε ιατρός, που θα επιθυμεί να εισέρθει στο εν λόγω σύστημα, θα πρέπει να καταχωρήσει το προσωπικό του όνομα χρήστη και τον προσωπικό του κωδικό στα ανάλογα πεδία της φόρμα εισόδου ιατρού του αρχικού παραθύρου αυτού του συστήματος. Κατόπιν, θα ακολουθήσει επαλήθευση των προαναφερόμενων στοιχείων με βάση τις τιμές των αντίστοιχων πεδίων οι οποίες υπάρχουν στη βάση δεδομένων με την οποία επικοινωνεί το πιο άνω σύστημα. Αν η προηγούμενη επαλήθευση αποτύχει, τότε θα εμφανιστεί το ίδιο ακριβώς παράθυρο προειδοποίησης το οποίο εμφανίζεται, όταν δεν ταυτοποιούνται τα στοιχεία του παραπάνω διαχειριστή. Αν, ωστόσο, η πρωτότερη επαλήθευση επιτύχει, τότε θα προβάλλει το παράθυρο το οποίο φαίνεται στην ύστερη Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων		
Κατηγορίες	Αρχικό Παράθυρο Ιατρού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Παρακαλούμε επιλέξτε μία από τις κατηγορίες...!!!</p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδάκης</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p><input type="button" value="Αποσύνδεση"/></p>
<p>ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ</p> <p style="text-align: right;">ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΩΓΙΑΣ (ΑΜ 679)</p>		

Εικόνα 62: Το αρχικό παράθυρο ενός ιατρού

Το παράθυρο που φαίνεται στην παραπάνω Εικόνα αποτελεί το αρχικό παράθυρο το οποίο αντικρίζει ένας ιατρός αμέσως μόλις εισέλθει στο σύστημα το οποίο περιγράφεται στο παρόν έγγραφο. Το συγκεκριμένο παράθυρο, μάλιστα, προδιαγράφει το καλούπι σύμφωνα με το οποίο είναι δομημένα όλα τα επιμέρους παράθυρα αυτής της εφαρμογής, που περιγράφουν τις ενέργειες τόσο του εκάστοτε ιατρού όσο και της εκάστοτε γραμματείας ενός νοσοκομείου, που έχουν τη δυνατότητα να εισέρχονται στην εν λόγω εφαρμογή. Έτσι, καθένα από τα προαναφερόμενα παράθυρα αποτελείται από τρεις στήλες, οι οποίες έχουν ως εξής:

- Η αριστερή στήλη είναι σταθερή για κάθε επιμέρους παράθυρο της εφαρμογής που αναπτύσσεται στο τρέχον έγγραφο, είτε αλληλεπιδρά με αυτή την εφαρμογή ένας ιατρός είτε η γραμματεία ενός νοσοκομείου, και παρουσιάζει τις κατηγορίες των διαφορετικών ενεργειών στις οποίες μπορεί να προβεί τόσο ο προαναφερόμενος διαχειριστής όσο και η προαναφερόμενη γραμματεία.
- Η δεξιά στήλη είναι, επίσης, σταθερή για κάθε επιμέρους παράθυρο της εφαρμογής που αναπτύσσεται στο τρέχον έγγραφο, είτε αλληλεπιδρά με αυτή την εφαρμογή ένας ιατρός είτε η γραμματεία ενός νοσοκομείου, και παρουσιάζει μερικά αναγνωριστικά στοιχεία του εκάστοτε τρέχοντος χρήστη της εν λόγω εφαρμογής, ο οποίος θα είναι είτε ένας ιατρός είτε η γραμματεία ενός νοσοκομείου. Επίσης, δίνει τη δυνατότητα και στους δύο προαναφερόμενους χρήστες, να αποσυνδέονται από την εν λόγω εφαρμογή, όποτε το θελήσουν, απλά πατώντας στο κουμπί «Αποσύνδεση» το οποίο απαντάται στην εν λόγω στήλη.
- Η μεσαία στήλη είναι αυτή που κάθε φορά τροποποιείται με κατάλληλο τρόπο κατά τη μετάβαση από ένα επιμέρους παράθυρο της εφαρμογής που αναπτύσσεται στο τρέχον έγγραφο σε ένα άλλο, είτε αλληλεπιδρά με αυτή την εφαρμογή ένας ιατρός είτε η γραμματεία ενός νοσοκομείου, προκειμένου να παρουσιάζει τη ροή της εν λόγω εφαρμογής. Από το σημείο αυτό της

παρούσας αναφοράς και στο εξής το όνομα της μεσαίας στήλης όλων των επιμέρους παραθύρων της προαναφερόμενης εφαρμογής, που παρουσιάζουν τις δραστηριότητες τόσο του εκάστοτε ιατρού όσο και της εκάστοτε γραμματείας ενός νοσοκομείου, που είναι ικανοί να πραγματοποιούν είσοδο σε αυτή την εφαρμογή, θα ταυτίζεται με το όνομα των συγκεκριμένων παραθύρων.

Το αρχικό παράθυρο, τώρα, που προβάλλει στον εκάστοτε ιατρού, ο οποίος έχει την εξουσιοδότηση να πραγματοποιήσει είσοδο στο λογισμικό που ορίζεται στην τρέχουσα εργασία, παρέχει την ευχέρεια σε αυτόν τον ιατρό να προχωρά κάθε φορά σε μία σειρά από ενέργειες, οι οποίες είναι οι κατοπινές:

- Να συντάσσει ένα παραπεμπτικό για τον εκάστοτε ασθενή, ο οποίος τον επισκέπτεται και φέρει οφθαλμολογικό πρόβλημα, το οποίο χρήζει ιδιαίζουσας αντιμετώπισης, προκειμένου αυτός ο ασθενής να εξεταστεί περαιτέρω σε κάποιο νοσοκομείο, είτε κρατικό είτε ιδιωτικό.
- Να ανακαλεί οποιαδήποτε στιγμή τα στοιχεία οποιουδήποτε παραπεμπτικού ασθενούς έχει συνταχθεί είτε από τον ίδιο είτε από τον άλλο ιατρό.
- Να είναι σε θέση ανά πάσα στιγμή να μελετά το ιστορικό ενός ασθενούς, το οποίο έχει να κάνει με την οφθαλμιατρική του υγεία.
- Να μπορεί σε κάθε χρονική στιγμή να ανατρέχει στα παραπεμπτικά ασθενών που ο ίδιος έχει συντάξει.

Εφόσον, τώρα, ο εκάστοτε ιατρός, που είναι ο τρέχων χρήστης του συστήματος που παρουσιάζεται στην παρούσα εργασία, πρέπει να συντάξει ένα παραπεμπτικό ασθενούς, θα οφείλει να κάνει κλικ στην κατηγορία «Σύνταξη Παραπεμπτικού» του αρχικού παραθύρου ιατρού του εν λόγω συστήματος. Τότε θα προβάλλει το παράθυρο το οποίο απεικονίζεται στη μετέπειτα Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων		
Κατηγορίες	Σύνταξη Παραπεμπτικού	Χρήστης
Σύνταξη Παραπεμπτικού	Προκειμένου να συντάξετε ένα νέο παραπεμπτικό, παρακαλούμε συμπληρώστε κατάλληλα τα κενά πεδία που ακολουθούν και κατόπιν πατήστε το κουμπί "Υποβολή" στο τέλος της τρέχουσας φόρμας:	Κωδικός Ιατρού: 1
Στοιχεία Παραπεμπτικού	ID Παραπεμπτικού: 4377946	Όνοματεπώνυμο Ιατρού: Νικόλαος Παππαδάκης
Ιστορικό Ασθενούς	ID Ιατρού: 1	Ειδικότητα Ιατρού: Οφθαλμίατρος
Ιστορικό Ιατρού	Όνομα Ιατρού: Νικόλαος	<input type="button" value="Αποσύνδεση"/>
	Επώνυμο Ιατρού: Παππαδάκης	
	ΑΜΚΑ Ασθενούς:	
	Ασφάλεια Ασθενούς:	
	Νοσοκομείο:	
	Αιτία Παραπομπής:	
	<input type="button" value="Υποβολή"/>	
	<i>ΠΡΟΣΟΧΗ!!! Παρακαλούμε κάθε φορά, εφόσον συντάξετε ένα παραπεμπτικό, αποσυνδεθείτε, πριν προχωρήσετε στη σύνταξη καινούργιου, για να λάβετε νέο ID παραπεμπτικού και να είστε έτσι σε θέση να συντάξετε καινούργιο παραπεμπτικό!!!</i>	
ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ		
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΑΣ (ΑΜ 679)		

Εικόνα 63: Το παράθυρο της φόρμας σύνταξης ενός παραπεμπτικού ασθενούς από έναν ιατρό

Το παράθυρο που φαίνεται στην παραπάνω Εικόνα δείχνει τη φόρμα σύνταξης παραπεμπτικού ασθενούς του λογισμικού το οποίο αναλύεται στο παρόν έργο από τον εκάστοτε ιατρό, ο οποίος δύναται

να χειρίζεται το εν λόγω λογισμικό. Έτσι, για να συντάξει κάθε φορά ένα καινούργιο παραπεμπτικό αυτός ο ιατρός, θα πρέπει να συμπληρώσει κατάλληλα τα κενά πεδία της προαναφερόμενης φόρμας (δε χρειάζεται να συμπληρώσει όλα τα πεδία της συγκεκριμένης φόρμας, διότι κάποια πεδία αυτής της φόρμας θα έχουν ήδη συμπληρωθεί με αυτόματο τρόπο, με το που θα προβάλλει το πιο πάνω παράθυρο) και να υποβάλει έπειτα τη συγκεκριμένη φόρμα στο διακομιστή Ιστού του προαναφερόμενου λογισμικού, «κλικάροντας» το κουμπί «Υποβολή».

Κάπου εδώ να τονιστεί πως το πεδίο «ID Παραπεμπτικού» της φόρμας σύνταξης παραπεμπτικού ασθενούς της εφαρμογή η οποία εξετάζεται στην παρούσα διατριβή, από έναν ιατρό, ο οποίος μπορεί να εισέρχεται σε αυτήν την εφαρμογή, παίρνει αυτόματα ως τιμή έναν επταψήφιο αριθμό, τη στιγμή που εμφανίζεται το παράθυρο της εν λόγω εφαρμογής που περικλείει την ανώτερη φόρμα. Ο συγκεκριμένος αριθμός παράγεται κάθε φορά που ο προαναφερόμενος ιατρός κάνει είσοδο στην τρέχουσα εφαρμογή και παραμένει σταθερός μέχρι αυτός ο ιατρός να αποσυνδεθεί από το εν λόγω λογισμικό. Επειδή, ωστόσο, κάθε παραπεμπτικό ασθενούς που δύναται να συντάξει ο προαναφερόμενος ιατρός δε μπορεί να έχει το ίδιο ID με κάποιο άλλο παραπεμπτικό ασθενούς που έχει ήδη συνταχθεί, είτε από τον εν λόγω ιατρό είτε από άλλο ιατρό, θεωρείται αναγκαίο κάθε φορά που ο εν λόγω ιατρός χρειάζεται να συντάξει ένα νέο παραπεμπτικό ασθενούς, να αποσυνδεθεί και να ξανασυνδεθεί στο παρόν λογισμικό, προκειμένου να προκύψει νέο ID παραπεμπτικού ασθενούς και είναι ικανός αυτός ο ιατρός να προχωρά στη σύνταξη καινούργιου παραπεμπτικού ασθενούς. Οπότε, άμα ο εν λόγω ιατρός έχει ήδη συντάξει ένα παραπεμπτικό ασθενούς και προβεί στη σύνταξη νέου, δίχως πρωτύτερα να έχει πραγματοποιήσει έξοδο από την παρούσα εφαρμογή, τότε, πιέζοντας ο προγενέστερος ιατρός το κουμπί «Υποβολή» της πρωτύτερης φόρμας, θα φανεί το παράθυρο που δείχνεται στην πιο κάτω Εικόνα:



Εικόνα 64: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός παραπεμπτικού ασθενούς στο λογισμικό από έναν ιατρό, λόγω ήδη χρησιμοποιούμενου ID του παραπεμπτικού ασθενούς

Το παράθυρο που παριστάνεται στην πιο πάνω Εικόνα συμβουλεύει τον εκάστοτε ιατρό, ο οποίος είναι ο τρέχων χρήστης του λογισμικού το οποίο αναλύεται στην παρούσα πτυχιακή, πως το πεδίο «ID Παραπεμπτικού» της φόρμας σύνταξης παραπεμπτικού ασθενούς του προγενέστερου λογισμικού έχει καταχωρημένη μία τιμή η οποία είναι εγγεγραμμένη ήδη στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία διασυνδέεται η τρέχουσα εφαρμογή. Κατόπιν, παρέχει την ευχέρεια στον εν λόγω ιατρό να αποσυνδεθεί από το συγκεκριμένο σύστημα και να επανασυνδεθεί σε αυτό, για να αποκτήσει το πεδίο «ID Παραπεμπτικού» της ανώτερης φόρμας μία νέα τιμή η οποία δεν συναντάται στην προηγούμενη βάση, «κλικάροντας» το κουμπί «Αποσύνδεση» του ανώτερου παραθύρου.

Επίσης, αν ο εκάστοτε ιατρός, που είναι συνδεδεμένος στο σύστημα το οποίο οριοθετείται στο παρόν έργο, κατά τη φάση συμπλήρωσης της φόρμας σύνταξης παραπεμπτικού ασθενούς αυτού του συστήματος, καταχωρήσει στο πεδίο «ΑΜΚΑ Ασθενούς» της εν λόγω φόρμας έναν ΑΜΚΑ ασθενούς ο οποίος δεν προϋπάρχει στο ανάλογο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί το εν λόγω σύστημα, τότε, κάνοντας κλικ ο συγκεκριμένος ιατρός στο κουμπί «Υποβολή» της πρωτύτερης φόρμας, θα προκύψει το παράθυρο προειδοποίησης το οποίο απεικονίζεται στην ύστερη Εικόνα:

Ανεπιτυχής καταχώρηση παραπεμπτικού: Ο ΑΜΚΑ που πληκτρολογήσατε δεν αντιστοιχεί σε κάποιον ασθενή!!!

Δοκιμάστε ξανά

Εικόνα 65: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός παραπεμπτικού ασθενούς στο λογισμικό από έναν ιατρό, λόγω ήδη χρησιμοποιούμενου ΑΜΚΑ του ασθενούς

Το παράθυρο που αντικατοπτρίζεται στην ανώτερη Εικόνα προειδοποιεί τον εκάστοτε ιατρό, οποίος είναι ο τρέχων χρήστης του συστήματος το οποίο αναφέρεται στην παρούσα διατριβή, πως εισήγαγε στο πεδίο «ΑΜΚΑ Ασθενούς» της φόρμας σύνταξης παραπεμπτικού ασθενούς του προγενέστερου συστήματος έναν ΑΜΚΑ ασθενούς ο οποίος βρίσκεται ήδη καταχωρημένο στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία αλληλεπιδρά η συγκεκριμένη εφαρμογή. Ύστερα, δίνει την ευχέρεια στον εν λόγω ιατρό να εισάγει στο πεδίο «ΑΜΚΑ Ασθενούς» της προαναφερόμενης φόρμας έναν ΑΜΚΑ ασθενούς ο οποίος δε συναντάται στην παραπάνω βάση, πατώντας το κουμπί «Δοκιμάστε ξανά» του πιο πάνω παραθύρου.

Επιπρόσθετα, άμα ο εκάστοτε ιατρός, που είναι συνδεδεμένος στο σύστημα λογισμικού το οποίο προδιαγράφεται στην τρέχουσα εργασία, κατά τη φάση συμπλήρωσης της φόρμας σύνταξης παραπεμπτικού ασθενούς αυτού του λογισμικού, καταχωρήσει στο πεδίο «Ασφάλεια Ασθενούς» της εν λόγω φόρμας ένα Φορέα Ασφάλισης ασθενούς ο οποίος δε συνδέεται με τον ΑΜΚΑ ασθενούς ο οποίος είναι γραμμένος στο πεδίο «ΑΜΚΑ Ασθενούς» αυτής της φόρμας, τότε, πατώντας ο συγκεκριμένος ιατρός το κουμπί «Υποβολή» της συγκεκριμένης φόρμας, θα αναδυθεί το παράθυρο προειδοποίησης το οποίο παρουσιάζεται στην κατοπινή Εικόνα:

Ανεπιτυχής καταχώρηση παραπεμπτικού: Ο ασθενής με τον ΑΜΚΑ που πληκτρολογήσατε δεν είναι συμβεβλημένος με το Ασφαλιστικό Ταμείο το οποίο πληκτρολογήσατε!!!

Δοκιμάστε ξανά

Εικόνα 66: Το παράθυρο προειδοποίησης της ανεπιτυχούς εγγραφής ενός παραπεμπτικού ασθενούς στο λογισμικό από έναν ιατρό, λόγω λανθασμένης σύνδεσης του ΑΜΚΑ του ασθενούς με την Ασφάλεια του ασθενούς

Το παράθυρο που δείχνεται στην πιο πάνω Εικόνα πληροφορεί τον εκάστοτε ιατρό, ο οποίος είναι ο παρών χρήστης του λογισμικού το οποίο πραγματεύεται η τρέχουσα πτυχιακή, πως καταχώρησε στο πεδίο «Ασφάλεια Ασθενούς» της φόρμας σύνταξης παραπεμπτικού ασθενούς αυτού του λογισμικού μία Ασφάλεια ασθενούς η οποία δεν αντιστοιχεί στο ΑΜΚΑ ασθενούς το οποίο είναι καταχωρημένο στο πεδίο «ΑΜΚΑ Ασθενούς» της εν λόγω φόρμας της βάσης δεδομένων με την οποία αλληλεπιδρά η συγκεκριμένη εφαρμογή. Έπειτα, καθιστά δυνατό στον προαναφερόμενο ιατρό να εισάγει στο πεδίο «Ασφάλεια Ασθενούς» της προαναφερόμενης φόρμας μία Ασφάλεια ασθενούς η οποία συμφωνεί με τον ΑΜΚΑ ασθενούς ο οποίος είναι εγγεγραμμένος στο πεδίο «ΑΜΚΑ Ασθενούς» της παραπάνω φόρμας, κάνοντας κλικ στο κουμπί Δοκιμάστε ξανά» του ανώτερου παραθύρου.

Αφότου, φυσικά, ο εκάστοτε ιατρός, που χειρίζεται το διαδικτυακό λογισμικό το οποίο εξετάζεται στην παρούσα εργασία, συμπληρώσει με ορθό τρόπο όλα τα πεδία της φόρμας σύνταξης παραπεμπτικού ασθενούς αυτού του λογισμικού, τότε, πιέζοντας ο πιο άνω ιατρός το κουμπί «Υποβολή» της εν λόγω φόρμας, θα προκύψει το παράθυρο επιβεβαίωσης το οποίο δείχνεται στην πιο κάτω Εικόνα:

Επιτυχής καταχώρηση παραπεμπτικού!!!

Εντάξει

Εικόνα 67: Το παράθυρο επιβεβαίωσης της επιτυχούς καταχώρησης ενός παραπεμπτικού ασθενούς στο λογισμικό από έναν ιατρό

Το παράθυρο που αναπαρίσταται στην πιο άνω Εικόνα βεβαιώνει τον εκάστοτε ιατρό, ο οποίος σε είναι συνδεδεμένος στην εφαρμογή που αποτελεί τη θεματολογία τούτης της διατριβής, ότι συμπλήρωσε έγκυρα όλα τα πεδία της φόρμας σύνταξης παραπεμπτικού αυτής της εφαρμογής και άρα καταχώρησε με επιτυχία ένα καινούργιο παραπεμπτικό ασθενούς στη βάση δεδομένων η οποία είναι σε επικοινωνία με την εν λόγω εφαρμογή. Ακόμα, παρέχει στον προαναφερόμενο ιατρό τη δυνατότητα να μεταβεί στο αρχικό παράθυρο ιατρού στη συγκεκριμένη εφαρμογή, πατώντας το κουμπί «Εντάξει» του παραπάνω παραθύρου.

Στην περίπτωση, τώρα, που ο εκάστοτε ιατρός, ο οποίος έχει εισέλθει στο διαδικτυακό σύστημα, το οποίο αναλύεται στην τρέχουσα αναφορά, θεωρήσει απαραίτητο να αναζητήσει τα στοιχεία ενός παραπεμπτικού ασθενούς, το οποίο έχει συντάξει είτε ο ίδιος είτε οποιοσδήποτε άλλος ιατρός, θα πρέπει να πατήσει στην κατηγορία «Στοιχεία Παραπεμπτικού» της αριστερής στήλης του αρχικού παραθύρου ιατρού του προαναφερόμενου συστήματος. Τότε θα αναδυθεί το παράθυρο το οποίο απεικονίζεται στην παρακάτω Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων		
Κατηγορίες	Στοιχεία Παραπεμπτικού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενή Ιστορικό Ιατρού 	<p>Προκειμένου να αναζητήσετε τα στοιχεία ενός παραπεμπτικού, το οποίο έχει συνταχθεί είτε από εσάς είτε από οποιονδήποτε άλλον ιατρό, παρακαλούμε πληκτρολογήστε το ID του στο αμέσως επόμενο πλαίσιο κειμένου:</p> <p><input type="text"/></p> <p>και κατόπιν πατήστε στο κουμπί "Αναζήτηση":</p> <p><input type="button" value="Αναζήτηση"/></p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπασάκης</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p><input type="button" value="Αποσύνδεση"/></p>
<p>ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ</p> <p>ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)</p>		

Εικόνα 68: Το παράθυρο της αναζήτησης των στοιχείων ενός παραπεμπτικού ασθενούς από έναν ιατρό

Το παράθυρο που αντικατοπτρίζεται στην πιο πάνω Εικόνα δίνει τη δυνατότητα στον εκάστοτε ιατρό, ο οποίος αλληλεπιδρά με το λογισμικό το οποίο είναι το αντικείμενο μελέτης της τρέχουσας αναφοράς, να αναζητήσει τα στοιχεία οποιουδήποτε παραπεμπτικού ασθενούς, εισάγοντας στο πεδίο κειμένου που υπάρχει στο παραπάνω παράθυρο το ID του προς αναζήτηση κάθε φορά παραπεμπτικού και πατώντας μετά στο κουμπί «Αναζήτηση» που υπάρχει στην κάτω περιοχή του πρωτύπου παραθύρου.

Ωστόσο, αν ο εκάστοτε ιατρός, που είναι ο παρών χρήστης της εφαρμογής που αναπτύσσεται στο τρέχον έργο, εισάγει στο πεδίο κειμένου που περιέχεται στο παράθυρο αναζήτησης στοιχείων παραπεμπτικού ασθενούς της προαναφερόμενης εφαρμογής ένα ID παραπεμπτικού που δεν υφίσταται στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί η πρωτότερη εφαρμογή, τότε, «κλικάροντας» ο συγκεκριμένος ιατρός στο κουμπί «Αναζήτηση» του πρωτύπου παραθύρου, θα ξεπροβάλλει το παράθυρο της ύστερης Εικόνας:

Δε βρέθηκε παραπεμπτικό με το ID που πληκτρολογήσατε!!!

Εικόνα 69: Το παράθυρο προειδοποίησης της ανεπιτυχούς αναζήτησης ενός παραπεμπτικού ασθενούς από έναν ιατρό

Το παράθυρο που παριστάνεται στην πιο άνω Εικόνα ειδοποιεί τον εκάστοτε ιατρό, ο οποίος αλληλεπιδρά με το λογισμικό το οποίο περιγράφεται στην παρούσα πτυχιακή, ότι το ID παραπεμπτικού ασθενή το οποίο έγραψε στο πεδίο κειμένου που είναι τοποθετημένο στο παράθυρο αναζήτησης στοιχείων παραπεμπτικού ασθενούς του προαναφερόμενου λογισμικού δεν είναι καταχωρημένο στο ανάλογο πεδίο της βάσης δεδομένων με την οποία συνδέεται αυτό ο λογισμικό. Ακόμα προσφέρει τη δυνατότητα στον προηγούμενο ιατρό να εισάγει στο παραπάνω πεδίο κειμένου ένα ID παραπεμπτικού ασθενούς το οποίο υπάρχει στο ανάλογο πεδίο της προγενέστερης βάσης, πιέζοντας το κουμπί «Δοκιμάστε ξανά» του ανώτερου παραθύρου.

Από τη στιγμή, βέβαια, που ο εκάστοτε ιατρός, ο οποίος έχει εισέλθει στο πρόγραμμα το οποίο αναπτύσσεται στην τρέχουσα διπλωματική, γράψει στο πεδίο κειμένου που υπάρχει στο παράθυρο αναζήτησης στοιχείων παραπεμπτικού ασθενούς αυτού του προγράμματος ένα έγκυρο ID παραπεμπτικού ασθενούς, τότε, πιέζοντας ο προγενέστερος ιατρός το κουμπί «Αναζήτηση» του προηγούμενου παραθύρου, θα εμφανιστούν τα στοιχεία του παραπεμπτικού ασθενούς το οποίο φέρει το προαναφερόμενο ID και δύναται να έχει συνταχθεί είτε από τον εν λόγω ιατρό είτε από άλλον ιατρό, όπως φαίνεται, αντίστοιχα, στις δύο ακόλουθες Εικόνες:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Στοιχεία Παραπεμπτικού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Τα στοιχεία του παραπεμπτικού με ID 4377946 είναι τα ακόλουθα:</p> <p>ID Παραπεμπτικού: <input type="text" value="4377946"/></p> <p>ID Ιατρού: <input type="text" value="1"/></p> <p>Όνομα Ιατρού: <input type="text" value="Νικόλαος"/></p> <p>Επώνυμο Ιατρού: <input type="text" value="Παπαδόκης"/></p> <p>ΑΜΚΑ Ασθενούς: <input type="text" value="01014025894"/></p> <p>Ασφάλεια Ασθενούς: <input type="text" value="ΟΓΑ"/></p> <p>ID Νοσοκομείου: <input type="text" value="10000"/></p> <p>Όνομασία Νοσοκομείου: <input type="text" value="Υγεία"/></p> <p>Αιτία Παραπομπής: <input style="width: 100%; height: 40px;" type="text" value="Εκφύλιση ωχράς κηλίδας."/></p> <p style="text-align: center;"><input type="button" value="Εντάξει"/></p>	<p>Κωδικός Ιατρού: <input type="text" value="1"/></p> <p>Όνοματεπώνυμο Ιατρού: <input type="text" value="Νικόλαος Παπαδόκης"/></p> <p>Ειδικότητα Ιατρού: <input type="text" value="Οφθαλμίατρος"/></p> <p style="text-align: center;"><input type="button" value="Αποσύνδεση"/></p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 70: Το παράθυρο εμφάνισης των στοιχείων ενός παραπεμπτικού ασθενούς, το οποίο έχει συντάξει ο ιατρός ο οποίος είναι κάθε φορά ο τρέχων χρήστης του λογισμικού

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

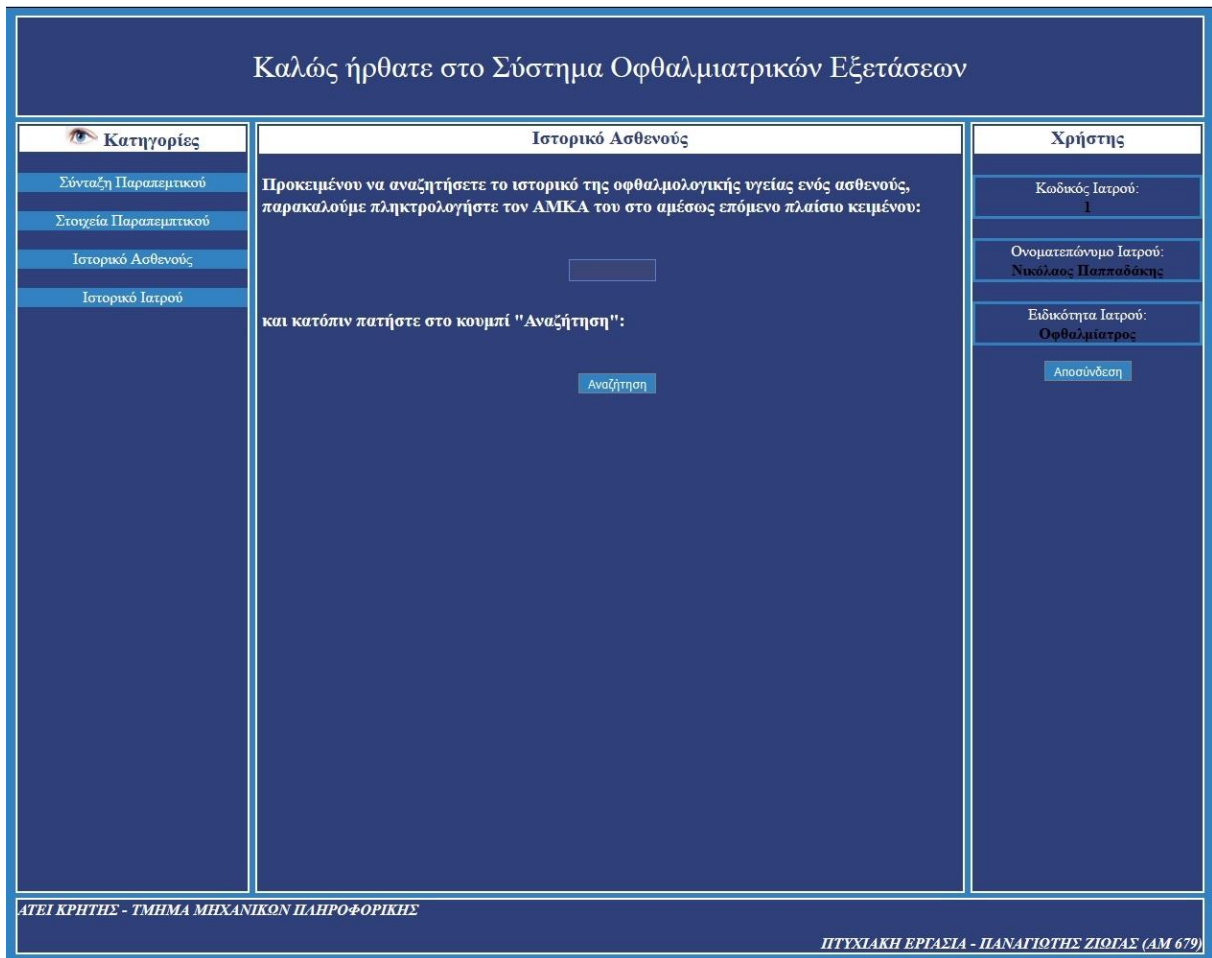
Κατηγορίες	Στοιχεία Παραπεμπτικού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Τα στοιχεία του παραπεμπτικού με ID 2213454 είναι τα ακόλουθα:</p> <p>ID Παραπεμπτικού: 2213454</p> <p>ID Ιατρού: 3</p> <p>Όνομα Ιατρού: Παναγιώτης</p> <p>Επώνυμο Ιατρού: Νικηφόρου</p> <p>ΑΜΚΑ Ασθενούς: 18759641223</p> <p>Ασφάλεια Ασθενούς: ΔΗΜΟΣΙΟ</p> <p>ID Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p>Αιτία Παραπομπής: <input type="text" value="Λοσιγμοεισμός"/></p> <p style="text-align: center;"><input type="button" value="Εντάξει"/></p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παππούδης</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p style="text-align: center;"><input type="button" value="Αποσύνδεση"/></p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΩΓΙΑΣ (ΑΜ 679)

Εικόνα 71: Το παράθυρο εμφάνισης των στοιχείων ενός παραπεμπτικού ασθενούς, το οποίο έχει συντάξει ένας διαφορετικός ιατρός από τον ιατρό ο οποίος είναι κάθε φορά ο τρέχων χρήστης του λογισμικού

Το μεν πρώτο από τα δύο ανώτερα παράθυρα δείχνει τα στοιχεία ενός παραπεμπτικού ασθενούς, το οποίο έχει συνταχθεί είτε από τον εκάστοτε ιατρό, ο οποίος αποτελεί τον τρέχων χρήστη του συστήματος το οποίο καθορίζεται στην παρούσα εργασία. Το δε δεύτερο από τα δύο πιο άνω παράθυρα εμφανίζει τα στοιχεία ενός παραπεμπτικού ασθενούς, το οποίο έχει συνταχθεί είτε από έναν διαφορετικό σε σχέση με τον προαναφερόμενο ιατρό. Ωστόσο, και τα δύο πιο πάνω παράθυρα παρέχουν στον παρών ιατρό να μεταφερθεί στο αρχικό παράθυρο ιατρού αυτού του εν λόγω συστήματος, πατώντας στο κουμπί «Εντάξει» που υπάρχει και στα δύο παραπάνω παράθυρα.

Αν, τώρα, ο εκάστοτε ιατρός, ο οποίος έχει πραγματώσει είσοδο στο λογισμικό, το οποίο ορίζεται στην τρέχουσα διπλωματική, θελήσει να αναζητήσει το σχετιζόμενο με την υγεία των ματιών ιστορικό ενός ασθενούς, θα πρέπει να επιλέξει την κατηγορία «Ιστορικό Ασθενούς» της αριστερής στήλης του αρχικού παραθύρου ιατρού του προαναφερόμενου λογισμικού. Τότε θα εμφανιστεί το παράθυρο το οποίο αναπαρίσταται στην παρακάτω Εικόνα:



Εικόνα 72: Το παράθυρο της αναζήτησης του ιστορικού ενός ασθενούς από έναν ιατρό

Το παράθυρο που αποδίδεται στην πιο πάνω Εικόνα καθιστά ικανό τον εκάστοτε ιατρό, ο οποίος σε αλληλεπιδρά με την εφαρμογή η οποία μελετάται στην τρέχουσας εργασία, να ψάξει το ιστορικό οποιουδήποτε ασθενούς, σχετικά με την οφθαλμιατρική υγεία αυτού του ασθενούς, εισάγοντας στο πεδίο κειμένου που απαντάται στο πιο άνω παράθυρο τον ΑΜΚΑ του εν λόγω ασθενούς και πατώντας κατόπιν στο κουμπί «Αναζήτηση» που υπάρχει στο κάτω μέρος του πιο πάνω παραθύρου.

Πάραυτα, αν ο εκάστοτε ιατρός, που χειρίζεται την εφαρμογή η οποία αναπτύσσεται στο τρέχον έργο, καταχωρήσει στο πεδίο κειμένου το οποίο υπάρχει στο παράθυρο αναζήτησης ιστορικού ασθενούς (από ιατρό) αυτής της εφαρμογής έναν ΑΜΚΑ ασθενούς, ο οποίος δε συναντάται στο ανάλογο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί η πρωτύτερη εφαρμογή, τότε πιέζοντας ο παρών ιατρός το κουμπί «Αναζήτηση» του πιο πάνω παραθύρου, θα προβάλλει το παράθυρο της μεταγενέστερης Εικόνας:



Εικόνα 73: Το παράθυρο προειδοποίησης της ανεπιτυχούς αναζήτησης του ιστορικού ενός ασθενούς από έναν ιατρό

Το παράθυρο που αποδίδεται στην ανώτερη Εικόνα προειδοποιεί τον εκάστοτε ιατρό, ο οποίος χειρίζεται το πρόγραμμα το οποίο περιγράφεται στην παρούσα πτυχιακή, πως ο ΑΜΚΑ ασθενούς τον οποίο κατέγραψε στο πεδίο κειμένου το οποίο βρίσκεται στο παράθυρο αναζήτησης ιστορικού ασθενούς (από ιατρό) του εν λόγω προγράμματος δεν είναι καταχωρημένο στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία διασυνδέεται αυτό το λογισμικό. Επίσης, προσφέρει την ευχέρεια στον προηγούμενο ιατρό να εγγράψει στο προγενέστερο πεδίο κειμένου έναν ΑΜΚΑ ασθενούς ο οποίος είναι εγγεγραμμένος στο αντίστοιχο πεδίο της πρωτύτης βάσης, πατώντας στο κουμπί «Δοκιμάστε ξανά» του πιο πάνω παραθύρου.

Εφόσον, φυσικά, ο εκάστοτε ιατρός, ο οποίος έχει εισέλθει στο σύστημα λογισμικού το οποίο προδιαγράφεται στην τρέχουσα εργασία, εισάγει στο πεδίο κειμένου το οποίο υπάρχει στο παράθυρο αναζήτησης ιστορικού ασθενούς (από ιατρό) αυτού του συστήματος έναν έγκυρο ΑΜΚΑ ασθενούς, τότε, πατώντας ο προαναφερόμενος ιατρός στο κουμπί «Αναζήτηση» του πρωτύτου παραθύρου, ανάλογα με το αν ο προαναφερόμενος ασθενής έχει ή δεν έχει έστω και μία φορά παραπεμφθεί από οποιονδήποτε ιατρό σε κάποια οφθαλμιατρική κλινική, για περαιτέρω εξετάσεις, θα αναδυθούν, αντίστοιχα, τα παράθυρα που φαίνονται στις δύο πιο κάτω Εικόνες:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενούς	Χρήστης																														
Σύνταξη Παραπεμπτικού	<p>Το ιστορικό του ασθενούς με ΑΜΚΑ 01014025894 φαίνεται στον ακόλουθο πίνακα:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ΑΜΚΑ Ασθενούς</th> <th>Ασφάλεια Ασθενούς</th> <th>Αιτία Παραπομπής</th> <th>Όνομασία Νοσοκομείου</th> <th>Ημερομηνία εξέτασης</th> <th>Αποτέλεσμα εξέτασης</th> </tr> </thead> <tbody> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Μυωπία.</td> <td>Υγεία</td> <td>02/02/1980</td> <td>Αφαίρεση με χειρουργείο.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Καταράκτης.</td> <td>Υγεία</td> <td>12/03/2012</td> <td>Αφαίρεση με λέιζερ.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Πίεση.</td> <td>ΙΑΣΩ Θεσσαλίας</td> <td>14/05/2015</td> <td>Φαρμακευτική αγωγή.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Εκφύλιση αχράς κηλίδας.</td> <td>Υγεία</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> </tbody> </table> <p style="text-align: center; color: blue; font-weight: bold;">Εντάξει</p>	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Αιτία Παραπομπής	Όνομασία Νοσοκομείου	Ημερομηνία εξέτασης	Αποτέλεσμα εξέτασης	01014025894	ΟΓΑ	Μυωπία.	Υγεία	02/02/1980	Αφαίρεση με χειρουργείο.	01014025894	ΟΓΑ	Καταράκτης.	Υγεία	12/03/2012	Αφαίρεση με λέιζερ.	01014025894	ΟΓΑ	Πίεση.	ΙΑΣΩ Θεσσαλίας	14/05/2015	Φαρμακευτική αγωγή.	01014025894	ΟΓΑ	Εκφύλιση αχράς κηλίδας.	Υγεία	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!	Κωδικός Ιατρού: 1
ΑΜΚΑ Ασθενούς		Ασφάλεια Ασθενούς	Αιτία Παραπομπής	Όνομασία Νοσοκομείου	Ημερομηνία εξέτασης	Αποτέλεσμα εξέτασης																										
01014025894		ΟΓΑ	Μυωπία.	Υγεία	02/02/1980	Αφαίρεση με χειρουργείο.																										
01014025894		ΟΓΑ	Καταράκτης.	Υγεία	12/03/2012	Αφαίρεση με λέιζερ.																										
01014025894		ΟΓΑ	Πίεση.	ΙΑΣΩ Θεσσαλίας	14/05/2015	Φαρμακευτική αγωγή.																										
01014025894	ΟΓΑ	Εκφύλιση αχράς κηλίδας.	Υγεία	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!																											
Στοιχεία Παραπεμπτικού		Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος																														
Ιστορικό Ασθενούς		Ειδικότητα Ιατρού: Οφθαλμιατρός																														
Ιστορικό Ιατρού		Αποσύνδεση																														

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 74: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος έχει παραπεμφθεί έστω μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περαιτέρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από έναν ιατρό

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενούς	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Για τον ασθενή με ΑΜΚΑ 04055536363 δεν έχει συνταχθεί κανένα παραπεμπτικό και συνεπώς δεν έχει πραγματοποιήσει καμία εξέταση!!!</p> <p style="text-align: center;">Εντάξει</p>	<ul style="list-style-type: none"> Κωδικός Ιατρού: 1 Όνοματεπώνυμο Ιατρού: Νικόλαος Παπασάκης Ειδικότητα Ιατρού: Οφθαλμίατρος Αποσύνδεση

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΥΓΑΣ (ΑΜ 679)

Εικόνα 75: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος δεν έχει παραπεμφθεί ούτε μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περεταίρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από έναν ιατρό

Το μεν πρώτο από τα δύο πιο άνω παράθυρα αναπαριστά το ιστορικό της οφθαλμιατρικής υγείας ενός ασθενούς, ο οποίος έχει τουλάχιστον μία φορά παραπεμφθεί από οποιονδήποτε ιατρό σε κάποια οφθαλμολογική κλινική. Το δε δεύτερο από τα δύο ανώτερα παράθυρα προβάλλει το ιστορικό της οφθαλμιατρικής υγείας ενός ασθενούς, ο οποίος καμία φορά δεν παραπεμφθεί από οποιονδήποτε ιατρό σε κάποιο νοσοκομείο. Στο σημείο τούτο, να τονιστεί ότι και στα δύο ανώτερα παράθυρα η αναζήτηση του ιστορικού ασθενούς πραγματώνεται από τον εκάστοτε ιατρό, ο οποίος εισέρχεται στο σύστημα το οποίο παρουσιάζεται στην παρούσα διπλωματική. Επιπρόσθετα, και τα δύο πιο πάνω παράθυρα προσφέρουν στον προαναφερόμενο ιατρό την ευκαιρία να μετακινηθεί στο αρχικό παράθυρο ιατρού αυτού του συστήματος, κάνοντας κλικ στο κουμπί «Εντάξει» που υπάρχει και στα δύο πιο άνω παράθυρα.

Κλείνοντας, άμα, ο εκάστοτε ιατρός, που χειρίζεται την εφαρμογή Διαδικτύου, η οποία καθορίζεται στην παρούσα αναφορά, επιθυμήσει να κάνει αναζήτηση των παραπεμπτικών ασθενών που αυτός ο ίδιος ιατρός έχει συντάξει, θα χρειαστεί να διαλέξει την κατηγορία «Ιστορικό Ιατρού» της αριστερής στήλης του αρχικού παραθύρου ιατρού της εν λόγω εφαρμογής. Τότε θα ξεπροβάλλει το παράθυρο το οποίο δίδεται στην ύστερη Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων														
<table border="1"> <thead> <tr> <th>Κατηγορίες</th> </tr> </thead> <tbody> <tr><td>Σύνταξη Παραπεμπτικού</td></tr> <tr><td>Στοιχεία Παραπεμπτικού</td></tr> <tr><td>Ιστορικό Ασθενούς</td></tr> <tr><td>Ιστορικό Ιατρού</td></tr> </tbody> </table>	Κατηγορίες	Σύνταξη Παραπεμπτικού	Στοιχεία Παραπεμπτικού	Ιστορικό Ασθενούς	Ιστορικό Ιατρού	<table border="1"> <thead> <tr> <th>Ιστορικό Ιατρού</th> </tr> </thead> <tbody> <tr> <td> <p>Προκειμένου να αναζητήσετε το ιστορικό των παραπεμπτικών τα οποία αποκλειστικά εσείς έχετε συντάξει, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;"><input type="button" value="Αναζήτηση"/></p> </td> </tr> </tbody> </table>	Ιστορικό Ιατρού	<p>Προκειμένου να αναζητήσετε το ιστορικό των παραπεμπτικών τα οποία αποκλειστικά εσείς έχετε συντάξει, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;"><input type="button" value="Αναζήτηση"/></p>	<table border="1"> <thead> <tr> <th>Χρήστης</th> </tr> </thead> <tbody> <tr> <td>Κωδικός Ιατρού: 1</td> </tr> <tr> <td>Όνοματεπώνυμο Ιατρού: Νικόλαος Παππαδάκης</td> </tr> <tr> <td>Ειδικότητα Ιατρού: Οφθαλμίατρος</td> </tr> <tr> <td style="text-align: center;"><input type="button" value="Αποσύνδεση"/></td> </tr> </tbody> </table>	Χρήστης	Κωδικός Ιατρού: 1	Όνοματεπώνυμο Ιατρού: Νικόλαος Παππαδάκης	Ειδικότητα Ιατρού: Οφθαλμίατρος	<input type="button" value="Αποσύνδεση"/>
Κατηγορίες														
Σύνταξη Παραπεμπτικού														
Στοιχεία Παραπεμπτικού														
Ιστορικό Ασθενούς														
Ιστορικό Ιατρού														
Ιστορικό Ιατρού														
<p>Προκειμένου να αναζητήσετε το ιστορικό των παραπεμπτικών τα οποία αποκλειστικά εσείς έχετε συντάξει, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;"><input type="button" value="Αναζήτηση"/></p>														
Χρήστης														
Κωδικός Ιατρού: 1														
Όνοματεπώνυμο Ιατρού: Νικόλαος Παππαδάκης														
Ειδικότητα Ιατρού: Οφθαλμίατρος														
<input type="button" value="Αποσύνδεση"/>														
<p>ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ</p> <p style="text-align: right;">ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)</p>														

Εικόνα 76: Το παράθυρο της αναζήτησης του ιστορικού ενός ιατρού από τον ίδιο τον ιατρό

Το παράθυρο που δείχνεται στην πιο πάνω Εικόνα καθιστά εφικτό στον εκάστοτε ιατρό, ο οποίος συνδέεται με το πληροφοριακό σύστημα το οποίο μελετάται στην τρέχουσα διατριβή, να προβάλλει όλα τα παραπεμπτικά ασθενών τα οποία έχει συντάξει αυτός ο ίδιος ιατρός. Πατώντας στο κουμπί «Αναζήτηση» που υπάρχει στο παραπάνω παράθυρο. Τότε, ανάλογα με το αν ο προαναφερόμενος ιατρός έχει ή δεν έχει συντάξει κάποιο παραπεμπτικό ασθενούς, θα αναδυθούν, αντίστοιχα, τα παράθυρα που αντικατοπτρίζονται στις δύο πιο κάτω Εικόνες:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ιατρού	Χρήστης																																				
Σύνταξη Παραπεμπτικού	Το ιστορικό σας φαίνεται στον ακόλουθο πίνακα:	Κωδικός Ιατρού: 1																																				
Στοιχεία Παραπεμπτικού		Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόκης																																				
Ιστορικό Ασθενούς		Ειδικότητα Ιατρού: Οφθαλμίατρος																																				
Ιστορικό Ιατρού		<input type="button" value="Αποσύνδεση"/>																																				
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ID Ιατρού</th> <th style="width: 15%;">ID Παραπεμπτικού</th> <th style="width: 15%;">ΑΜΚΑ Ασθενούς</th> <th style="width: 15%;">Ασφάλεια Ασθενούς</th> <th style="width: 15%;">Όνομασία Νοσοκομείου</th> <th style="width: 15%;">Αιτία Παραπομπής</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1202016</td> <td style="text-align: center;">01014025894</td> <td style="text-align: center;">ΟΓΑ</td> <td style="text-align: center;">Υγεία</td> <td style="text-align: center;">Μυωπία.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">4377946</td> <td style="text-align: center;">01014025894</td> <td style="text-align: center;">ΟΓΑ</td> <td style="text-align: center;">Υγεία</td> <td style="text-align: center;">Εκφύλιση ωχράς κηλίδας.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">6124477</td> <td style="text-align: center;">18759641223</td> <td style="text-align: center;">ΔΗΜΟΣΙΟ</td> <td style="text-align: center;">Υγεία</td> <td style="text-align: center;">Κερατόκωνος.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">6427958</td> <td style="text-align: center;">14058321543</td> <td style="text-align: center;">ΙΚΑ</td> <td style="text-align: center;">Υγεία</td> <td style="text-align: center;">Πίεση.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">7627958</td> <td style="text-align: center;">18759641223</td> <td style="text-align: center;">ΔΗΜΟΣΙΟ</td> <td style="text-align: center;">Υγεία</td> <td style="text-align: center;">Γλαύκωμα.</td> </tr> </tbody> </table>	ID Ιατρού	ID Παραπεμπτικού	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Όνομασία Νοσοκομείου	Αιτία Παραπομπής	1	1202016	01014025894	ΟΓΑ	Υγεία	Μυωπία.	1	4377946	01014025894	ΟΓΑ	Υγεία	Εκφύλιση ωχράς κηλίδας.	1	6124477	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Κερατόκωνος.	1	6427958	14058321543	ΙΚΑ	Υγεία	Πίεση.	1	7627958	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Γλαύκωμα.
ID Ιατρού		ID Παραπεμπτικού	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Όνομασία Νοσοκομείου	Αιτία Παραπομπής																																
1	1202016	01014025894	ΟΓΑ	Υγεία	Μυωπία.																																	
1	4377946	01014025894	ΟΓΑ	Υγεία	Εκφύλιση ωχράς κηλίδας.																																	
1	6124477	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Κερατόκωνος.																																	
1	6427958	14058321543	ΙΚΑ	Υγεία	Πίεση.																																	
1	7627958	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Γλαύκωμα.																																	
	<input type="button" value="Εντάξει"/>																																					

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 77: Το παράθυρο της αναζήτησης του ιστορικού ενός ιατρού από τον ίδιο τον ιατρό, ο οποίος έχει συντάξει τουλάχιστον ένα παραπεμπτικό ασθενούς

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων														
<table border="1"> <thead> <tr> <th>Κατηγορίες</th> </tr> </thead> <tbody> <tr><td>Σύνταξη Παραπεμπτικού</td></tr> <tr><td>Στοιχεία Παραπεμπτικού</td></tr> <tr><td>Ιστορικό Ασθενούς</td></tr> <tr><td>Ιστορικό Ιατρού</td></tr> </tbody> </table>	Κατηγορίες	Σύνταξη Παραπεμπτικού	Στοιχεία Παραπεμπτικού	Ιστορικό Ασθενούς	Ιστορικό Ιατρού	<table border="1"> <thead> <tr> <th>Ιστορικό Ιατρού</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"> <p>Δεν έχετε συντάξει κανένα παραπεμπτικό!!!</p> <p><input type="button" value="Εντάξει"/></p> </td> </tr> </tbody> </table>	Ιστορικό Ιατρού	<p>Δεν έχετε συντάξει κανένα παραπεμπτικό!!!</p> <p><input type="button" value="Εντάξει"/></p>	<table border="1"> <thead> <tr> <th>Χρήστης</th> </tr> </thead> <tbody> <tr> <td>Κωδικός Ιατρού: 4</td> </tr> <tr> <td>Όνοματεπώνυμο Ιατρού: Γεωργία Κυριακού</td> </tr> <tr> <td>Ειδικότητα Ιατρού: Ειδικός Οφθαλμίατρος</td> </tr> <tr> <td style="text-align: center;"><input type="button" value="Αποσύνδεση"/></td> </tr> </tbody> </table>	Χρήστης	Κωδικός Ιατρού: 4	Όνοματεπώνυμο Ιατρού: Γεωργία Κυριακού	Ειδικότητα Ιατρού: Ειδικός Οφθαλμίατρος	<input type="button" value="Αποσύνδεση"/>
Κατηγορίες														
Σύνταξη Παραπεμπτικού														
Στοιχεία Παραπεμπτικού														
Ιστορικό Ασθενούς														
Ιστορικό Ιατρού														
Ιστορικό Ιατρού														
<p>Δεν έχετε συντάξει κανένα παραπεμπτικό!!!</p> <p><input type="button" value="Εντάξει"/></p>														
Χρήστης														
Κωδικός Ιατρού: 4														
Όνοματεπώνυμο Ιατρού: Γεωργία Κυριακού														
Ειδικότητα Ιατρού: Ειδικός Οφθαλμίατρος														
<input type="button" value="Αποσύνδεση"/>														
<p>ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ</p> <p style="text-align: right;">ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΑΣ (ΑΜ 679)</p>														

Εικόνα 78: Το παράθυρο της αναζήτησης του ιστορικού ενός ιατρού από τον ίδιο τον ιατρό, ο οποίος δεν έχει συντάξει ούτε ένα παραπεμπτικό ασθενούς

Το μεν πρώτο από τα δύο πιο πάνω παράθυρα απεικονίζει τα παραπεμπτικά ασθενών τα οποία έχει συντάξει ο εκάστοτε ιατρός, ο οποίος αποτελεί τον τρέχων χρήστη του προγράμματος που προσδιορίζεται στο παρόν έργο. Το δε δεύτερο από τα δύο πιο άνω παράθυρα πληροφορεί τον προαναφερόμενο ιατρό πως δεν προχωρήσει στη σύνταξη κανενός παραπεμπτικού ασθενούς. Ακόμα, και τα δύο πιο άνω παράθυρα παρέχουν στον προαναφερόμενο ιατρό την ευχέρεια να πλοηγηθεί στο αρχικό παράθυρο ιατρού αυτού του λογισμικού, πατώντας στο κουμπί «Εντάξει» που βρίσκεται και στα δύο ανώτερα παράθυρα.

3.4.3 Η Είσοδος της Γραμματείας ενός Νοσοκομείου στο Σύστημα Λογισμικού της Τρέχουσας Πτυχιακής Διατριβής

Εκτός από τον εκάστοτε διαχειριστή του συστήματος λογισμικού που καθορίζεται στην τρέχουσα πτυχιακή διατριβή και τον εκάστοτε συνεργαζόμενο με έναν ή περισσότερους Ασφαλιστικούς Φορείς ενός κράτους ιατρός οι οποίοι μπορούν να πραγματοποιούν είσοδο σε αυτό το σύστημα, θα δύναται να εισέρθει σε αυτό το λογισμικό και η γραμματεία του εκάστοτε νοσοκομείου-οφθαλμολογικής κλινικής, είτε κρατικού είτε ιδιωτικού, που είναι συμβεβλημένο με ένα ή περισσότερα Ασφαλιστικά Ταμεία μίας χώρας. Όπως ο εκάστοτε διαχειριστής και ο εκάστοτε ιατρός, έτσι και η γραμματεία του εκάστοτε νοσοκομείου, η οποία θα θέλει να μπει στο συγκεκριμένο σύστημα, θα πρέπει να καταχωρήσει το προσωπικό της όνομα χρήστη και τον προσωπικό της κωδικό στα αντίστοιχα πεδία της φόρμα εισόδου νοσοκομείου του αρχικού παραθύρου του εν λόγω λογισμικού. Έπειτα, θα ακολουθήσει επαλήθευση των προαναφερόμενων στοιχείων με βάση τις τιμές των αντίστοιχων πεδίων οι οποίες υπάρχουν στη

βάση δεδομένων με την οποία είναι συνδεδεμένο το παραπάνω σύστημα. Άμα η προγενέστερη επαλήθευση αποτύχει, τότε θα αναδυθεί το ίδιο ακριβώς παράθυρο προειδοποίησης το οποίο αναδύεται, όταν δεν ταυτοποιούνται τα στοιχεία τόσο του παραπάνω διαχειριστή όσο και του ανώτερου ιατρού. Αν, όμως, η πρωτύτερη επαλήθευση επιτύχει, τότε θα εμφανιστεί το παράθυρο το οποίο αντικατοπτρίζεται στη μετέπειτα Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων											
<table border="1"> <thead> <tr> <th>Κατηγορίες</th> </tr> </thead> <tbody> <tr><td>Ορισμός Ραντεβού</td></tr> <tr><td>Αποτέλεσμα Εξέτασης</td></tr> <tr><td>Ιστορικό Ασθενούς</td></tr> <tr><td>Ιστορικό Νοσοκομείου</td></tr> </tbody> </table>	Κατηγορίες	Ορισμός Ραντεβού	Αποτέλεσμα Εξέτασης	Ιστορικό Ασθενούς	Ιστορικό Νοσοκομείου	<p>Αρχικό Παράθυρο Νοσοκομείου</p> <p>Παρακαλούμε επιλέξτε μία από τις κατηγορίες...!!!</p>	<table border="1"> <thead> <tr> <th>Χρήστης</th> </tr> </thead> <tbody> <tr><td>Κωδικός Νοσοκομείου: 10000</td></tr> <tr><td>Όνομασία Νοσοκομείου: Υγεία</td></tr> <tr><td>Αποσύνδεση</td></tr> </tbody> </table>	Χρήστης	Κωδικός Νοσοκομείου: 10000	Όνομασία Νοσοκομείου: Υγεία	Αποσύνδεση
Κατηγορίες											
Ορισμός Ραντεβού											
Αποτέλεσμα Εξέτασης											
Ιστορικό Ασθενούς											
Ιστορικό Νοσοκομείου											
Χρήστης											
Κωδικός Νοσοκομείου: 10000											
Όνομασία Νοσοκομείου: Υγεία											
Αποσύνδεση											
<p>ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)</p>											

Εικόνα 79: Το αρχικό παράθυρο ενός νοσοκομείου

Το παράθυρο που δείχνεται στην πιο πάνω Εικόνα συνιστά το αρχικό παράθυρο το οποίο αντικρίζει η γραμματεία ενός νοσοκομείου, μόλις μπει στη διαδικτυακή εφαρμογή η οποία μελετάται στο παρόν έργο. Αυτό το παράθυρο, τώρα, καθιστά ικανή την προαναφερόμενη γραμματεία να προβαίνει κάθε φορά σε μία ακολουθία από ενέργειες, οι οποίες είναι οι παρακάτω:

- Να ορίζει ραντεβού για τα παραπεμπτικά ασθενών για τα οποία δεν έχει καθοριστεί ημερομηνία και ώρα εξέτασης.
- Να καταχωρεί στον ηλεκτρονικό φάκελο του εκάστοτε ασθενούς το αποτέλεσμα της κατάλληλης σε κάθε περίπτωση εξέτασης η οποία έλαβε χώρα στο εν λόγω νοσοκομείο.
- Να αναζητά το ιστορικό της οφθαλμιατρικής υγείας ενός ασθενούς.
- Να ψάχνει το ιστορικό όλων των εξετάσεων οι οποίες έχουν διενεργηθεί ή πρόκειται να διενεργηθούν στο συγκεκριμένο νοσοκομείο.

Εφόσον, τώρα, η γραμματεία του εκάστοτε νοσοκομείου, η οποία εισέρχεται στην εφαρμογή η οποία υλοποιείται στην παρούσα εργασία, θα πρέπει να ορίσει ραντεβού για ένα παραπεμπτικό ασθενούς, θα χρειαστεί να «κλικάρει» στην κατηγορία «Ορισμός Ραντεβού» του αρχικού παραθύρου νοσοκομείου της εν λόγω εφαρμογής. Τότε θα ξεπροβάλει το παράθυρο το οποίο απεικονίζεται στη μετέπειτα Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων													
<table border="1"> <thead> <tr> <th>Κατηγορίες</th> </tr> </thead> <tbody> <tr><td>Ορισμός Ραντεβού</td></tr> <tr><td>Αποτέλεσμα Εξέτασης</td></tr> <tr><td>Ιστορικό Ασθενούς</td></tr> <tr><td>Ιστορικό Νοσοκομείου</td></tr> </tbody> </table>	Κατηγορίες	Ορισμός Ραντεβού	Αποτέλεσμα Εξέτασης	Ιστορικό Ασθενούς	Ιστορικό Νοσοκομείου	<table border="1"> <thead> <tr> <th>Ορισμός Ραντεβού</th> </tr> </thead> <tbody> <tr> <td> <p>Προκειμένου να αναζητήσετε τα εκκρεμή παραπεμπτικά ως προς τον ορισμό ραντεβού του νοσοκομείου σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;"><input type="button" value="Αναζήτηση"/></p> </td> </tr> </tbody> </table>	Ορισμός Ραντεβού	<p>Προκειμένου να αναζητήσετε τα εκκρεμή παραπεμπτικά ως προς τον ορισμό ραντεβού του νοσοκομείου σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;"><input type="button" value="Αναζήτηση"/></p>	<table border="1"> <thead> <tr> <th>Χρήστης</th> </tr> </thead> <tbody> <tr> <td>Κωδικός Νοσοκομείου: 10000</td> </tr> <tr> <td>Όνομασία Νοσοκομείου: Υγεία</td> </tr> <tr> <td><input type="button" value="Αποσύνδεση"/></td> </tr> </tbody> </table>	Χρήστης	Κωδικός Νοσοκομείου: 10000	Όνομασία Νοσοκομείου: Υγεία	<input type="button" value="Αποσύνδεση"/>
Κατηγορίες													
Ορισμός Ραντεβού													
Αποτέλεσμα Εξέτασης													
Ιστορικό Ασθενούς													
Ιστορικό Νοσοκομείου													
Ορισμός Ραντεβού													
<p>Προκειμένου να αναζητήσετε τα εκκρεμή παραπεμπτικά ως προς τον ορισμό ραντεβού του νοσοκομείου σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;"><input type="button" value="Αναζήτηση"/></p>													
Χρήστης													
Κωδικός Νοσοκομείου: 10000													
Όνομασία Νοσοκομείου: Υγεία													
<input type="button" value="Αποσύνδεση"/>													
<p>ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ</p> <p style="text-align: right;">ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΑΣ (ΑΜ 679)</p>													

Εικόνα 80: Το παράθυρο της αναζήτησης των εκκρεμών παραπεμπτικών ασθενών ως προς τον ορισμό ραντεβού από ένα νοσοκομείο

Το παράθυρο που απεικονίζεται στην πιο πάνω Εικόνα καθιστά εφικτό στη γραμματεία του εκάστοτε νοσοκομείου, η οποία συνδέεται στο λογισμικό το οποίο μελετάται στην τρέχουσα αναφορά, να αναζητήσει όλα παραπεμπτικά ασθενών για τα οποία δεν έχει προσδιοριστεί ημερομηνία και ώρα εξέτασης, πιέζοντας το κουμπί «Αναζήτηση» του παραπάνω παραθύρου. Τότε, θα παρουσιαστεί, το παράθυρο που απεικονίζεται στην πιο κάτω Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ορισμός Ραντεβού	Χρήστης																														
Ορισμός Ραντεβού	<p>Τα παραπεμπτικά σε αναμονή ορισμού ραντεβού του νοσοκομείου σας φαίνονται στον ακόλουθο πίνακα:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>ID Παραπεμπτικού</th> <th>ΑΜΚΑ Ασθενός</th> <th>Ασφάλεια Ασθενός</th> <th>Αιτία Παραπομπής</th> <th>ID Νοσοκομείου</th> <th>Ημερομηνία Εξέτασης</th> </tr> </thead> <tbody> <tr> <td>4377946</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Εκφύλιση οφθ. κηλίδας.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> <tr> <td>6427958</td> <td>14058321543</td> <td>ΙΚΑ</td> <td>Πίεση.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> <tr> <td>7627958</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Γλαύκωμα.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> <tr> <td>8827958</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Εξασθένηση οπτικών νεύρων.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> </tbody> </table> <p>Προκειμένου να ορίσετε ραντεβού για ένα από τα εκκρεμή παραπεμπτικά του παραπάνω πίνακα, παρακάλλουμε πληκτρολογήσετε το ID του στο αμέσως επόμενο πλαίσιο κειμένου:</p> <div style="text-align: center; margin: 10px 0;"> <input style="width: 100px; height: 20px;" type="text"/> </div> <p>και κατόπιν πατήστε στο κουμπί "Επιλογή":</p> <div style="text-align: center; margin-top: 10px;"> <input style="background-color: #0070c0; color: white; padding: 5px 15px;" type="button" value="Επιλογή"/> </div>	ID Παραπεμπτικού	ΑΜΚΑ Ασθενός	Ασφάλεια Ασθενός	Αιτία Παραπομπής	ID Νοσοκομείου	Ημερομηνία Εξέτασης	4377946	01014025894	ΟΓΑ	Εκφύλιση οφθ. κηλίδας.	10000	Δεν έχει οριστεί!!!	6427958	14058321543	ΙΚΑ	Πίεση.	10000	Δεν έχει οριστεί!!!	7627958	18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	10000	Δεν έχει οριστεί!!!	8827958	18759641223	ΔΗΜΟΣΙΟ	Εξασθένηση οπτικών νεύρων.	10000	Δεν έχει οριστεί!!!	Κωδικός Νοσοκομείου: 10000
ID Παραπεμπτικού		ΑΜΚΑ Ασθενός	Ασφάλεια Ασθενός	Αιτία Παραπομπής	ID Νοσοκομείου	Ημερομηνία Εξέτασης																										
4377946		01014025894	ΟΓΑ	Εκφύλιση οφθ. κηλίδας.	10000	Δεν έχει οριστεί!!!																										
6427958		14058321543	ΙΚΑ	Πίεση.	10000	Δεν έχει οριστεί!!!																										
7627958		18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	10000	Δεν έχει οριστεί!!!																										
8827958	18759641223	ΔΗΜΟΣΙΟ	Εξασθένηση οπτικών νεύρων.	10000	Δεν έχει οριστεί!!!																											
Αποτέλεσμα Εξέτασης		Όνομασία Νοσοκομείου: Υγεία																														
Ιστορικό Ασθενός		<input style="background-color: #0070c0; color: white; padding: 2px 5px;" type="button" value="Αποσύνδεση"/>																														
Ιστορικό Νοσοκομείου																																

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 81: Το παράθυρο της επιλογής των εκκρεμών παραπεμπτικών ασθενών ως προς τον ορισμό ραντεβού από ένα νοσοκομείο

Το παράθυρο που δείχνεται στην πιο πάνω Εικόνα καθιστά δυνατό στη γραμματοειδή του εκάστοτε νοσοκομείου, η οποία αλληλεπιδρά με την εφαρμογή η οποία μελετάται στην τρέχουσα αναφορά, να επιλέξει ένα από τα εν παραπεμπτικά ασθενών που βρίσκονται σε αναμονή ορισμού ραντεβού, εισάγοντας στο πεδίο κειμένου που υφίσταται στο ανώτερο παράθυρο το ID του προαναφερόμενου παραπεμπτικού και πατώντας μετά στο κουμπί «Επιλογή» στο κάτω μέρος του ανώτερου παραθύρου, προκειμένου να καταχωρήσει στο εν λόγω παραπεμπτικό ημερομηνία και ώρα εξέτασης.

Παρόλα αυτά, αν η γραμματοειδή του εκάστοτε νοσοκομείου, η οποία παίζει το ρόλο του τρέχοντος χρήστη της εφαρμογής που αναπτύσσεται στην παρούσα διπλωματική, γράφει στο πεδίο κειμένου το οποίο περιέχεται στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ασθενών ως προς τον ορισμό ραντεβού της προαναφερόμενης εφαρμογής ένα ID παραπεμπτικού ασθενούς, το οποίο δεν υπάρχει στην ανάλογη στήλη του πίνακα εμφάνισης των σχετικών παραπεμπτικών ασθενών, ο οποίος, επίσης, είναι τοποθετημένος στο πάνω μέρος του πρωτύπου παραθύρου, τότε, πατώντας η εν λόγω γραμματοειδή το κουμπί «Επιλογή» στην κάτω μεριά του συγκεκριμένου παραθύρου, θα προβάλει το παράθυρο προειδοποίησης της κατοπινης Εικόνας:

Το ID που πληκτρολογήσατε δεν αντιστοιχεί σε εκκρεμές παραπεμπτικό ως προς τον ορισμό ραντεβού του νοσοκομείου σας!!!

Εικόνα 82: Το παράθυρο προειδοποίησης της ανεπιτυχούς επιλογής εκκρεμούς παραπεμπτικού ασθενούς ως προς τον ορισμό ραντεβού από ένα νοσοκομείο

Το παράθυρο που δείχνεται στην ανώτερη Εικόνα προειδοποιεί τη γραμματεία του εκάστοτε του νοσοκομείου, η οποία χειρίζεται το πρόγραμμα το οποίο περιγράφεται στην παρούσα αναφορά, πως το ID παραπεμπτικού ασθενούς, το οποίο κατέγραψε στο πεδίο κειμένου το οποίο βρίσκεται στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ασθενών ως προς τον ορισμό ραντεβού του εν λόγω προγράμματος, δεν είναι εγγεγραμμένο στην ανάλογη στήλη του πίνακα εμφάνισης των σχετικών παραπεμπτικών ασθενών, ο οποίος απαντάται στο άνω μέρος του προηγούμενου παράθυρο. Επίσης, προσφέρει την ευκαιρία στην ανώτερη γραμματεία να καταχωρήσει στο προγενέστερο πεδίο κειμένου ένα ID παραπεμπτικού ασθενούς, το οποίο είναι γραμμένο στο αντίστοιχο πεδίο του προγενέστερου πίνακα, κάνοντας κλικ στο κουμπί «Δοκιμάστε ξανά» του πιο πάνω παραθύρου.

Εφόσον, βέβαια, η γραμματεία του εκάστοτε νοσοκομείου, η οποία έχει μπει στο σύστημα το οποίο αναπτύσσεται στην τρέχουσα διπλωματική, γράψει στο πεδίο κειμένου το οποίο υπάρχει στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ασθενών ως προς τον ορισμό ραντεβού του εν λόγω συστήματος ένα ID παραπεμπτικού ασθενούς, το οποίο είναι καταχωρημένο στην αντίστοιχη στήλη του πίνακα εμφάνισης των σχετικών παραπεμπτικών ασθενών, ο οποίος βρίσκεται στην άνω περιοχή αυτού του παραθύρου, τότε, πατώντας η παραπάνω γραμματεία στο κουμπί «Επιλογή» του προηγούμενου παραθύρου, θα προκύψει παράθυρο που φαίνεται στη μεταγενέστερη Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ορισμός Ραντεβού	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Προκειμένου να ορίσετε ραντεβού για το εκκρεμές παραπεμπτικό με ID 4377946 του νοσοκομείου σας, παρακαλούμε συμπληρώστε τόσο την ημερομηνία όσο και την ώρα εξέτασης στα πεδία "Ημερομηνία Εξέτασης" και "Ωρα Εξέτασης", αντίστοιχα, που ακολουθούν και κατόπιν πατήστε στο κουμπί "Υποβολή" στο τέλος της τρέχουσας φόρμας:</p> <p>ID Παραπεμπτικού: <input type="text" value="4377946"/></p> <p>ID Ιατρού: <input type="text" value="1"/></p> <p>ΑΜΚΑ Ασθενούς: <input type="text" value="01014025894"/></p> <p>Ασφάλεια Ασθενούς: <input type="text" value="ΟΓΑ"/></p> <p>Αιτία Παραπομπής: <input style="width: 100%;" type="text" value="Εκφύλιση ωχράς κηλίδας."/></p> <p>ID Νοσοκομείου: <input type="text" value="10000"/></p> <p>Ημερομηνία Εξέτασης: <input type="text"/></p> <p>Ωρα Εξέτασης: <input type="text"/></p> <p style="text-align: right;"><input type="button" value="Υποβολή"/></p>	<p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p style="text-align: center;"><input type="button" value="Αποσύνδεση"/></p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 83: Το παράθυρο ορισμού ραντεβού για έναν ασθενή από ένα νοσοκομείο

Το παράθυρο που αποδίδεται στην πιο πάνω Εικόνα παρέχει την ικανότητα στη γραμματεία του εκάστοτε νοσοκομείου, η οποία λειτουργεί το πρόγραμμα το οποίο υλοποιείται στο παρόν έργο, να καθορίσει ημερομηνία και ώρα εξέτασης για το παραπεμπτικό ασθενούς το οποίο επιλέχθηκε από την εν λόγω γραμματεία στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ως προς τον ορισμό ραντεβού. Για να πραγματοποιήσει κάτι τέτοιο η εν λόγω γραμματεία θα πρέπει να συμπληρώνει με κατάλληλο τρόπο τα πεδία «Ημερομηνία Εξέτασης» και «Ωρα Εξέτασης» της φόρμας ορισμού ραντεβού του συγκεκριμένου προγράμματος (δε χρειάζεται να συμπληρώσει τα υπόλοιπα πεδία της αυτής φόρμας, διότι τα εν λόγω

πεδία έχουν ήδη συμπληρωθεί με αυτόματο τρόπο, με το που θα εμφανιστεί το ανώτερο παράθυρο) και να υποβάλει έπειτα τη συγκεκριμένη φόρμα στο διακομιστή Ιστού του τρέχοντος προγράμματος, πιέζοντας το κουμπί «Υποβολή» της ανώτερης εικόνας.

Ωστόσο, αν η γραμματεία του εκάστοτε νοσοκομείου, η οποία είναι ο τρέχων χρήστης της εφαρμογής που αναπτύσσεται στο τρέχον έγγραφο, εισάγει στα πεδία «Ημερομηνία Εξέτασης» και «Ωρα Εξέτασης» της φόρμας ορισμού ραντεβού του αυτής της εφαρμογής μία ημερομηνία και ώρα εξέτασης αντίστοιχα, οι οποίες είναι ήδη καταχωρημένες στα αντίστοιχα πεδία της βάσης δεδομένων με την οποία επικοινωνεί η εν λόγω εφαρμογή, τότε, πατώντας η παραπάνω γραμματεία στο κουμπί «Αναζήτηση» του πρωτύπου παραθύρου, θα εμφανιστεί το παράθυρο της κατώτερης Εικόνας:



Εικόνα 84: Το παράθυρο προειδοποίησης του ανεπιτυχούς ορισμού ραντεβού για έναν ασθενή από ένα νοσοκομείο

Το παράθυρο που φαίνεται στην ανώτερη Εικόνα ειδοποιεί τη γραμματεία του εκάστοτε νοσοκομείου, η οποία αλληλεπιδρά με το λογισμικό το οποίο είναι το αντικείμενο ανάλυσης της παρούσας διπλωματικής, ότι οι τιμές που κατέγραψε στα πεδία «Ημερομηνία Εξέτασης» και «Ωρα Εξέτασης» της φόρμας ορισμού ραντεβού του ομώνυμου παραθύρου του εν λόγω λογισμικού είναι ήδη καταχωρημένες στα αντίστοιχα πεδία της βάσης δεδομένων με την οποία διασυνδέεται το εν λόγω λογισμικό. Ακόμα, δίνει την ευχέρεια στην τρέχουσα γραμματεία να γράψει τιμές στα προαναφερόμενα πεδία οι οποίες δεν προϋπάρχουν στην προηγούμενη βάση, με το πάτημα του κουμπιού «Δοκιμάστε ξανά» του πιο πάνω παραθύρου.

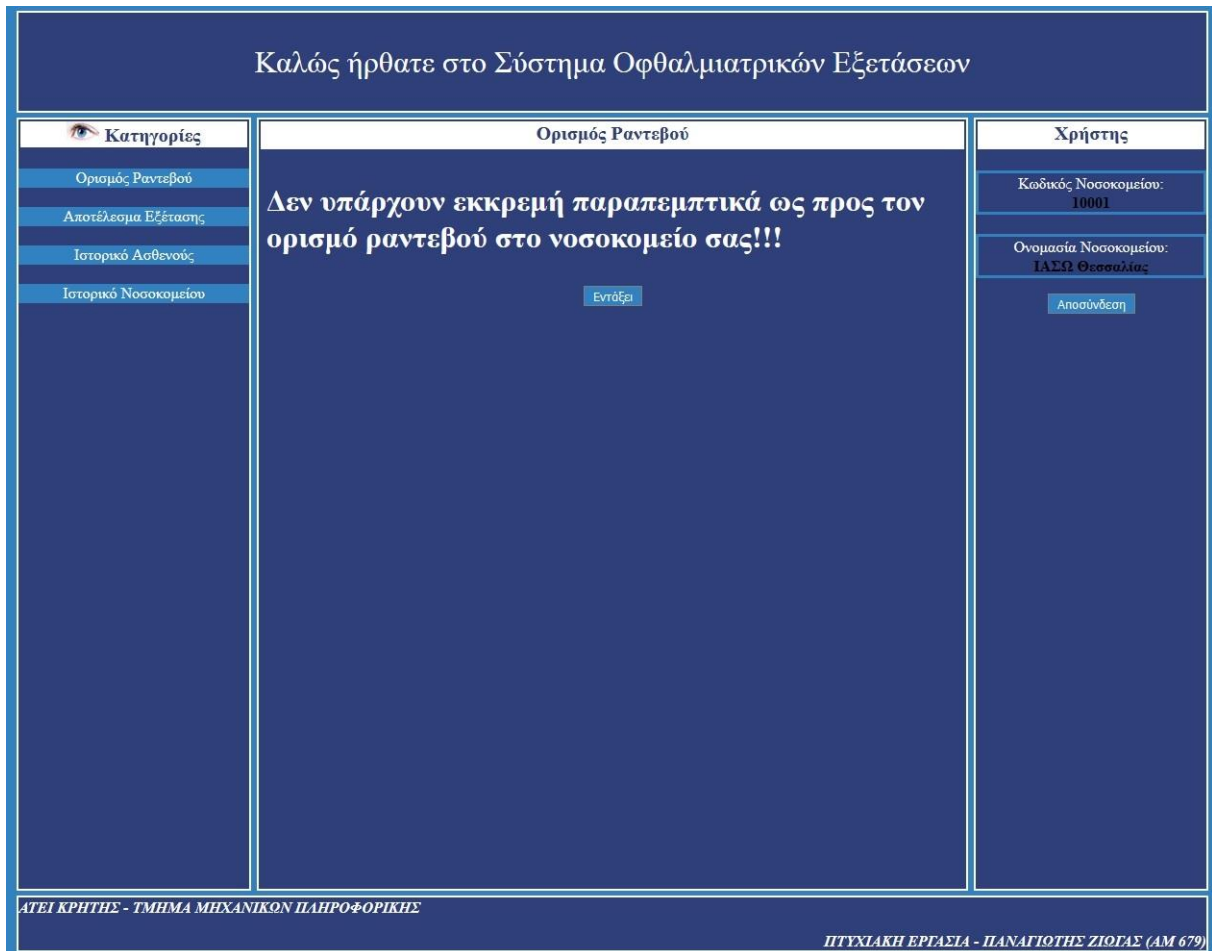
Εφόσον, φυσικά, η γραμματεία του εκάστοτε νοσοκομείου, η οποία σε αλληλεπιδρά με το σύστημα το οποίο προδιαγράφεται στο παρόν έργο, συμπληρώσει με έγκυρο τρόπο τα πεδία «Ημερομηνία Εξέτασης» και «Ωρα Εξέτασης» της φόρμας ορισμού ραντεβού του ομώνυμου παραθύρου αυτού του συστήματος, τότε, πιέζοντας ο πιο πάνω ιατρός το κουμπί «Υποβολή» της συγκεκριμένης φόρμας, θα ξεπροβάλει το παράθυρο επιβεβαίωσης το οποίο φαίνεται στην πιο κάτω Εικόνα:



Εικόνα 85: Το παράθυρο επιβεβαίωσης του επιτυχούς ορισμού ραντεβού για έναν ασθενή από ένα νοσοκομείο

Το παράθυρο που αναπαρίσταται στην ανώτερη Εικόνα πληροφορεί τη γραμματεία του εκάστοτε νοσοκομείου, η οποία αποτελεί τον τρέχων λειτουργό του συστήματος λογισμικού το οποίο οριοθετείται στο τρέχον έργο, ότι συμπλήρωσε έγκυρα τα πεδία τα πεδία «Ημερομηνία Εξέτασης» και «Ωρα Εξέτασης» της φόρμας ορισμού ραντεβού αυτού του συστήματος και συνεπώς όρισε με επιτυχία ένα ραντεβού για το παραπεμπτικό ασθενούς το οποίο είχε επιλεγεί στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ως προς τον ορισμό ραντεβού. Επιπλέον, παρέχει στην εν λόγω γραμματεία τη δυνατότητα να μεταφερθεί στο αρχικό παράθυρο ιατρού στη συγκεκριμένη εφαρμογή, πατώντας το κουμπί «Εντάξει» του ανώτερου παραθύρου.

Στο σημείο τούτο, να παρατηρηθεί πως, εφόσον δεν έχουν σταλεί καθόλου παραπεμπτικά ασθενών σε ένα νοσοκομείο ή η γραμματεία αυτού του νοσοκομείου έχει ορίσει ραντεβού για όλα τα παραπεμπτικά ασθενών που έχουν σταλεί στον εν λόγω νοσοκομείο, τότε, αν η προαναφερόμενη γραμματεία επιλέξει την κατηγορία «Ορισμός Ραντεβού» στο αρχικό παράθυρο νοσοκομείου της εφαρμογής η οποία εξετάζεται στο τρέχον έργο, θα προκύψει το παράθυρο που αντικατοπτρίζεται στην κατοπινή Εικόνα:



Εικόνα 86: Το παράθυρο πληροφόρησης για τη μη ύπαρξη εκκρεμών παραπεμπτικών ως προς τον ορισμό ραντεβού σε ένα νοσοκομείο

Το παράθυρο που δείχνεται στην πιο άνω Εικόνα ενημερώνει τη γραμματεία του εκάστοτε νοσοκομείου, η οποία συνιστά τον τρέχων χρήστη του συστήματος λογισμικού το οποίο περιγράφεται στην τρέχουσα πτυχιακή, ότι δεν παραπεμπτικά σε αναμονή καθορισμού ραντεβού για το εν λόγω νοσοκομείο, είτε γιατί δεν έχει αποσταλεί κανένα παραπεμπτικό ασθενούς σε αυτό το νοσοκομείο είτε γιατί η γραμματεία του έχει καταχωρήσει ημερομηνία και ώρα εξέτασης σε όσα παραπεμπτικά ασθενών έχουν αποσταλεί σε αυτό. Επιπλέον, δίνει στην προηγούμενη γραμματεία την ευκαιρία να μεταφερθεί στο αρχικό παράθυρο νοσοκομείου του τρέχοντος λογισμικού, πατώντας το κουμπί «Εντάξει» του ανώτερου παραθύρου.

Καθώς, τώρα, η γραμματεία του εκάστοτε νοσοκομείου, η οποία μπαίνει στην εφαρμογή την οποία πραγματεύεται η παρούσα αναφορά, θα χρειαστεί να καταχωρήσει στον ηλεκτρονικό φάκελο ενός ασθενούς το αποτέλεσμα της κατάλληλης σε κάθε περίπτωση, σε σχέση με τον εν λόγω ασθενή, εξέτασης που έλαβε χώρα στο προαναφερόμενο νοσοκομείο, θα πρέπει να επιλέξει την κατηγορία «Αποτέλεσμα Εξέτασης» του αρχικού παραθύρου νοσοκομείου της παρούσας εφαρμογής. Τότε θα προβάλει το παράθυρο το οποίο δείχνεται στη μετέπειτα Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων		
<p> Κατηγορίες</p> <ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Αποτέλεσμα Εξέτασης</p> <p>Προκειμένου να αναζητήσετε τα εκκρεμή παραπεμπτικά ως προς την καταχώρηση αποτελέσματος εξέτασης του νοσοκομείου σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;">Αναζήτηση</p>	<p>Χρήστης</p> <p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p>Απουύδωση</p>
<p>ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΙΑΣ (ΑΜ 679)</p>		

Εικόνα 87: Το παράθυρο της αναζήτησης των εκκρεμών παραπεμπτικών ασθενών ως προς την καταχώρηση αποτελέσματος εξέτασης από ένα νοσοκομείο

Το παράθυρο που αποδίδεται στην παραπάνω Εικόνα καθιστά εφικτό στη γραμματεία του εκάστοτε νοσοκομείου, η οποία συνδέεται στο σύστημα το οποίο αναλύεται στο παρόν έργο, να ψάξει όλα τα εκκρεμή προς την καταχώρηση αποτελέσματος εξέτασης παραπεμπτικά ασθενών πατώντας το κουμπί «Αναζήτηση» του παραπάνω παραθύρου. Τότε, θα φανεί, το παράθυρο που δείχνεται στην κατοπινή Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αποτέλεσμα Εξέτασης	Χρήστης																		
Ορισμός Ραντεβού	<p>Τα παραπεμπτικά του νοσοκομείου σας για τα οποία έχει οριστεί ραντεβού, έχει εξεταστεί ο κατάλληλος κάθε φορά ασθενής, αλλά δεν έχει καταχωρηθεί στον ηλεκτρονικό φάκελο του ασθενούς το αποτέλεσμα της εξέτασής του, φαίνονται στον ακόλουθο πίνακα:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>ID Παραπεμπτικό</th> <th>ΑΜΚΑ Ασθενούς</th> <th>Ασφάλεια Ασθενούς</th> <th>ID Νοσοκομείου</th> <th>Ημερομηνία Εξέτασης</th> <th>Αποτέλεσμα Εξέτασης</th> </tr> </thead> <tbody> <tr> <td>4377946</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>10000</td> <td>10/05/2016</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> <tr> <td>6124477</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>10000</td> <td>12/06/2016</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> </tbody> </table> <p>Προκειμένου να καταχωρήσετε το αποτέλεσμα της εξέτασης για ένα από τα εκκρεμή παραπεμπτικά του παραπάνω πίνακα, παρακάλούμε πληκτρολογήσετε το ID του στο αμέσως επόμενο πλαίσιο κειμένου:</p> <div style="text-align: center; margin: 10px 0;"> <input style="width: 100px; height: 20px;" type="text"/> </div> <p>και κατόπιν πατήστε στο κουμπί "Επιλογή":</p> <div style="text-align: center; margin-top: 10px;"> <input style="background-color: #003366; color: white; padding: 2px 10px;" type="button" value="Επιλογή"/> </div>	ID Παραπεμπτικό	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	ID Νοσοκομείου	Ημερομηνία Εξέτασης	Αποτέλεσμα Εξέτασης	4377946	01014025894	ΟΓΑ	10000	10/05/2016	Δεν έχει καταχωρηθεί!!!	6124477	18759641223	ΔΗΜΟΣΙΟ	10000	12/06/2016	Δεν έχει καταχωρηθεί!!!	Κωδικός Νοσοκομείου: 10000
ID Παραπεμπτικό		ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	ID Νοσοκομείου	Ημερομηνία Εξέτασης	Αποτέλεσμα Εξέτασης														
4377946		01014025894	ΟΓΑ	10000	10/05/2016	Δεν έχει καταχωρηθεί!!!														
6124477		18759641223	ΔΗΜΟΣΙΟ	10000	12/06/2016	Δεν έχει καταχωρηθεί!!!														
Αποτέλεσμα Εξέτασης		Ονομασία Νοσοκομείου: Υγεία																		
Ιστορικό Ασθενούς	<input style="background-color: #003366; color: white; padding: 2px 10px;" type="button" value="Αποσύνδεση"/>																			
Ιστορικό Νοσοκομείου																				

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΙΑΣ (ΑΜ 679)

Εικόνα 88: Το παράθυρο της επιλογής των εκκρεμών παραπεμπτικών ασθενών ως προς την καταχώρηση αποτελέσματος εξέτασης από ένα νοσοκομείο

Το παράθυρο που απεικονίζεται στην πιο άνω Εικόνα δίνει την ευχέρεια στη γραμματεία του εκάστοτε νοσοκομείου, η οποία συνδέεται στην εφαρμογή η οποία ορίζεται στην τρέχουσα διατριβή, να διαλέξει ένα από τα εκκρεμή ως προς την καταχώρηση αποτελέσματος εξέτασης παραπεμπτικά ασθενών, καταχωρώντας στο πεδίο κειμένου που υπάρχει στο πιο πάνω παράθυρο το ID του προαναφερόμενου παραπεμπτικού και κάνοντας κλικ έπειτα στο κουμπί «Επιλογή» στην κάτω περιοχή του παραπάνω παραθύρου, προκειμένου να γράψει σε αυτό το παραπεμπτικό ασθενούς το κατάλληλο σε κάθε φάση αποτέλεσμα εξέτασης.

Εντούτοις, αν η γραμματεία του εκάστοτε νοσοκομείου, η οποία συνιστά τον τρέχον χρήστη του λογισμικού που περιγράφεται στην παρούσα διπλωματική, εισάγει στο πεδίο κειμένου το οποίο υφίσταται στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ασθενών ως προς την καταχώρηση αποτελέσματος εξέτασης ένα ID παραπεμπτικού ασθενούς, το οποίο δεν είναι καταχωρημένο στην αντίστοιχη στήλη του πίνακα παρουσίασης των σχετικών παραπεμπτικών, ο οποίος, επίσης, υπάρχει στο πάνω μέρος του εν λόγω παραθύρου, τότε, πατώντας η εν ανώτερη γραμματεία στο κουμπί «Επιλογή» του προηγούμενου παραθύρου, θα αναδυθεί το παράθυρο προειδοποίησης το οποίο δίδεται στην ύστερη Εικόνα:

Το ID που πληκτρολογήσατε δεν αντιστοιχεί σε εκκρεμές παραπεμπτικό ως προς την καταχώρηση αποτελέσματος εξέτασης του νοσοκομείου σας!!!

Δοκιμάστε ξανά

Εικόνα 89: Το παράθυρο προειδοποίησης της ανεπιτυχούς επιλογής εκκρεμούς παραπεμπτικού ασθενούς ως προς την καταχώρηση αποτελέσματος εξέτασης από ένα νοσοκομείο

Το παράθυρο που καθρεπτίζεται στην ανώτερη Εικόνα ενημερώνει τη γραμματεία του εκάστοτε του νοσοκομείου, η οποία λειτουργεί το πρόγραμμα το οποίο παρουσιάζεται στην παρούσα πτυχιακή, πως το ID παραπεμπτικού ασθενούς, το οποίο εισήγαγε στο πεδίο κειμένου το οποίο υπάρχει στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ασθενών ως προς την καταχώρηση αποτελέσματος εξέτασης του εν λόγω προγράμματος, δεν είναι γραμμένο στην αντίστοιχη στήλη του πίνακα παρουσίασης των σχετικών παραπεμπτικών ασθενών, ο οποίος συναντάται στην πάνω περιοχή του προγενέστερου παραθύρου. Επιπλέον, παρέχει την ικανότητα στην πιο άνω γραμματεία να εισάγει στο πρωτύερο πεδίο κειμένου ένα ID παραπεμπτικού ασθενούς, το οποίο είναι καταχωρημένο στο ανάλογο πεδίο του προηγούμενου πίνακα, κάνοντας κλικ στο κουμπί «Δοκιμάστε ξανά» του πιο άνω παραθύρου.

Αμα, φυσικά, η γραμματεία του εκάστοτε νοσοκομείου, η οποία αλληλεπιδρά με τη διαδικτυακή εφαρμογή η οποία αναπτύσσεται στην παρούσα μελέτη, καταχωρήσει στο πεδίο κειμένου το οποίο υπάρχει στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ασθενών ως προς την καταχώρησης αποτελέσματος εξέτασης του εν λόγω προγράμματος ένα ID παραπεμπτικού ασθενούς, το οποίο είναι καταχωρημένο στην αντίστοιχη στήλη του πίνακα εμφάνισης των σχετικών παραπεμπτικών ασθενών, ο οποίος είναι τοποθετημένος στην πάνω μεριά του προαναφερόμενου παραθύρου, τότε, πατώντας η ανώτερη γραμματεία στο κουμπί «Επιλογή» του προγενέστερου παραθύρου, θα προκύψει το πιο κάτω παράθυρο:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων			
Κατηγορίες	Αποτέλεσμα Εξέτασης	Χρήστης	
Ορισμός Ραντεβού	Προκειμένου να καταχωρήσετε το αποτέλεσμα της εξέτασης για το παραπεμπτικό με ID 4377946 του νοσοκομείου σας, παρακαλούμε καταγράψτε το συγκεκριμένο αποτέλεσμα στο πεδίο "Αποτέλεσμα Εξέτασης" που ακολουθεί και κατόπιν πατήστε στο κουμπί "Υποβολή" στο τέλος της τρέχουσας φόρμας:	Κωδικός Νοσοκομείου: 10001	
Αποτέλεσμα Εξέτασης		Όνομασία Νοσοκομείου: ΓΑΣΩ Θεσσαλίας	
Ιστορικό Ασθενούς		<input type="button" value="Αποσύνδεση"/>	
Ιστορικό Νοσοκομείου			
ID Παραπεμπτικού		<input type="text" value="4377946"/>	
ID Ιατρού		<input type="text" value="1"/>	
ΑΜΚΑ Ασθενούς		<input type="text" value="01014025894"/>	
Ασφάλεια Ασθενούς		<input type="text" value="ΟΓΑ"/>	
ID Νοσοκομείου		<input type="text" value="10001"/>	
Αιτία Παραπομπής		<input type="text" value="Εκφύλιση χωρίς κηλίδας."/>	
Ημερομηνία Εξέτασης	<input type="text" value="10/05/2016"/>		
Ωρα Εξέτασης	<input type="text" value="10:30"/>		
Αποτέλεσμα Εξέτασης	<input type="text"/>		
	<input type="button" value="Υποβολή"/>		

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΑΣ (ΑΜ 679)

Εικόνα 90: Το παράθυρο της καταχώρησης αποτελέσματος εξέτασης για έναν ασθενή από ένα νοσοκομείο

Το παράθυρο που δίδεται στην πιο άνω Εικόνα δίνει την ικανότητα στη γραμματεία του εκάστοτε νοσοκομείου, η οποία χρησιμοποιεί το σύστημα λογισμικού Διαδικτύου το οποίο προσδιορίζεται στο παρόν έργο, να καταχωρήσει το αποτέλεσμα της εξέτασης για το παραπεμπτικό ασθενούς το οποίο επιλέχθηκε από την εν λόγω γραμματεία στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ως προς την καταχώρηση αποτελέσματος εξέτασης αυτού του λογισμικού. Για να πραγματοποιήσει κάτι τέτοιο η συγκεκριμένη γραμματεία πρέπει να συμπληρώσει κατάλληλα το πεδίο «Αποτέλεσμα Εξέτασης» της φόρμας καταχώρησης αποτελέσματος εξέτασης του συγκεκριμένου συστήματος (δεν είναι ανάγκη να γεμίσει τα άλλα πεδία της προηγούμενης φόρμας, καθώς τα συγκεκριμένα πεδία θα έχουν ήδη συμπληρωθεί αυτόματα, με το που θα αναδυθεί το ανώτερο παράθυρο) και να υποβάλει έπειτα τη συγκεκριμένη φόρμα στο διακομιστή Ιστού του τρέχοντος προγράμματος, πιέζοντας το κουμπί «Υποβολή» της ανώτερης εικόνας. Τότε, εμφανίζεται το παράθυρο που απεικονίζεται στην κατοπινή Εικόνα:



Εικόνα 91: Το παράθυρο επιβεβαίωσης της επιτυχούς καταχώρησης αποτελέσματος εξέτασης για έναν ασθενή από ένα νοσοκομείο

Το παράθυρο που φαίνεται στην παραπάνω Εικόνα πληροφορεί τη γραμματεία του εκάστοτε νοσοκομείου, η οποία συνιστά τον τρέχων χρήστη της εφαρμογής η οποία καθορίζεται στην παρούσα εργασία, πως καταχώρησε με επιτυχία το αποτέλεσμα της σχετιζόμενης εξέτασης με το παραπεμπτικό ασθενούς, το οποίο είχε επιλεγεί στο παράθυρο επιλογής εκκρεμών παραπεμπτικών ως προς την καταχώρηση αποτελέσματος εξέτασης αυτής της εφαρμογής, στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί η εν λόγω εφαρμογή. Επιπρόσθετα, παρέχει στη συγκεκριμένη γραμματεία την ικανότητα να πλοηγηθεί στο αρχικό παράθυρο νοσοκομείου της τρέχουσας εφαρμογής, «κλικάροντας» το κουμπί «Εντάξει» του πιο άνω παραθύρου.

Κάπου εδώ, να τονιστεί πως, άμα δεν έχουν αποσταλεί καθόλου παραπεμπτικά ασθενών σε ένα νοσοκομείο ή η γραμματεία του εν λόγω νοσοκομείου έχει καταχωρήσει όλα τα αποτελέσματα των εξετάσεων, οι οποίες έχουν ολοκληρωθεί στο προαναφερόμενο νοσοκομείο, στους κατάλληλους ηλεκτρονικούς φακέλους των ασθενών, τότε, εφόσον η προαναφερόμενη γραμματεία επιλέξει την κατηγορία «Ορισμός Ραντεβού» στο αρχικό παράθυρο νοσοκομείου του λογισμικού το οποίο μελετάται στην τρέχουσα διπλωματική, θα αναδυθεί το παράθυρο που δείχνεται στην πιο κάτω Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αποτέλεσμα Εξέτασης	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Δεν υπάρχουν εκκρεμή παραπεμπτικά ως προς την καταχώρηση αποτελέσματος εξέτασης στο νοσοκομείο σας!!!</p> <p style="text-align: center;"><input type="button" value="Εντάξει"/></p>	<ul style="list-style-type: none"> Κωδικός Νοσοκομείου: 10001 Όνομασία Νοσοκομείου: ΙΑΣΩ Θεσσαλίας <input type="button" value="Αποσύνδεση"/>

ΑΓΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΥΓΑΣ (ΑΜ 679)

Εικόνα 92: Το παράθυρο πληροφόρησης για τη μη ύπαρξη εκκρεμών παραπεμπτικών ως προς την καταχώρηση αποτελέσματος εξέτασης σε ένα νοσοκομείο

Το παράθυρο που δείχνεται στην ανώτερη Εικόνα γνωστοποιεί στη γραμματεία του εκάστοτε νοσοκομείου, η οποία αποτελεί τον τρέχων χρήστη του συστήματος το οποίο υλοποιείται στην τρέχουσα πτυχιακή, πως δεν υπάρχουν παραπεμπτικά ασθενών σε αναμονή καταχώρησης αποτελέσματος εξέτασης για το εν λόγω νοσοκομείο, είτε γιατί δεν έχει αποσταλεί κανένα παραπεμπτικό ασθενούς σε αυτό το νοσοκομείο είτε γιατί η γραμματεία του έχει καταχωρήσει όλα τα αποτελέσματα των εξετάσεων οι οποίες έχουν πραγματωθεί στο προγενέστερο νοσοκομείο, στους απαραίτητους ηλεκτρονικούς φακέλους των ασθενών. Ακόμα, προσφέρει στην προγενέστερη γραμματεία την ικανότητα να πλοηγηθεί στο αρχικό παράθυρο νοσοκομείου του παρόντος συστήματος, πιέζοντας το κουμπί «Εντάξει» του πιο πάνω παραθύρου.

Αν, τώρα, η γραμματεία του εκάστοτε νοσοκομείου, η οποία πραγματώνει είσοδο στο λογισμικό, το οποίο καθορίζεται στην παρούσα πτυχιακή, χρειαστεί να αναζητήσει το σχετικό με την οφθαλμιατρική υγεία ιστορικό ενός ασθενούς, θα πρέπει να διαλέξει την κατηγορία «Ιστορικό Ασθενούς» της αριστερής στήλης του αρχικού παραθύρου νοσοκομείου του εν λόγω λογισμικού. Τότε θα αναδυθεί το παράθυρο το οποίο αναπαρίσταται στην παρακάτω Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενή	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενή Ιστορικό Νοσοκομείου 	<p>Προκειμένου να αναζητήσετε το ιστορικό της οφθαλμολογικής υγείας ενός ασθενή, παρακαλούμε πληκτρολογήστε τον ΑΜΚΑ του στο αμέσως επόμενο πλαίσιο κειμένου:</p> <div style="border: 1px solid black; width: 100px; height: 20px; margin: 10px auto;"></div> <p>και κατόπιν πατήστε στο κουμπί "Αναζήτηση":</p> <div style="text-align: center; margin-top: 20px;"> <input type="button" value="Αναζήτηση"/> </div>	<p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="Αποσύνδεση"/> </div>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 93: Το παράθυρο της αναζήτησης του ιστορικού ενός ασθενούς από ένα νοσοκομείο

Το παράθυρο που δείχνεται στην παραπάνω Εικόνα προσφέρει την ευχέρεια στη γραμματεία του εκάστοτε νοσοκομείου, η οποία αλληλεπιδρά με το σύστημα το οποίο αναλύεται στην τρέχουσα αναφορά, να αναζητήσει το ιστορικό οποιουδήποτε ασθενούς, το οποίο έχει να κάνει με την υγεία των οφθαλμών αυτού του ασθενούς, καταχωρώντας στο πεδίο κειμένου που υπάρχει στο ανώτερο παράθυρο τον ΑΜΚΑ του προαναφερόμενου ασθενούς και πιέζοντας στη συνέχεια το κουμπί «Αναζήτηση» που βρίσκεται στην κάτω περιοχή του πιο πάνω παραθύρου.

Ωστόσο, εφόσον η γραμματεία του εκάστοτε νοσοκομείου, η οποία εισέρχεται στην εφαρμογή Διαδικτύου, η οποία παρουσιάζεται στην τρέχουσα πτυχιακή, καταχωρήσει στο πεδίο κειμένου το οποίο περιέχεται στο παράθυρο αναζήτησης ιστορικού ασθενούς (από τη γραμματεία νοσοκομείου) αυτής της εφαρμογής έναν ΑΜΚΑ ασθενούς ο οποίος δεν απαντάται στο αντίστοιχο πεδίο της βάσης δεδομένων με την οποία συνδέεται η προηγούμενη εφαρμογή, τότε, πατώντας η εν λόγω γραμματεία το κουμπί «Αναζήτηση» του πιο άνω παραθύρου, θα ξεπροβάλλει το παράθυρο της κατοπινής Εικόνας:

Ο ΑΜΚΑ που πληκτρολογήσατε δεν αντιστοιχεί σε κάποιον ασθενή!!!

Εικόνα 94: Το παράθυρο προειδοποίησης της ανεπιτυχούς αναζήτησης του ιστορικού ενός ασθενούς από ένα νοσοκομείο

Το παράθυρο που δίδεται στην πιο πάνω Εικόνα προειδοποιεί τη γραμματεία του εκάστοτε νοσοκομείου, η οποία είναι συνδεδεμένη στο πρόγραμμα το οποίο περιγράφεται στην τρέχουσα πτυχιακή, πως ο ΑΜΚΑ ασθενούς τον οποίο κατέγραψε στο πεδίο κειμένου το οποίο υφίσταται στο παράθυρο αναζήτησης ιστορικού ασθενούς (από τη γραμματεία νοσοκομείου) του εν λόγω προγράμματος δεν είναι εγγεγραμμένο στο ανάλογο πεδίο της βάσης δεδομένων με την οποία επικοινωνεί το προαναφερόμενο πρόγραμμα. Ακόμα, δίνει τη δυνατότητα στην προαναφερόμενη γραμματεία να γράψει στο παραπάνω πεδίο κειμένου έναν ΑΜΚΑ ασθενούς το οποίο είναι καταχωρημένο στο αντίστοιχο πεδίο της πρωτότερης βάσης, κάνοντας κλικ στο κουμπί «Δοκιμάστε ξανά» του παραπάνω παραθύρου.

Αμα, βέβαια, η γραμματεία του εκάστοτε νοσοκομείου, η οποία χειρίζεται το σύστημα λογισμικού, το οποίο αναλύεται στην τρέχουσα εργασία εισάγει στο πεδίο κειμένου το οποίο περικλείεται στο παράθυρο αναζήτησης ιστορικού ασθενούς (από τη γραμματεία νοσοκομείου) αυτού του συστήματος έναν ορθό ΑΜΚΑ ασθενούς, τότε, πατώντας η προαναφερόμενη γραμματεία το κουμπί «Αναζήτηση» του πρωτύτερου παραθύρου, ανάλογα με το αν ο προαναφερόμενος ασθενής έχει ή δεν έχει έστω και μία φορά παραπεμφθεί από οποιονδήποτε ιατρό σε κάποια οφθαλμιατρική κλινική, για αναλυτικότερες εξετάσεις, θα εμφανιστούν, αντίστοιχα, τα παράθυρα που δείχνονται στις επόμενες δύο Εικόνες:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενή	Χρήστης																														
Ορισμός Ραντεβού	<p>Το ιστορικό του ασθενή με ΑΜΚΑ 01014025894 φαίνεται στον ακόλουθο πίνακα:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">ΑΜΚΑ Ασθενή</th> <th style="width: 15%;">Ασφάλεια Ασθενή</th> <th style="width: 15%;">Αιτία Παραπομπής</th> <th style="width: 15%;">Ονομασία Νοσοκομείου</th> <th style="width: 15%;">Ημερομηνία εξέτασης</th> <th style="width: 20%;">Αποτέλεσμα εξέτασης</th> </tr> </thead> <tbody> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Μυωπία.</td> <td>Υγεία</td> <td>02/02/1980</td> <td>Αφαίρεση με χειρουργείο.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Καταρράκτης.</td> <td>Υγεία</td> <td>12/03/2012</td> <td>Αφαίρεση με λέιζερ.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Πίεση.</td> <td>ΙΑΣΩ Θεσσαλίας</td> <td>14/05/2015</td> <td>Φαρμακευτική αγωγή.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Εκφύλιση ωχράς κηλίδας.</td> <td>Υγεία</td> <td>01/06/2016</td> <td>Ενδοθεραπεία.</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 10px;">Εντόξα</p>	ΑΜΚΑ Ασθενή	Ασφάλεια Ασθενή	Αιτία Παραπομπής	Ονομασία Νοσοκομείου	Ημερομηνία εξέτασης	Αποτέλεσμα εξέτασης	01014025894	ΟΓΑ	Μυωπία.	Υγεία	02/02/1980	Αφαίρεση με χειρουργείο.	01014025894	ΟΓΑ	Καταρράκτης.	Υγεία	12/03/2012	Αφαίρεση με λέιζερ.	01014025894	ΟΓΑ	Πίεση.	ΙΑΣΩ Θεσσαλίας	14/05/2015	Φαρμακευτική αγωγή.	01014025894	ΟΓΑ	Εκφύλιση ωχράς κηλίδας.	Υγεία	01/06/2016	Ενδοθεραπεία.	Κωδικός Νοσοκομείου: 10000
ΑΜΚΑ Ασθενή		Ασφάλεια Ασθενή	Αιτία Παραπομπής	Ονομασία Νοσοκομείου	Ημερομηνία εξέτασης	Αποτέλεσμα εξέτασης																										
01014025894		ΟΓΑ	Μυωπία.	Υγεία	02/02/1980	Αφαίρεση με χειρουργείο.																										
01014025894		ΟΓΑ	Καταρράκτης.	Υγεία	12/03/2012	Αφαίρεση με λέιζερ.																										
01014025894		ΟΓΑ	Πίεση.	ΙΑΣΩ Θεσσαλίας	14/05/2015	Φαρμακευτική αγωγή.																										
01014025894	ΟΓΑ	Εκφύλιση ωχράς κηλίδας.	Υγεία	01/06/2016	Ενδοθεραπεία.																											
Αποτέλεσμα Εξέτασης		Ονομασία Νοσοκομείου: Υγεία																														
Ιστορικό Ασθενή		Αποσύνδεση																														
Ιστορικό Νοσοκομείου																																

ΔΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 95: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος έχει παραπεμφθεί έστω μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περεταίρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από ένα νοσοκομείο

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενή	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενή Ιστορικό Νοσοκομείου 	<p>Για τον ασθενή με ΑΜΚΑ 04055536363 δεν έχει συνταχθεί κάποιο παραπεμπτικό και συνεπώς δεν έχει πραγματοποιήσει κάποια εξέταση!!</p> <p>Εντάξει</p>	<ul style="list-style-type: none"> Κωδικός Νοσοκομείου: 10000 Όνομασία Νοσοκομείου: Υγεία Αποσύνδεση

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΙΑΣ (ΑΜ 679)

Εικόνα 96: Το παράθυρο εμφάνισης του ιστορικού ενός ασθενούς, ο οποίος δεν έχει παραπεμφθεί ούτε μία φορά από οποιονδήποτε ιατρό σε ένα νοσοκομείο για περεταίρω εξετάσεις, κατόπιν αναζήτησης του ιστορικού του ασθενούς από ένα νοσοκομείο

Το μεν πρώτο από τα δύο ανώτερα παράθυρα απεικονίζει το ιστορικό της οφθαλμιατρικής υγείας ενός ασθενούς, ο οποίος έχει τουλάχιστον μία φορά παραπεμφθεί από οποιονδήποτε ιατρό σε κάποιο νοσοκομείο. Το δε δεύτερο από τα δύο πιο πάνω παράθυρα εμφανίζει το ιστορικό της υγείας των ματιών ενός ασθενούς, ο οποίος καμία φορά δεν παραπεμφθεί από οποιονδήποτε ιατρό σε κάποιο νοσοκομείο. Στο σημείο αυτό, να αναφερθεί πως και στα δύο ανώτερα παράθυρα η αναζήτηση του ιστορικού ασθενούς γίνεται από τη γραμματεία του εκάστοτε νοσοκομείου, η οποία εισέρχεται στο σύστημα Διαδικτύου το οποίο μελετάται στην τρέχουσα εργασία. Επίσης, και τα δύο πιο άνω παράθυρα παρέχουν στην προαναφερόμενη γραμματεία την ευχέρεια να πλοηγηθεί στο αρχικό παράθυρο νοσοκομείου αυτού του συστήματος, πιέζοντας το κουμπί «Εντάξει» που υπάρχει και στα δύο πιο άνω παράθυρα.

Τέλος, από τη στιγμή που η γραμματεία του εκάστοτε νοσοκομείου, η οποία πραγματώνει λειτουργεί την εφαρμογή, η οποία μελετάται στην τρέχουσα αναφορά, θεωρήσει επιθυμητό να κάνει αναζήτηση των εξετάσεων ασθενών που πρόκειται να γίνουν ή έχουν γίνει στο νοσοκομείο της, θα χρειαστεί να επιλέξει την κατηγορία «Ιστορικό Νοσοκομείου» της αριστερής στήλης του αρχικού παραθύρου νοσοκομείου της προαναφερόμενης εφαρμογής. Τότε θα προβάλει το παράθυρο το οποίο αναπαρίσταται στην παρακάτω Εικόνα:

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Νοσοκομείου	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενή Ιστορικό Νοσοκομείου 	<p>Προκειμένου να αναζητήσετε το ιστορικό των οφθαλμολογικών εξετάσεων που έχουν πραγματοποιηθεί ή πρόκειται να πραγματοποιηθούν στο νοσοκομείο σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;">Αναζήτηση</p>	<ul style="list-style-type: none"> Κωδικός Νοσοκομείου: 10000 Όνομασία Νοσοκομείου: Υγεία Αποσύνδεση

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΑΣ (ΑΜ 679)

Εικόνα 97: Το παράθυρο της αναζήτησης του ιστορικού ενός νοσοκομείο από το ίδιο το νοσοκομείο

Το παράθυρο που φαίνεται στην πιο πάνω Εικόνα προσφέρει την ευκαιρία στη γραμματεία του εκάστοτε νοσοκομείου, η οποία αλληλεπιδρά με το διαδικτυακό σύστημα το οποίο εξετάζεται στην παρούσα πτυχιακή, να μελετήσει όλες τις εξετάσεις ασθενών που είτε έχουν πραγματοποιηθεί είτε πρόκειται να ολοκληρωθούν στο νοσοκομείο της, πιέζοντας το κουμπί «Αναζήτηση» το οποίο είναι τοποθετημένο στο ανώτερο παράθυρο. Τότε, ανάλογα με το αν στην εν λόγω γραμματεία έχει ή δεν έχει αποσταλεί έστω και ένα παραπεμπτικό ασθενούς, θα προκύψουν, αντίστοιχα, τα παράθυρα που φαίνονται στις δύο μεταγενέστερες Εικόνες:

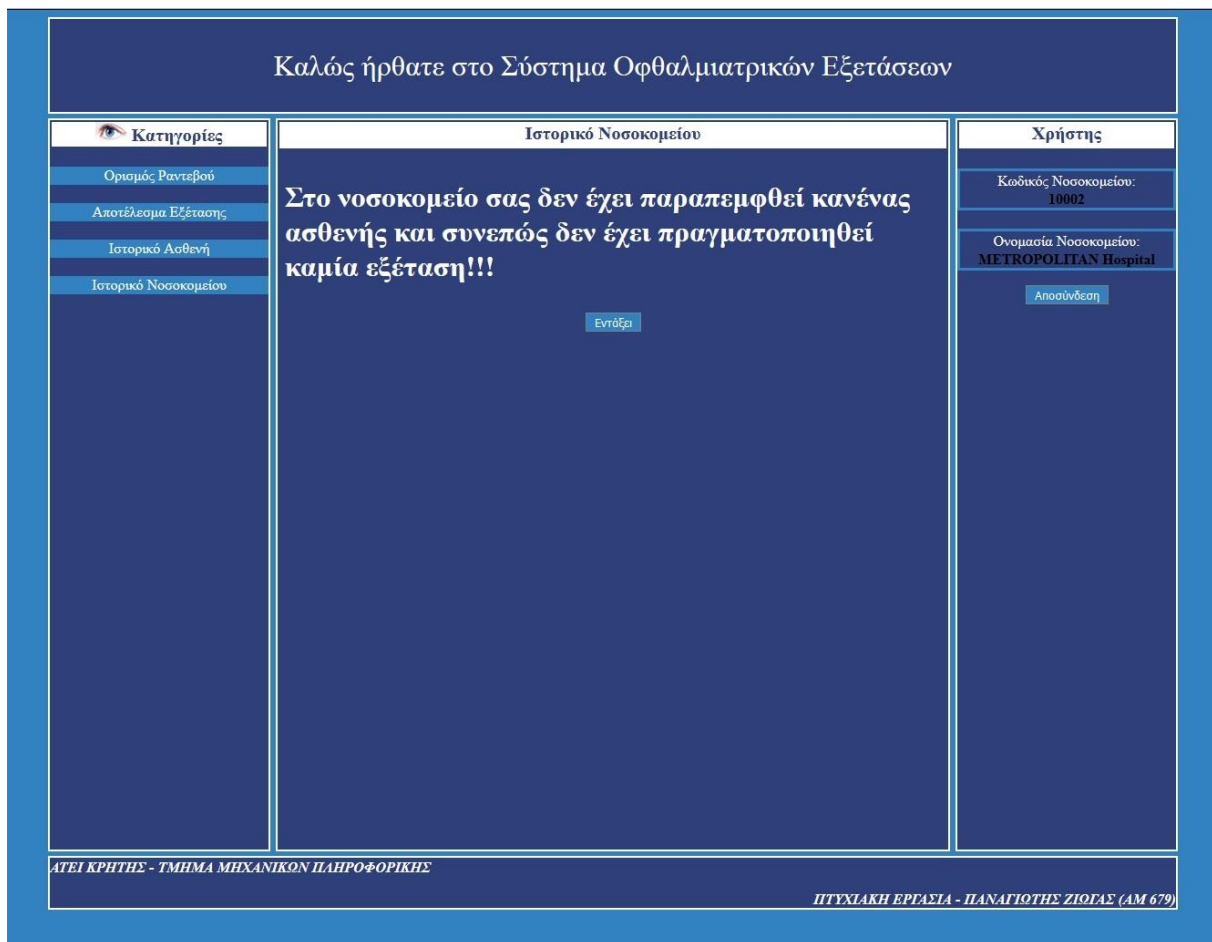
Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

👁️ Κατηγορίες	Ιστορικό Νοσοκομείου	Χρήστης																																																						
Ορισμός Ραντεβού	<p style="text-align: center;">Το ιστορικό σας φαίνεται στον ακόλουθο πίνακα:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID Νοσοκομείου</th> <th>ΑΜΚΑ Ασθενή</th> <th>Ασφάλεια Ασθενή</th> <th>Αιτία Παραπομπής</th> <th>Ημερομηνία Εξέτασης</th> <th>Αποτέλεσμα Εξέτασης</th> </tr> </thead> <tbody> <tr> <td>10000</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Μυωπία.</td> <td>02/02/1980</td> <td>Αφαίρεση με χειρουργείο.</td> </tr> <tr> <td>10000</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Καταρράκτης.</td> <td>12/03/2012</td> <td>Αφαίρεση με λέιζερ.</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Αστιγματισμός.</td> <td>24/11/2013</td> <td>Σύσταση για χρήση φακών επαφής.</td> </tr> <tr> <td>10000</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Εκφύλιση ωχράς κηλίδας.</td> <td>01/06/2016</td> <td>Ενεσθεραπεία.</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Κερατόκωνος.</td> <td>12/06/2016</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> <tr> <td>10000</td> <td>14058321543</td> <td>ΙΚΑ</td> <td>Πίεση.</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Γλαύκωμα.</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Εξασθένηση οπτικών νεύρων.</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 5px;">Εντάξει</p>	ID Νοσοκομείου	ΑΜΚΑ Ασθενή	Ασφάλεια Ασθενή	Αιτία Παραπομπής	Ημερομηνία Εξέτασης	Αποτέλεσμα Εξέτασης	10000	01014025894	ΟΓΑ	Μυωπία.	02/02/1980	Αφαίρεση με χειρουργείο.	10000	01014025894	ΟΓΑ	Καταρράκτης.	12/03/2012	Αφαίρεση με λέιζερ.	10000	18759641223	ΔΗΜΟΣΙΟ	Αστιγματισμός.	24/11/2013	Σύσταση για χρήση φακών επαφής.	10000	01014025894	ΟΓΑ	Εκφύλιση ωχράς κηλίδας.	01/06/2016	Ενεσθεραπεία.	10000	18759641223	ΔΗΜΟΣΙΟ	Κερατόκωνος.	12/06/2016	Δεν έχει καταχωρηθεί!!!	10000	14058321543	ΙΚΑ	Πίεση.	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!	10000	18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!	10000	18759641223	ΔΗΜΟΣΙΟ	Εξασθένηση οπτικών νεύρων.	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!	Κωδικός Νοσοκομείου: 10000
ID Νοσοκομείου		ΑΜΚΑ Ασθενή	Ασφάλεια Ασθενή	Αιτία Παραπομπής	Ημερομηνία Εξέτασης	Αποτέλεσμα Εξέτασης																																																		
10000		01014025894	ΟΓΑ	Μυωπία.	02/02/1980	Αφαίρεση με χειρουργείο.																																																		
10000		01014025894	ΟΓΑ	Καταρράκτης.	12/03/2012	Αφαίρεση με λέιζερ.																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Αστιγματισμός.	24/11/2013	Σύσταση για χρήση φακών επαφής.																																																		
10000		01014025894	ΟΓΑ	Εκφύλιση ωχράς κηλίδας.	01/06/2016	Ενεσθεραπεία.																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Κερατόκωνος.	12/06/2016	Δεν έχει καταχωρηθεί!!!																																																		
10000		14058321543	ΙΚΑ	Πίεση.	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Εξασθένηση οπτικών νεύρων.	Δεν έχει οριστεί!!!	Δεν έχει καταχωρηθεί!!!																																																		
Αποτέλεσμα Εξέτασης		Όνομασία Νοσοκομείου: Υγεία																																																						
Ιστορικό Ασθενή		Αποσύνδεση																																																						
Ιστορικό Νοσοκομείου																																																								

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΙΑΣ (ΑΜ 679)

Εικόνα 98: Το παράθυρο της αναζήτησης του ιστορικού ενός νοσοκομείου από το ίδιο το νοσοκομείο, στο οποίο έχει αποσταλεί τουλάχιστον ένα παραπεμπτικό ασθενούς



Εικόνα 99: Το παράθυρο της αναζήτησης του ιστορικού ενός νοσοκομείου από το ίδιο το νοσοκομείο, στο οποίο δεν έχει αποσταλεί ούτε ένα παραπεμπτικό ασθενούς

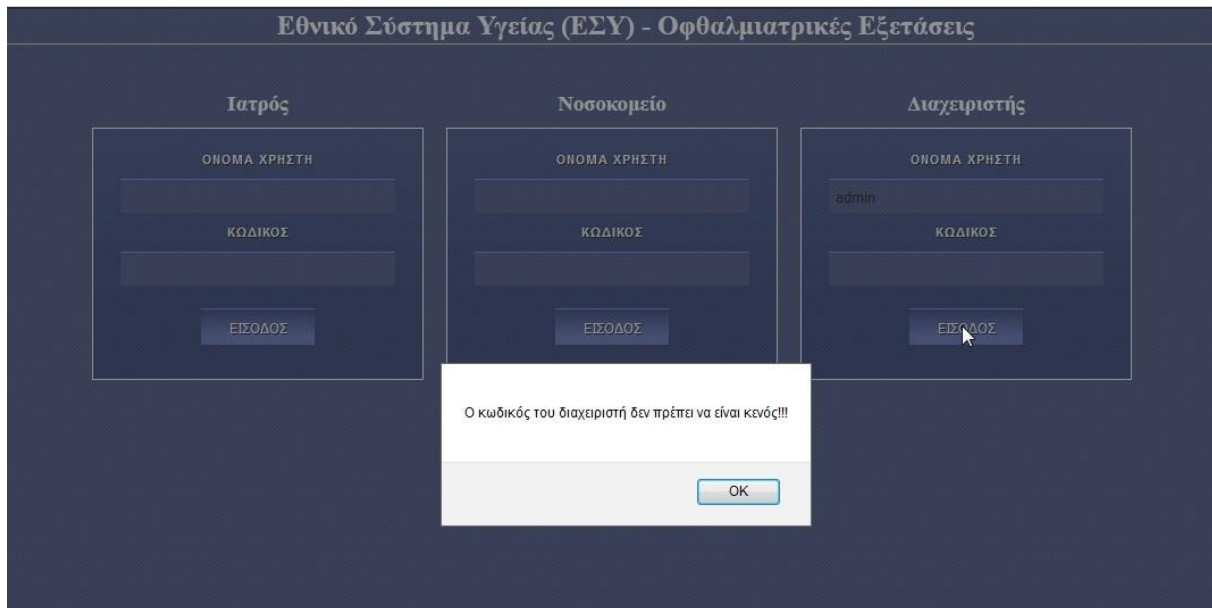
Το μεν πρώτο από τα δύο πιο άνω παράθυρα απεικονίζει τις εξετάσεις ασθενών που έχουν γίνει ή πρόκειται να γίνουν στο εκάστοτε νοσοκομείο, η γραμματεία του οποίου είναι ο τρέχων χρήστης του λογισμικού που υλοποιείται στα πλαίσια τούτης της διπλωματικής αναφοράς και πραγματοποιεί την αναζήτηση των προαναφερόμενων εξετάσεων. Το δε δεύτερο από τα δύο πιο πάνω παράθυρα πληροφορεί την προαναφερόμενη γραμματεία ότι δεν έχει αποσταλεί στο νοσοκομείο της κανένα παραπεμπτικό ασθενούς. Συνάμα, και τα δύο παραπάνω παράθυρα δίνουν τη δυνατότητα στην προηγούμενη γραμματεία να μετακινηθεί στο αρχικό παράθυρο νοσοκομείου του εν λόγω λογισμικού, κάνοντας κλικ στο κουμπί «Εντάξει» που περιλαμβάνεται και στα δύο παραπάνω παράθυρα.

3.4.4 Η Επικύρωση των Πεδίων των Φορμών του Συστήματος Λογισμικού της Τρέχουσας Πτυχιακής Εργασίας

Ένα ιδιαίζουσα σημασίας κομμάτι που οφείλει να λάβει σοβαρά υπόψιν του ο εκάστοτε χρήστης, είτε διαχειριστής, είτε ιατρός, είτε ασθενής, του συστήματος λογισμικού το οποίο αναπτύσσεται στα πλαίσια της τρέχουσας διπλωματικής εργασίας είναι η επικύρωση (validation) των πεδίων των φορμών οι οποίες απαντώνται στα επιμέρους παράθυρα που συγκροτούν το προαναφερόμενο σύστημα. Όσον αφορά, λοιπόν, την εν λόγω επικύρωση, πρέπει σε γενικές γραμμές να τονιστούν τα ακόλουθα δύο σημεία:

Πρώτον, δεν πρέπει σε καμία περίπτωση ο εκάστοτε χρήστης της διαδικτυακής εφαρμογής την οποία πραγματεύεται η τρέχουσα αναφορά, κατά τη φάση συμπλήρωσης των πεδίων μίας φόρμας της εν λόγω εφαρμογής να αφήσει κενό κάποιο από προαναφερόμενα πεδία, εκτός αν επισημαίνεται ότι δεν είναι υποχρεωτική η συμπλήρωση του. Σε αντίθετη περίπτωση ο χρήστης αυτός θα λάβει κατάλληλο

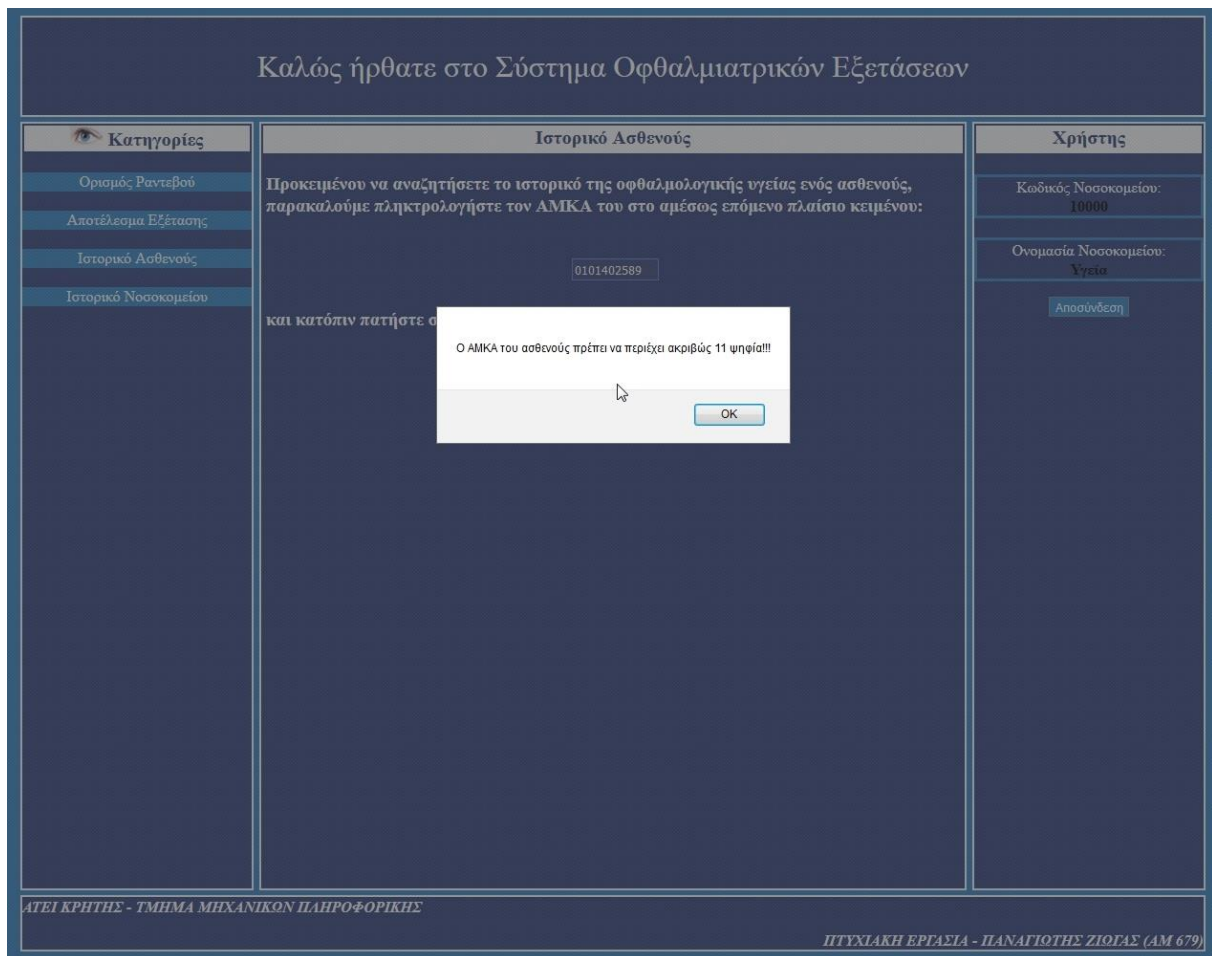
μήνυμα προειδοποίησης παρόμοιο με αυτό που φαίνεται στο παράθυρο που απεικονίζεται στην κατώτερη Εικόνα:



Εικόνα 100: Το παράθυρο προειδοποίησης που αναδύεται, όταν ένας χρήστης-διαχειριστής του λογισμικού δεν καταχωρήσει τον κωδικό του στη φόρμα εισόδου του λογισμικού

Το παράθυρο που δίδεται στην πιο πάνω Εικόνα προειδοποιεί τον εκάστοτε χρήστη-διαχειριστή του προγράμματος το οποίο περιγράφεται στην τρέχουσα εργασία, ότι δεν μπορεί να εισέλθει στο σύστημα, αν δεν καταχωρήσει έναν κωδικό στο πεδίο «ΚΩΔΙΚΟΣ» της φόρμας εισόδου του σε αυτό το πρόγραμμα. Ακόμα, δίνει τη δυνατότητα στον τρέχον χρήστη να εισάγει στο παραπάνω πεδίο έναν κωδικό, κάνοντας κλικ στο κουμπί «OK», του πιο πάνω παραθύρου.

Δεύτερον, δεν επιτρέπεται σε καμία περίπτωση ο εκάστοτε χρήστης του λογισμικού το οποίο λαμβάνει χώρα στην παρούσα διατριβή, κατά την ενέργεια καταχώρησης δεδομένων στα πεδία μίας φόρμας του εν λόγω λογισμικού, να εισάγει άκυρα δεδομένα. Κάτι τέτοιο μπορεί να συμβεί είτε γιατί τα εν λόγω δεδομένα δε θα συμφωνούν με τη μορφή που πρέπει να έχουν και η οποία θα υποδεικνύεται άμεσα στο παράθυρο το οποίο περιέχει την προαναφερόμενη φόρμα, είτε διότι δε θα συνάδουν εξ ορισμού με το μέγεθος που αντιπροσωπεύουν, είτε επειδή δε θα συμβαδίζουν με την κοινή λογική. Για καθεμία από τις προηγούμενες τρεις περιπτώσεις ο παρών χρήστης θα λαμβάνει σχετικά μηνύματα προειδοποίησης, όμοια με αυτά τα οποία παρουσιάζονται, κατά αντιστοιχία, στα παράθυρα τα οποία φαίνονται στις μετέπειτα τρεις Εικόνες:



Εικόνα 102: Το παράθυρο προειδοποίησης που αναδύεται, όταν ένας χρήστης-ιατρός του λογισμικού καταχωρήσει στη φόρμα αναζήτησης ιστορικού ασθενούς του λογισμικού ένα ΑΜΚΑ ασθενούς το οποίο δε συμφωνεί με τις εξ ορισμού ιδιότητες της έννοιας ΑΜΚΑ

Το παράθυρο που δείχνεται στην ανώτερη Εικόνα ειδοποιεί τον εκάστοτε χρήστη-ιατρό του πληροφοριακού συστήματος το οποίο αναδεικνύεται στο παρόν έργο, πως δεν μπορεί να εισάγει στο πεδίο κειμένου της φόρμας αναζήτησης ιστορικού ασθενούς έναν δεκαψήφιο ΑΜΚΑ, τη στιγμή που ο ΑΜΚΑ ως έννοια ορίζεται αποκλειστικά από έντεκα ακριβώς ψηφία. Ακόμα, δίνει τη δυνατότητα στον παρών χρήστη να καταχωρήσει στο προαναφερόμενο πεδίο έναν έγκυρο ΑΜΚΑ ασθενούς, πατώντας το κουμπί «OK», του ανώτερου παραθύρου.

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ορισμός Ραντεβού	Χρήστης
Ορισμός Ραντεβού	<p>Προκειμένου να ορίσετε ραντεβού για το εκκρεμές παραπεμπτικό με ID 6124477 του νοσοκομείου σας, παρακαλούμε συμπληρώστε τόσο την ημερομηνία όσο και την ώρα εξέτασης στα πεδία "Ημερομηνία Εξέτασης" και "Ωρα Εξέτασης", αντίστοιχα, που ακολουθούν και κατόπιν πατήστε στο κουμπί "Υποβολή" στο τέλος της τρέχουσας φόρμας:</p> <p>ID Παραπεμπτικού: 6124477</p> <p>ID Ιατρού: <input type="text"/></p> <p>ΑΜΚΑ Ασθενή: <input type="text"/></p> <p>Ασφάλεια Ασθενή: <input type="text"/></p> <p>Αιτία Παραπομπής: <input type="text" value="Κερατόκωνος"/></p> <p>ID Νοσοκομείου: 10000</p> <p>Ημερομηνία Εξέτασης: 04/04/2016</p> <p>Ωρα Εξέτασης: 16:30</p> <p style="text-align: center;"><input type="button" value="Υποβολή"/></p>	<p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p style="text-align: center;"><input type="button" value="Αποσύνδεση"/></p>

Η ημερομηνία εξέτασης πρέπει να είναι μεταγενέστερη της τρέχουσας!!!

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

Εικόνα 103: Το παράθυρο προειδοποίησης που αναδύεται, όταν ένας χρήστης- νοσοκομείο του λογισμικού καταχωρήσει στη φόρμα ορισμού ραντεβού του λογισμικού μία ημερομηνία εξέτασης ασθενούς η οποία είναι προγενέστερη της τρέχουσας σε κάθε φάση ημερομηνίας

Το παράθυρο που αντικατοπτρίζεται στην παραπάνω Εικόνα πληροφορεί τον εκάστοτε χρήστη-γραμματεία νοσοκομείου του λογισμικού το οποίο προδιαγράφεται στην παρούσα εργασία, πως είναι αδύνατο στη φόρμα ορισμού ραντεβού του εν λόγω λογισμικού να επιλέξει μία ημερομηνία εξέτασης ασθενούς πρωτύτερης της εκάστοτε τρέχουσας ημερομηνίας. Επίσης, του παρέχει την ευχέρεια να επιλέξει μία έγκυρη ημερομηνία εξέτασης ασθενούς, «κλικάροντας» το κουμπί «OK», του πιο άνω παραθύρου.

[77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94][95] [96] [97]

4 Επίλογος της Πτυχιακής Εργασίας

Το τέταρτο και τελευταίο κεφάλαιο τούτης της πτυχιακής εργασίας είναι επιλογικό και από τη μία παρουσιάζει με σύντομο τρόπο τόσο τα συμπεράσματα που προκύπτουν από την ανάλυση του διαδικτυακού συστήματος λογισμικού το οποίο έλαβε χώρα στα πλαίσια της εν λόγω αναφοράς και από την άλλη προτείνει τρόπους επέκτασης του πληροφοριακού συστήματος Διαδικτύου το οποίο ήταν το αντικείμενο μελέτης της παρούσας εργασίας.

4.1 Τα Συμπεράσματα της Πτυχιακής Εργασίας

Η παρούσα διπλωματική εργασία επιχείρησε να εστιάσει και να αναπτύξει μία διαδικτυακή εφαρμογή στην οποία θα αλληλεπιδρούν άμεσα ή έμμεσα ο εκάστοτε διαχειριστής της εν λόγω εφαρμογής, οι συμβεβλημένοι με τα Ασφαλιστικά Ταμεία ενός κράτους ιατροί, η γραμματεία του υφιστάμενου κάθε φορά είτε κρατικού νοσοκομείου είτε ιδιωτικής οφθαλμιατρικής κλινικής, που θα συνεργάζεται με ένα από τα προηγούμενα Ταμεία και φυσικά οι ασφαλισμένοι ασθενείς μίας χώρας. Αρκετά μεγάλη βαρύτητα δόθηκε στο σχεδιαστικό κομμάτι του προαναφερόμενου λογισμικού με τη χρήση μίας σειράς UML διαγραμμάτων, ώστε να αποσαφηνιστούν πλήρως οι απαιτήσεις και οι ανάγκες αυτού του λογισμικού.

Ένα σύστημα, όπως αυτό που εξετάστηκε και υλοποιήθηκε στον παρόν έγγραφο, μπορεί να προάγει σημαντικά το σύστημα Υγείας μίας χώρας, απλοποιώντας διαδικασίες οι οποίες σχετίζονται με τον εν λόγω τομέα και που η διεκπεραίωσή τους θα ήταν ιδιαίτερα χρονοβόρα και κουραστική, δίχως τη χρήση του συγκεκριμένου συστήματος λογισμικού. Επιπλέον, ένα πληροφοριακό σύστημα, όπως το προαναφερόμενο, παρέχει τη δυνατότητα εκπόνησης ηλεκτρονικού ιατρικού φακέλου ασθενούς, ο οποίος, εκτός των άλλων, εξασφαλίζει σε μεγάλο βαθμό, και σίγουρα σε μεγαλύτερο σε σχέση με τον κλασικό χειρόγραφο φάκελο ασθενούς, το ιατρικό απόρρητο.

4.2 Οι Επεκτάσεις της Πτυχιακής Εργασίας

Σε αυτή την διπλωματική διατριβή αν και χρησιμοποιήθηκε Struts Framework το οποίο είναι λίγο παλιό σε σχέση με ότι υπάρχει διαθέσιμο τώρα. Θα μπορούσε φυσικά να χρησιμοποιηθούν πιο επαγγελματικά Frameworks και εργαλεία, αλλά απαιτούνταν μεγαλύτερη εμπειρία και βαθιά γνώση των τεχνικών προγραμματισμού Ιστοτόπων. Σίγουρα, όμως, η εφαρμογή Διαδικτύου που οριοθετήθηκε, προσδιορίστηκε και τελικά υλοποιήθηκε στην τρέχουσα διπλωματική είναι μία καλή βάση για την υλοποίηση πολύπλοκότερων εφαρμογών με τη χρήση νεότερων γλωσσών, τεχνικών και τεχνολογιών προγραμματισμού. Το μόνο σίγουρο είναι ότι οι μελλοντικοί αναγνώστες της τρέχουσας πτυχιακής εργασίας θα αποκτήσουν ένα επαρκές θεωρητικό και πρακτικό υπόβαθρο, όσον αφορά τον προγραμματισμό Διαδικτύου, το οποίο στη συνέχεια μπορούν να αναβαθμίσουν, προκειμένου να εξελιχτούν σε σπεσιαλίστες του είδους, αν και εφόσον το επιθυμούν.

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Ι

ΒΙΒΛΙΟΓΡΑΦΙΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

ΚΕΦΑΛΑΙΟ 1^ο:

- [1] Bruce I. Blum, “Clinical Information Systems”, Springer-Verlag, June 1986
- [2] E.H. Shortliffe, “Computer Applications in Health Care and Biomedicine (3rd Edition)”, Springer-Verlag, July 2006.
- [3] Τ. Μπότσης & Σ. Χαλκιώτης «Η Εφαρμογή της Πληροφορικής στο Χώρο της Υγείας», Εκδόσεις Διάυλος, Μάρτιος 2005.

ΚΕΦΑΛΑΙΟ 2^ο:

- [4] https://en.wikipedia.org/wiki/Web_page
- [5] <https://en.wikipedia.org/wiki/Website>
- [6] <http://www.edevelopers.gr/webpages/>
- [7] Αργύρης Τζικόπουλος, «Δημιουργία Ιστοσελίδας», Σημειώσεις για τα Κέντρα Δία Βίου Μάθησης, Ιούνιος 2013.
- [8] https://en.wikipedia.org/wiki/Web_server
- [9] Jason Brittain, “Tomcat: The Definitive Guide”, O'Reilly Media Inc, February 2011
- [10] Tim Berners-Lee, “HyperText Transfer Protocol”, World Wide Web Consortium, August 2010.
- [11] Ι. Σ. Βενιέρης, Σημειώσεις για τα πρωτόκολλα εφαρμογής HTTP, FTP, E-mail, DNS, Εθνικό Μετσόβειο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Ιανουάριος 2003.
- [12] Andrew S. Tanenbaum, «Δίκτυα Υπολογιστών (4^η Έκδοση)», Εκδόσεις Κλειδάριθμος, Οκτώβριος 2006.
- [13] David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, Sailu Reddy, “HTTP: The Definitive Guide”, O'Reilly Media Inc, September 2002.
- [14] Clinton Wong, “HTTP Pocket Reference: Hypertext Transfer Protocol”, O'Reilly Media Inc, June 2000.
- [15] Courtney Bowman, Ari Geshner, John K. Grant, Daniel Slate, “The Architecture of Privacy”, O'Reilly Media Inc, August 2015.
- [16] Ιωάννης Πρίτζος, Εργαστηριακές Σημειώσεις για το μάθημα «Παράλληλη Κατανεμημένη Επεξεργασία», Πανεπιστήμιο Μακεδονίας, Τμήμα Εφαρμοσμένης Πληροφορικής, Σεπτέμβριος 2006.
- [17] Alok Sinha, “Client-server computing”, Communications of the ACM, July 1992.
- [18] Scott M. Lewandowski, “Frameworks for component-based client-server computing”, ACM Computing Surveys, March 1998.
- [19] Joel Kanter, “Understanding Thin-Client-Server Computing”, Microsoft Press, July 1998.
- [20] Peter Duchessi, “Client-server benefits, problems, best practices”, Communications of the ACM, May 1998.
- [21] Παναγιώτης Ι. Κοτζανικολάου, «Ασφαλή συστήματα κινητών πρακτόρων», Διδακτορική Διατριβή, Πανεπιστήμιο Πειραιά, Τμήμα Πληροφορικής, Ιανουάριος 2003.

- [22] Ιωάννης Ε. Φουκαράκης, «Ανάπτυξη υποδομής κινητών πρακτόρων με χρήση τεχνολογιών Διαδικτύου», Διπλωματική Εργασία, Εθνικό Μετσόβειο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Νοέμβριος 2003.
- [23] Θεόδωρος Ε. Αθανηλέας, «Επέκταση πλατφόρμας κινητών αντιπροσώπων με χρήση δικτυακών υποδοχών», Διπλωματική Εργασία, Εθνικό Μετσόβειο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Ιούνιος 2005.
- [24] Τζανέτος Πομόνης, «Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web», Διδακτορική Διατριβή, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Ιούνιος 2010.
- [25] https://en.wikipedia.org/wiki/Software_framework
- [26] Sue Pielman, “The Struts Framework: Practical Guide for Java Programmers (1st Edition)” Morgan Kaufman Publishers, June 2003.
- [27] Graig Iarman, “Applying UML and patterns (2nd Edition)”, Prentice hall PTR, November 2002.
- [28] Martin Fowler, “UML Distilled (3rd Edition)”, Addison-Wesley, March 2004.
- [29] Κώστας Κοντογιάννης, Σημειώσεις για το μάθημα «Τεχνολογία Λογισμικού», Εθνικό Μετσόβειο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Μάρτιος 2005.
- [30] Γιάννης Τζιτζίκας, Σημειώσεις για το μάθημα «Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων», Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών, Φεβρουάριος 2009.
- [31] Jeremy Keith, “DOM Scripting: Web Design with JavaScript and the Document Object Model (2nd Edition)”, August 2015
- [32] <https://en.wikipedia.org/wiki/HTML>
- [33] Κώστας Κουρκουμπέτης, Εργαστηριακές Σημειώσεις για το μάθημα «Εισαγωγή στην Επιστήμη των Υπολογιστών», Οικονομικό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής, Ιανουάριος 2008.
- [34] Ι. Σ. Βενιέρης, Σημειώσεις για τη γλώσσα HTML, Εθνικό Μετσόβειο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Ιανουάριος 2003.
- [35] Laura Lemay, Rafe Colburn, «Πλήρες εγχειρίδιο της HTML 5 & CSS», Εκδόσεις Μ. Γκιούρδας, Σεπτέμβριος 2011.
- [36] Julie C. Melonie, «Μάθετε HTML 5, CSS και JavaScript», Εκδόσεις Μ. Γκιούρδας, Ιούνιος 2013.
- [37] https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [38] Μανώλης Μπαρκατσέλας, «CSS Θεωρία και Εφαρμογές», Εκδόσεις Ξαρχάκος, Σεπτέμβριος 2013.
- [39] Lea Verou, “CSS Secrets: Better Solutions to Everyday Web Design Problems (1st Edition)”, O’Reilly Media Inc, June 2015.
- [40] <http://www.purelybranded.com/notes/why-use-css-in-website-design/>
- [41] Γιώργος Λιακέας, «Η γλώσσα JavaScript», Εκδόσεις Μ. Γκιούρδας, Σεπτέμβριος 2002.
- [42] Martin Webb, «Πλήρες Εγχειρίδιο της JavaScript», Εκδόσεις Μ. Γκιούρδας, Σεπτέμβριος 2001.
- [43] CrockFouard Douglas, “JavaScript: The Good Parts”, O’Reilly Media Inc, May 2008.
- [44] John Ducket, “JavaScript & JQuery, Interactive Front-end Web Development”, John Wiley & Sons Ltd, July 2014.
- [45] Andy Harris, “JavaScript and Ajax For Dummies”, John Wiley and Sons Ltd, November 2009.
- [46] Peter Wayner, “Java & JavaScript Programming”, Anubis Publishers, June 2014.
- [47] Δημήτριος Σάμψων, «Η Γλώσσα Σήμανσης XML», Πανεπιστημιακές Σημειώσεις, Δεκέμβριος 2003.
- [48] Steven Holzner, «Οδηγός της XML», Εκδόσεις Μ. Γκιούρδας, Σεπτέμβριος 2009.
- [49] Michael J. Young, «XML βήμα βήμα», Εκδόσεις Κλειδάριθμος, Μάρτιος 2011.
- [50] <https://en.wikipedia.org/wiki/XHTML>
- [51] <http://www.webstandards.org/learn/articles/askw3c/oct2003/>
- [52] <http://www.wlearn.gr/index.php/ajax-intro2>

- [53] Kris Handlock, «AJAX Ανάπτυξη Web Εφαρμογών», Εκδόσεις Μ. Γκιούρδας, Μάιος 2007.
- [54] Edmond Woychowsky, “AJAX: Creating Web Pages with Asynchronous JavaScript and XML”, Pearson Education, June 2007.
- [55] Γιώργος Λιακέας, «Εισαγωγή στην Java 2», Εκδόσεις Κλειδάριθμος, Μάρτιος 2003.
- [56] Joshua Bloch, “Effective Java (2nd edition)”, Sun Microsystems Inc, April 2008.
- [57] Paul Deitel & Harvey Deitel, «Java Προγραμματισμός», Εκδόσεις Μ. Γκιούρδας, Μάρτιος 2010.
- [58] Jason Hunter, William Crawford, “Java Servlet Programming, (2nd Edition)”, O'Reilly Media Inc, April 2001.
- [59] Robert Englander, “Developing Java Beans”, O'Reilly Media Inc, June 1997.
- [60] “Java Server Programming”, Kogents Solutions Inc, October 2010.
- [61] George Reese, “Database Programming with JDBC & Java (2nd Edition)”, O'Reilly Media Inc, August 2000.
- [62] Allan R. Williamson, Ceri Moran “Java Database Programming: Servlets & JDBC”, Prentice Hall, November 1997.
- [63] Αθανάσιος Σταυρακούδης, «Βάσεις Δεδομένων και SQL», Εκδόσεις Κλειδάριθμος, Οκτώβριος 2010.
- [64] Jim Melton, Andrew Eisenberg, “Understanding SQL and Java Together”, Academic Press, May 2000.
- [65] Larry Ullman, «Εισαγωγή στη MySQL (2nd Edition)», Εκδόσεις Κλειδάριθμος, Δεκέμβριος 2006.
- [66] Hans Bergsten, “JavaServer Pages (3rd Edition)”, O'Reilly Media Inc, December 2003.
- [67] Santosh Kumar K, “JDBC, Servlets and JSP”, Kogents Solutions Inc, May 2008.
- [68] David Harms, “JSP, Servlets and MySQL”, John Wiley & Sons Ltd, April 2001.

ΚΕΦΑΛΑΙΟ 3^ο:

- [69] https://en.wikipedia.org/wiki/Waterfall_model
- [70] <http://www0.dmst.aueb.gr/louridas/lectures/dais/process/ar01s04.html>
- [71] Ιωάννης Τζιτζικας, «Γενικές Μεθοδολογίες Ανάπτυξης Λογισμικού», Σημειώσεις για το μάθημα «Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων», Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών, Οκτώβριος 2006.
- [72] Μανώλης Γιακουμάκης, Νίκος Διαμαντίδης, «Τεχνολογία Λογισμικού», Εκδόσεις Σταμούλη, Μάρτιος 2009.
- [73] Rajib Mall, “Fundamentals of Software Engineering”, Prentice Hall, May 2003.
- [74] Doug Bell, “Software Engineering for Students, (4th Edition)”, Addison-Wesley, June 2005.
- [75] <https://www.smartdraw.com/support/smartdraw-2016-quickrefguide.pdf>
- [76] <http://www.tutorialspoint.com/uml/uml/>
- [77] <http://www.tutorialspoint.com/eclipse/>
- [78] <https://blog.udemy.com/xampp-tutorial/>
- [79] https://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_More.html
- [80] <http://www.coreservlets.com/Apache-Tomcat-Tutorial/tomcat-7-with-eclipse.html>
- [81] <http://www.raistudies.com/struts-tutorials/>
- [82] <http://www.w3schools.com/html/>
- [83] <https://www.w3.org/Style/CSS/>
- [84] <http://www.w3schools.com/js/>
- [85] <http://www.w3schools.com/jquery/>
- [86] <http://www.tutorialspoint.com/javascript/>
- [87] <http://www.w3schools.com/xml/>
- [88] http://www.w3schools.com/html/html_xhtml.asp

- [89] <http://www.w3schools.com/ajax/>
- [90] http://www.w3schools.com/jquery/jquery_ajax_intro.asp
- [91] <http://www.tutorialspoint.com/java/>
- [92] <http://www.tutorialspoint.com/servlets/>
- [93] <http://www.tutorialspoint.com/jdbc/>
- [94] <http://www.w3schools.com/sql/>
- [95] <http://www.tutorialspoint.com/sql/>
- [96] <http://www.tutorialspoint.com/mysql/>
- [97] <http://www.tutorialspoint.com/jsp/>

ΠΑΡΑΡΤΗΜΑ ΙΙ

ΟΔΗΓΟΣ ΕΚΤΕΛΕΣΗΣ ΚΩΔΙΚΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Για να εκτελέσουμε τον κώδικα της εφαρμογής Διαδικτύου η οποία υλοποιείται στα πλαίσια τούτης της πτυχιακής εργασίας, στον Ηλεκτρονικό Υπολογιστή (H/Y) μας ((Σταθερό ή Φορητό) ο οποίος χρησιμοποιεί την πλατφόρμα λογισμικού “Microsoft Windows 7, Ultimate Edition, x64, Hellenic” (ανάλογα εκτελείται ο προαναφερόμενος κώδικας και σε οποιαδήποτε άλλη έκδοση της συγκεκριμένης πλατφόρμας ή ακόμα και σε διαφορετική συμβατή, όμως, πλατφόρμα λογισμικού), πρέπει να πραγματοποιήσουμε ένα προς ένα τα ακόλουθα βήματα:

- I. Να εγκαταστήσουμε στον τοπικό δίσκο του H/Y μας την τελευταία 64μπιτη έκδοση του προγράμματος “Java Development Kit (JDK)”⁽¹⁾.
- II. Να εγκαταστήσουμε στον τοπικό δίσκο του H/Y μας την τελευταία έκδοση του λογισμικού “XAMPP”⁽²⁾. Εφόσον, εγκαταστήσουμε το εν λόγω πρόγραμμα, ανοίγουμε τη διεπαφή του (ή τρέχοντας το εκτελέσιμο αρχείο του από τον εκάστοτε φάκελο εγκατάστασής του, ή από το μενού “Εναρξη” των Windows, ή από πιθανή συντόμευση του στην επιφάνεια εργασίας του υπολογιστή μας, ή και αυτόματα μετά την εγκατάστασή του). Στη συνέχεια από τη διεπαφή του προαναφερόμενου προγράμματος εκκινούμε τις υπηρεσίες “Apache” και “MySQL”, πατώντας στο κουμπί “Start”, το οποίο υπάρχει δίπλα σε καθεμία από αυτές τις υπηρεσίες (δεν έχει σημασία η σειρά εκκίνησής τους). Κατόπιν, αφότου εκκινήθούν οι προηγούμενες υπηρεσίες (στο σημείο αυτό να τονίσουμε ότι οι πόρτες που χρησιμοποιούν οι συγκεκριμένες υπηρεσίες δεν θα πρέπει να είναι δεσμευμένες από άλλα προγράμματα που τρέχουν παράλληλα και ακούν στις ίδιες πόρτες), πατάμε στο κουμπί “Admin”, το οποίο είναι στα δεξιά της υπηρεσίας “MySQL”. Τότε ανοίγει ένα παράθυρο στο προεπιλεγμένο πρόγραμμά μας περιήγησης στο Διαδίκτυο (καλό θα είναι αυτό το πρόγραμμα να είναι η τελευταία έκδοση του Mozilla Firefox). Στο αριστερό πλαίσιο του πρωτύπου παραθύρου (το οποίο φέρει το λογότυπο “phpMyadmin”), επιλέγουμε την ετικέτα «Νέα» και ύστερα στο δεξιό του πλαίσιο, εισάγουμε στο πεδίο κειμένου, το οποίο υφίσταται κάτω ακριβώς από την ετικέτα «Δημιουργία βάσης δεδομένων», το όνομα “e-eyes”, το οποίο αντιστοιχεί στη βάση δεδομένων με την οποία θα επικοινωνεί η εφαρμογή μας. Μετά κάνουμε κλικ στο κουμπί «Δημιουργία», το οποίο με τη σειρά του είναι τοποθετημένο στα δεξιά του προηγούμενου πεδίου. Έτσι, δημιουργούμε τη βάση δεδομένων που θα χρησιμοποιήσουμε στην εφαρμογή μας (και η οποία, όπως μόλις προαναφέραμε, θα λέγεται “e-eyes”). Ακολούθως, αφότου επιλέξουμε τη βάση που μόλις δημιουργήσαμε, δηλαδή την “e-eyes”, πιέζουμε το κουμπί «Εισαγωγή», το οποίο υπάρχει στο δεξιό πλαίσιο του παραθύρου που έχουμε μπροστά μας, προκειμένου να εισάγουμε τις απαραίτητες εγγραφές και πεδία των επιμέρους βάσεων που περιγράφουμε τις ανάγκες του λογισμικού μας. Τότε, στο νέο δεξιό πλαίσιο που εμφανίζεται, πατάμε στο κουμπί “Αναζήτηση...”, και αφού πλοηγηθούμε στη βάση “eyes.sql” (CD→Back End→Project→Eclipse→e-eyes.sql), κλικάρουμε το κουμπί «Ανοιγμα», ώστε να φορτώσουμε τη συγκεκριμένη βάση και κατόπιν το κουμπί «Εκτέλεση», το οποίο βρίσκεται στο κάτω μέρος το τρέχοντος πλαισίου, για να την εκτελέσουμε. Αν όλα γίνουν όπως τα περιγράψαμε παραπάνω, τότε θα εμφανιστεί στο δεξιό πλαίσιο του παρόντος παραθύρου το μήνυμα επιβεβαίωσης: «Η εισαγωγή ολοκληρώθηκε επιτυχώς, 16 ερωτήματα εκτελέστηκαν (e-eyes.sql)» και στο αριστερό πλαίσιο του τρέχοντος παραθύρου θα είμαστε σε θέση να δούμε τις επιμέρους βάσεις από τις οποίες αποτελείται η προαναφερόμενη βάση “e-eyes” και οι οποίες ανταποκρίνονται στις απαιτήσεις του συστήματός μας.
- III. Από το συνοδευτικό CD της παρούσας πτυχιακής εργασίας θα εκτελέσουμε το αρχείο “eclipse.exe” (CD→Back End→eclipse-jee-luna-R-win32-x86_64→eclipse→eclipse.exe). Στη συνέχεια, μόλις φορτώσει η διεπαφή της πλατφόρμας λογισμικού “Eclipse”, θα εισάγουμε το Project το οποίο περιέχει τον κώδικα της εφαρμογής μας και φέρει το όνομα “MyBriger”: File→Import→Existing Projects into Workspace→Next→Select Root Directory→Browse...→CD→Back End→Project→Eclipse→Finish. Έπειτα, εφόσον, φορτώσει το προαναφερόμενο

Project, θα πρέπει να εκκινήσουμε το διακομιστή Ιστού Apache Tomcat, τον οποίο χρησιμοποιεί το σύστημά μας. Για να επιτευχθεί κάτι τέτοιο, θα πρέπει στο παράθυρο Servers (Window→Show View→Servers) του “Eclipse” να εκκινήσουμε τον Tomcat (Δεξί κλικ→Start).

- IV. Ανοίγουμε μία νέα καρτέλα στον Firefox (προτείνεται) και πληκτρολογούμε τη διεύθυνση URL:<http://localhost:8080/MyBriger/>, οπότε, αν όλα έχουν πραγματοποιηθεί σύμφωνα με τις προαναφερόμενες οδηγίες, θα αναδυθεί στην οθόνη μας το αρχικό παράθυρο της εφαρμογής μας.

⁽¹⁾ Διαθέσιμο στο URL:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

⁽²⁾ Διαθέσιμο στο URL:

https://www.apachefriends.org/download_success.html

ΠΑΡΑΡΤΗΜΑ ΙΙΙ

ΠΑΡΟΥΣΙΑΣΗ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Κρήτης



Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής

Πτυχιακή Εργασία

Τίτλος: «Ανάπτυξη λογισμικού για την παρακολούθηση ασθενών οφθαλμολογικής κλινικής.»

Εισηγητής: Χατζάκης Ηλίας

Σπουδαστής: Ζιώγας Παναγιώτης (Α.Μ. 679)

1

Σκοπός Πτυχιακής Εργασίας

Ανάπτυξη πληροφοριακού συστήματος Διαδικτύου που θα επιτρέπει την αλληλεπίδραση μεταξύ των διαχειριστών αυτού του συστήματος, των ιατρών οι οποίοι είναι συμβεβλημένοι με ένα ή περισσότερα από τα Ασφαλιστικά Ταμεία μίας χώρας της εν λόγω χώρας, των οφθαλμολογικών κλινικών, είτε δημοσίων είτε ιδιωτικών, οι οποίες, επίσης, συνεργάζονται με τα παραπάνω Ταμεία, με απώτερο σκοπό και στόχο την όσο είναι εφικτό καλύτερη επίβλεψη της υγείας των ασθενών-ασφαλισμένων της εν λόγω χώρας.

2

Εισαγωγή – Γενικές Έννοιες

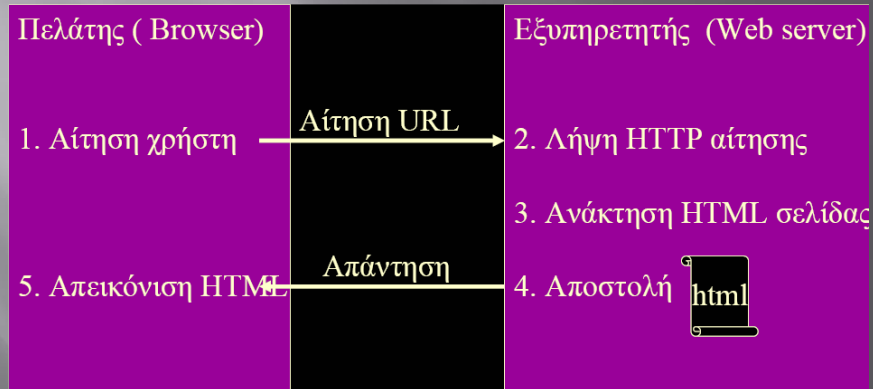
3

Διακομιστής Ιστού (Web Server)

- ▣ Είναι ένα πρόγραμμα που «σερβίρει» HTTP Σελίδες, δέχεται δηλαδή HTTP αιτήσεις και επιστρέφει ως απάντηση HTML αρχεία.
- ▣ Πρόκειται για λογισμικό που συνήθως απαντάει όταν καλείται μια διεύθυνση URL σε ένα φυλλομετρητή (browser).
- ▣ Οι δημοφιλέστεροι είναι ο Apache Web Server (της Apache Foundation) και ο IIS (της Microsoft).

4

Βήματα Αίτησης-Απόκρισης



5

Στατικές & Δυναμικές Σελίδες

- ▣ **Στατικές Σελίδες**
 - Στατική είναι μία Σελίδα που το περιεχόμενό της **δεν** αλλάζει από σύνοδο σε σύνοδο (session).
 - Το μοντέλο αυτό ήταν αρκετό και κυρίαρχο στο παρελθόν.
- ▣ **Δυναμικές Σελίδες**
 - Το περιεχόμενο των δυναμικών Σελίδων μπορεί να αλλάζει σε κάθε σύνοδο ή σε κάθε αίτηση.
 - Καθώς οι ανάγκες και οι υπηρεσίες στο Διαδίκτυο πλήθαιναν και οι τοπικές βάσεις δεδομένων μεταφέρονταν, για να υποστηρίξουν τις υπηρεσίες, δημιουργήθηκε η ανάγκη δυναμικών Σελίδων.
 - **Παραδείγματα:** *JSP, ASP, PHP, ...*

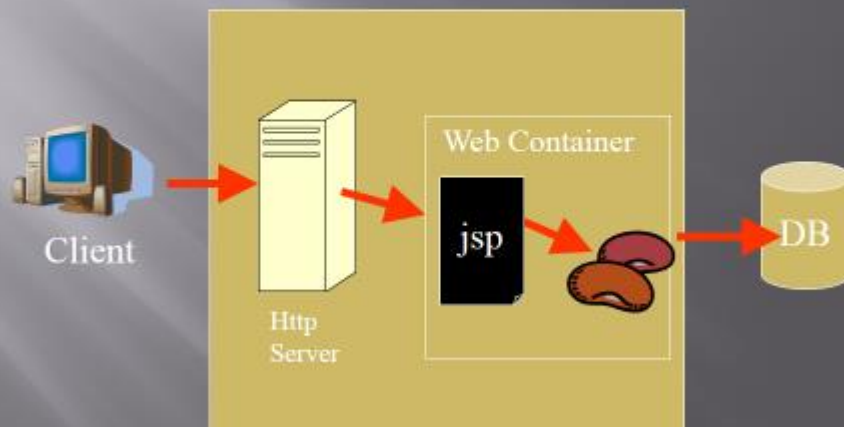
6

Java Server Pages (JSP)

- Είναι HTML Σελίδες που περιέχουν εκτός των HTML ετικετών και ειδικές ετικέτες με Java κώδικα, οι οποίες συνήθως έχουν την μορφή:
`<% Java code %>`
- Περιέχουν:
 - Πρότυπα δεδομένα.
 - Δυναμική πληροφορία.
- Μπορούν να κάνουν χρήση:
 - JavaBeans:
 - Προτυποποιημένες, σχετικά με το σχεδιασμό και την ονοματολογία τους, Java κλάσεις οι οποίες χρησιμοποιούνται για την υλοποίηση συγκεκριμένων λειτουργιών (πχ επικοινωνία με μια βάση δεδομένων).
 - Tag Libraries:
 - Συλλογές χρήσιμων JSP ετικετών οι οποίες περιέχουν βασικές λειτουργίες, κοινές σε πολλές JSP εφαρμογές. Δύναται να τις ορίσει ο εκάστοτε προγραμματιστής και είναι ιδιαίτερα χρήσιμες για επαναχρησιμοποιήσιμα τμήματα κώδικα.

7

Λειτουργία JSP



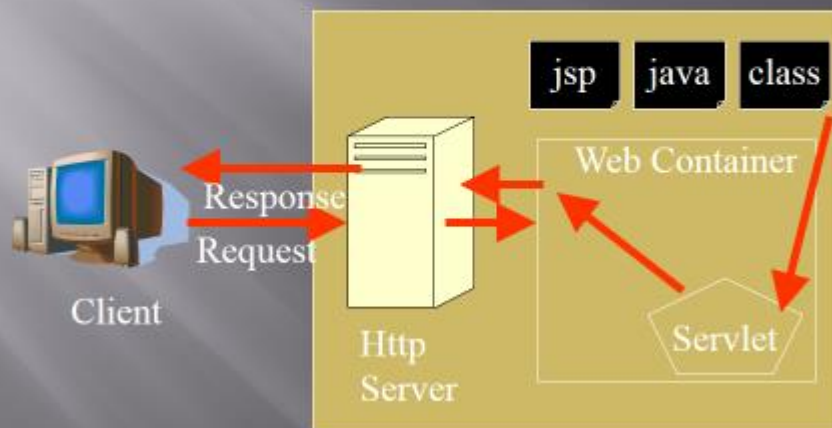
8

Java Servlets

- ▣ Είναι ενότητες κώδικα Java (Java Classes), οι οποίες τρέχουν σε Web servers και δημιουργούν δυναμικό περιεχόμενο (συνήθως HTML).
- ▣ Χρησιμοποιούνται σε εφαρμογές που απαιτούν προεπεξεργασία, πριν την αποστολή απόκρισης στους χρήστες.
- ▣ Κληρονομούν όλα τα πλεονεκτήματα που προσφέρει η Java.
- ▣ Αποτελούν το Server Side πρόσωπο της Java, σε αντίθεση με τα Applets που αποτελούν το Client Side πρόσωπο της Java.

9

Λειτουργία Java Servlet



10

Συνοψίζοντας...

- ☐ Στις Java Servlets:
 - το παραγόμενο HTML «εκτυπώνεται» από Java κλάσεις



- ☐ Στις JSP:
 - Java κώδικας εμπεριέχεται σε HTML κώδικα.



Πρακτικά όμως, τα JSP πριν εκτελεστούν από έναν Web server μετατρέπονται (compile) σε Java Servlets!!!

11

JavaBeans

- ☐ Είναι Java κλάσεις με συγκεκριμένο format:

```
public class PersonBean {  
    private String name;  
    private boolean deceased;  
    public PersonBean() {  
    }  
    public String getName() {  
        return this.name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public boolean isDeceased() {  
        return this.deceased;  
    }  
    public void setDeceased(boolean deceased) {  
        this.deceased = deceased;  
    }  
}
```

- ☐ Καλούνται εύκολα μέσα τις JSP...

12

JavaBeans & JSP

- Αρχικά δηλώνεται η κλάση που αντιστοιχεί στον εκάστοτε bean, στον οποίο αποδίδεται ένα όνομα (id), ώστε να μπορεί να καλεστεί μέσα από τον JSP κώδικα μίας Σελίδας.
 - `<jsp:useBean id="myBean" class="PersonBean" scope="page" />`
- Για να καλεστεί κάποιο "getter":
 - `<jsp:getProperty name="myBean" property="name" />`
- Για να καλεστεί κάποιο "setter":
 - `<jsp:setProperty name="myBean" property="name" value="Someone" />`
 - Ορίζει συγκεκριμένη τιμή για μία παράμετρο.
 - `<jsp:setProperty name="myBean" property="name" />`
 - Ορίζει αυτόματα για μία παράμετρο την τιμή που έχει η παράμετρος του request με το ίδιο όνομα.
 - `<jsp:setProperty name="myBean" property="*" />`
 - Ορίζει αυτόματα για όλες τις παραμέτρους την τιμή που έχουν οι παράμετροι του request με το ίδιο όνομα.

13

ΠΑΡΟΥΣΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

14

Αρχική Σελίδα Λογισμικού

Εθνικό Σύστημα Υγείας (ΕΣΥ) - Οφθαλμιατρικές Εξετάσεις

Ιατρός	Νοσοκομείο	Διαχειριστής
ΟΝΟΜΑ ΧΡΗΣΤΗ <input type="text"/>	ΟΝΟΜΑ ΧΡΗΣΤΗ <input type="text"/>	ΟΝΟΜΑ ΧΡΗΣΤΗ <input type="text"/>
ΚΩΔΙΚΟΣ <input type="text"/>	ΚΩΔΙΚΟΣ <input type="text"/>	ΚΩΔΙΚΟΣ <input type="text"/>
<input type="button" value="ΕΙΣΟΔΟΣ"/>	<input type="button" value="ΕΙΣΟΔΟΣ"/>	<input type="button" value="ΕΙΣΟΔΟΣ"/>

15

ΧΡΗΣΤΕΣ-ΑΝΕΠΙΤΥΧΗΣ ΕΙΣΟΔΟΣ

Το όνομα χρήστη ή ο κωδικός που πληκτρολογήσατε δε βρέθηκε!!!

16

Αρχικό Παράθυρο Διαχειριστή

Διαχείριση Πληροφοριών

Εγγραφή Ιατρού

Εγγραφή Νοσοκομείου

Εγγραφή Ασθενούς

ΑΠΟΣΥΝΔΕΣΗ

17

Διαχειριστής-Φόρμα Εγγραφής Ιατρού

Φόρμα Εγγραφής Ιατρού

Όνομα Χρήστη		[Πληκτρολογήστε το όνομα χρήστη του ιατρού, το οποίο πρέπει να περιέχει από 5 έως 16 μη λευκούς χαρακτήρες...]
Κωδικός		[Πληκτρολογήστε τον κωδικό του ιατρού, ο οποίος πρέπει να περιέχει από 6 έως 14 μη λευκούς χαρακτήρες...]
Επαλήθευση Κωδικού		[Πληκτρολογήστε ξανά τον κωδικό του ιατρού...]
Όνομα		[Πληκτρολογήστε το μικρό όνομα του ιατρού, το οποίο πρέπει να περιέχει από 2 έως 25 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών ονομάτων!!)]
Επώνυμο		[Πληκτρολογήστε το επίθετο του ιατρού, το οποίο πρέπει να περιέχει από 2 έως 35 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών επωνύμων!!!)]
Φύλο	<input type="radio"/>	[Επιλέξτε το φύλο του ιατρού: Άρρεν ή θήλυ...]
Ειδικότητα		[Πληκτρολογήστε την ειδικότητα του ιατρού, η οποία πρέπει να περιέχει από 2 έως 100 αλφαβητικούς χαρακτήρες ή λευκούς χαρακτήρες (εκτός από τον πρώτο χαρακτήρα) ή ένα ή περισσότερα από τα σύμβολα "-", " ", "(", ")", "...]
Κινητό		[Πληκτρολογήστε το κινητό τηλέφωνο του ιατρού, το οποίο πρέπει να περιέχει ακριβώς 10 ψηφία...]

18

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Ιατρού- 1^η Περίπτωση

Ανεπιτυχής καταχώρηση ιατρού: Το όνομα χρήστη του ιατρού το οποίο πληκτρολογήσατε χρησιμοποιείται ήδη!!!

Διακομίστε ένα

19

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Ιατρού- 2^η Περίπτωση

Ανεπιτυχής καταχώρηση ιατρού: Το κινητό τηλέφωνο του ιατρού το οποίο πληκτρολογήσατε χρησιμοποιείται ήδη!!!

Διακομίστε ένα

20

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Ιατρού- 3^η Περίπτωση

Ανεπιτυχής καταχώρηση ιατρού: Τόσο το όνομα χρήστη του ιατρού όσο και το κινητό του τηλέφωνο τα οποία πληκτρολογήσατε χρησιμοποιούνται ήδη!!!

Παρατήρησε

21

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Ιατρού- 4^η Περίπτωση

Επιτυχής καταχώρηση ιατρού!!!

Εντάξει

22

Διαχειριστής-Φόρμα Εγγραφής Νοσοκομείου

Φόρμα Εγγραφής Νοσοκομείου

Όνομα χρήστη	<input type="text"/>	[Πληκτρολογήστε το όνομα χρήστη του νοσοκομείου, το οποίο πρέπει να περιέχει από 5 έως 16 μη λευκούς χαρακτήρες...]
Κωδικός	<input type="text"/>	[Πληκτρολογήστε τον κωδικό του νοσοκομείου, ο οποίος πρέπει να περιέχει από 6 έως 14 μη λευκούς χαρακτήρες...]
Επαλήθευση Κωδικού	<input type="text"/>	[Πληκτρολογήστε ξανά τον κωδικό του νοσοκομείου...]
Όνομασία	<input type="text"/>	[Πληκτρολογήστε το όνομα του νοσοκομείου, το οποίο πρέπει να περιέχει από 2 έως 100 αλφαβητικούς ή αριθμητικούς ή λευκούς χαρακτήρες (εκτός από τον πρώτο χαρακτήρα) ή ένα ή περισσότερα από τα σύμβολα "-", " ", ".", "(", ")", "!" ή διπλά εισαγωγικά...]
Διεύθυνση	<input type="text"/>	[Πληκτρολογήστε την πλήρη διεύθυνση του νοσοκομείου, η οποία πρέπει να περιέχει από 10 έως και 120 αλφαβητικούς ή αριθμητικούς ή λευκούς χαρακτήρες (εκτός από τον 1ο χαρακτήρα) ή ένα ή περισσότερα από τα σύμβολα "-", " ", ".", "(", ")", "!" ...]

23

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Νοσοκομείου- 1^η Περίπτωση

Ανεπιτυχής καταχώρηση νοσοκομείου: Το όνομα χρήστη του νοσοκομείου που πληκτρολογήσατε χρησιμοποιείται ήδη!!!

24

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Νοσοκομείου- 2^η Περίπτωση

Ανεπιτυχής καταχώρηση νοσοκομείου: Η ονομασία του νοσοκομείου την οποία πληκτρολογήσατε χρησιμοποιείται ήδη!!!

[Παραίτηση](#)

25

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Νοσοκομείου- 3^η Περίπτωση

Ανεπιτυχής καταχώρηση νοσοκομείου: Τόσο το όνομα χρήστη του νοσοκομείου όσο και η ονομασία του που πληκτρολογήσατε χρησιμοποιούνται ήδη!!!

[Παραίτηση](#)

26

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Νοσοκομείου- 4^η Περίπτωση

Επιτυχής καταχώρηση νοσοκομείου!!!

Εντάξει

27

Διαχειριστής-Φόρμα Εγγραφής Ασθενούς

Φόρμα Εγγραφής Ασθενούς

ΑΜΚΑ	<input type="text"/>	[Πληκτρολογήστε τον Αριθμό Μητρώου Κοινωνικής Ασφάλισης (ΑΜΚΑ) του ασθενούς, ο οποίος πρέπει να περιέχει ακριβώς 11 ψηφία...]
Ασφάλεια	<input type="text"/>	[Πληκτρολογήστε το Ασφαλιστικό ταμείο του ασθενούς, το οποίο πρέπει να περιέχει από 2 έως 8 αποκλειστικά αλφαβητικούς χαρακτήρες...]
Όνομα	<input type="text"/>	[Πληκτρολογήστε το μικρό όνομα του ασθενούς, το οποίο πρέπει να περιέχει από 2 έως 25 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών ονομάτων!!!)]
Επώνυμο	<input type="text"/>	[Πληκτρολογήστε το επίθετο του ασθενούς, το οποίο πρέπει να περιέχει από 2 έως 35 αποκλειστικά αλφαβητικούς χαρακτήρες... (Απαγορεύεται η χρήση διπλών επωνύμων!!!)]
Φύλο	<input type="radio"/> <input type="radio"/>	[Επιλέξτε το φύλο του ασθενούς: Άρρεν ή Θήλυ...]
Τηλέφωνο	<input type="text"/>	[Πληκτρολογήστε το σταθερό ή το κινητό τηλέφωνο του ασθενούς, το οποίο πρέπει να περιέχει ακριβώς 10 ψηφία...]
E-Mail	<input type="text"/>	[Πληκτρολογήστε μία έγκυρη διεύθυνση ηλεκτρονικού ταχυδρομείου (e-mail) του ασθενούς...]

28

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Ασθενούς- 1^η Περίπτωση

Ανεπιτυχής καταχώρηση ασθενούς: Ο ΑΜΚΑ του ασθενούς που πληκτρολογήσατε χρησιμοποιείται ήδη!!!

Δοκιμάστε ξανά

29

Διαχειριστής-Υποβολή Φόρμας Εγγραφής Ασθενούς- 2^η Περίπτωση

Επιτυχής καταχώρηση ασθενούς!!!

Εντάξει

30

Αρχικό Παράθυρο Ιατρού

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αρχικό Παράθυρο Ιατρού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Παρακαλούμε επιλέξτε μια από τις κατηγορίες...!!!</p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p>Αποσύνδεση</p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

31

Ιατρός-Σύνταξη Παραπεμπτικού

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Σύνταξη Παραπεμπτικού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Προκειμένου να συντάξετε ένα νέο παραπεμπτικό, παρακαλούμε συμπληρώστε κατ'άλληλα τα κενά πεδία που ακολουθούν και κατόπιν πατήστε το κουμπί "Υποβολή" στο τέλος της τρέχουσας φόρμας:</p> <p>ID Παραπεμπτικού: <input type="text" value="4377946"/></p> <p>ID Ιατρού: <input type="text" value="1"/></p> <p>Όνομα Ιατρού: <input type="text" value="Νικόλαος"/></p> <p>Επώνυμο Ιατρού: <input type="text" value="Παπαδόπουλος"/></p> <p>ΑΜΚΑ Ασθενούς: <input type="text"/></p> <p>Ασφάλεια Ασθενούς: <input type="text"/></p> <p>Νοσοκομείο: <input type="text" value=""/></p> <p>Αιτία Παραπομπής: <input type="text"/></p> <p>Υποβολή</p> <p><i>ΠΡΟΣΟΧΗ!!! Παρακαλούμε κάθε φορά, εφόσον συντάξετε ένα παραπεμπτικό, αποσυνδεθείτε, πριν προχωρήσετε στη σύνταξη καινούργιο, για να λάβετε νέο ID παραπεμπτικό και να είστε έτσι σε θέση να συντάξετε καινούργιο παραπεμπτικό!!!</i></p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p>Αποσύνδεση</p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

32

Ιατρός-Υποβολή Παραπεμπτικού- 1^η Περίπτωση

Ανεπιτυχής καταχώρηση παραπεμπτικού: Το ID του παραπεμπτικού που συντάξατε χρησιμοποιείται ήδη!!! Παρακαλούμε, αποσυνδεθείτε, για να λάβετε νέο ID παραπεμπτικού!!!

Αποσύνδεση

33

Ιατρός-Υποβολή Παραπεμπτικού- 2^η Περίπτωση

Ανεπιτυχής καταχώρηση παραπεμπτικού: Ο ΑΜΚΑ που πληκτρολογήσατε δεν αντιστοιχεί σε κάποιον ασθενή!!!

Δοκιμάστε ξανά

34

Ιατρός-Υποβολή Παραπεμπτικού- 3^η Περίπτωση

Ανεπιτυχής καταχώρηση παραπεμπτικού: Ο ασθενής με τον ΑΜΚΑ που πληκτρολογήσατε δεν είναι συμβεβλημένος με το Ασφαλιστικό Ταμείο το οποίο πληκτρολογήσατε!!!

Διαγράψτε Ξανά

35

Ιατρός-Υποβολή Παραπεμπτικού- 4^η Περίπτωση

Επιτυχής καταχώρηση παραπεμπτικού!!!

Εντάξει

36

Ιατρός-Αναζήτηση Παραπεμπτικού

Καλώς ήρθατε στο Σύστημα Οφθαλματριών Εξετάσεων

Κατηγορίες	Στοιχεία Παραπεμπτικού	Χρήστης
Σύνταξη Παραπεμπτικού	Προκειμένου να αναζητήσετε τα στοιχεία ενός παραπεμπτικού, το οποίο έχει συνταχθεί είτε από εσάς είτε από οποιονδήποτε άλλον ιατρό, παρακαλούμε πληκτρολογήστε το ID του στο αμέσως επόμενο πλαίσιο κειμένου: <input type="text"/> και κατόπιν πατήστε στο κουμπί "Αναζήτηση": <input type="button" value="Αναζήτηση"/>	Κωδικός Ιατρού: 1
Στοιχεία Παραπεμπτικού		Ονοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος
Ιστορικό Ασθενή		Ειδικότητα Ιατρού: Οφθαλμίατρος
Ιστορικό Ιατρού		<input type="button" value="Αποσύρηση"/>

ΑΓΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΩΓΙΑΣ (ΑΜ 679)

37

Ιατρός-Ανεπιτυχής Αναζήτηση Παραπεμπτικού

Δε βρέθηκε παραπεμπτικό με το ID που πληκτρολογήσατε!!!

38

Ιατρός-Εμφάνιση Παραπεμπτικού- 1^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλματριών Εξετάσεων

Κατηγορίες	Στοιχεία Παραπεμπτικού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Τα στοιχεία του παραπεμπτικού με ID 4377946 είναι τα ακόλουθα:</p> <p>ID Παραπεμπτικό: 4377946</p> <p>ID Ιατρού: 1</p> <p>Όνομα Ιατρού: Νικόλαος</p> <p>Επώνυμο Ιατρού: Παπαδόπουλος</p> <p>ΑΜΚΑ Ασθενούς: 01014025894</p> <p>Ασφάλεια Ασθενούς: ΟΓΑ</p> <p>ID Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p>Αιτία Παραπομπής: <input type="text" value="Κεφαλίση χωρίς καλίλας."/></p> <p><input type="button" value="Εντάξη"/></p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p><input type="button" value="Αποσύνδεση"/></p>

ΑΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΥΓΓΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

39

Ιατρός-Εμφάνιση Παραπεμπτικού- 2^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλματριών Εξετάσεων

Κατηγορίες	Στοιχεία Παραπεμπτικού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Τα στοιχεία του παραπεμπτικού με ID 2213454 είναι τα ακόλουθα:</p> <p>ID Παραπεμπτικό: 2213454</p> <p>ID Ιατρού: 3</p> <p>Όνομα Ιατρού: Παναγιώτης</p> <p>Επώνυμο Ιατρού: Νικαφόρου</p> <p>ΑΜΚΑ Ασθενούς: 18729641223</p> <p>Ασφάλεια Ασθενούς: ΔΗΜΟΣΙΟ</p> <p>ID Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p>Αιτία Παραπομπής: <input type="text" value="Δοκίμηση οπτικού."/></p> <p><input type="button" value="Εντάξη"/></p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p><input type="button" value="Αποσύνδεση"/></p>

ΑΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΥΓΓΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

40

Ιατρός-Αναζήτηση Ιστορικού Ασθενή

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενούς	Χρήτης
Σύνταξη Παρακερμικού	Προκειμένου να αναζητήσετε το ιστορικό της οφθαλμολογικής υγείας ενός ασθενούς, παρακαλούμε πληκτρολογήστε τον ΑΜΚΑ του στο αμέσως επόμενο πλαίσιο κειμένου: <input type="text"/> και κατόπιν πατήστε στο κουμπί "Αναζήτηση": <input type="button" value="Αναζήτηση"/>	Κωδικός Ιατρού: <input type="text"/>
Στοιχεία Παρακερμικού		Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος
Ιστορικό Ασθενούς		Ειδικότητα Ιατρού: Οφθαλμίατρος
Ιστορικό Ιατρού		<input type="button" value="Αποστολή"/>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

41

Ιατρός-Ανεπιτυχής Αναζήτηση Ιστορικού Ασθενή

Ο ΑΜΚΑ που πληκτρολογήσατε δεν αντιστοιχεί σε κάποιον ασθενή!!!

42

Ιατρός-Εμφάνιση Ιστορικού Ασθενή- 1^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενούς	Χρήστης																														
Σύνταξη Παραπεμπτικού	<p>Το ιστορικό του ασθενούς με ΑΜΚΑ 01014025894 φαίνεται στον ακόλουθο πίνακα:</p> <table border="1"> <thead> <tr> <th>ΑΜΚΑ Ασθενούς</th> <th>Ασφάλεια Ασθενούς</th> <th>Αιτία Παραπομπής</th> <th>Όνομασία Νοσοκομείου</th> <th>Ημερομηνία εξέτασης</th> <th>Αποτέλεσμα εξέτασης</th> </tr> </thead> <tbody> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Μυωπία.</td> <td>Υγεία</td> <td>02/02/1980</td> <td>Ανίχνευση με χαρτοφργείο.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Καταρράκτης.</td> <td>Υγεία</td> <td>12/03/2012</td> <td>Ανίχνευση με λέιζερ.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Πίση.</td> <td>ΙΑΣΩ Θεσσαλίας</td> <td>14/05/2015</td> <td>Φαρμακευτική αγωγή.</td> </tr> <tr> <td>01014025894</td> <td>ΟΓΑ</td> <td>Εκφύλιση αγγείων κηλίδας.</td> <td>Υγεία</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταγραφεί!!!</td> </tr> </tbody> </table>	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Αιτία Παραπομπής	Όνομασία Νοσοκομείου	Ημερομηνία εξέτασης	Αποτέλεσμα εξέτασης	01014025894	ΟΓΑ	Μυωπία.	Υγεία	02/02/1980	Ανίχνευση με χαρτοφργείο.	01014025894	ΟΓΑ	Καταρράκτης.	Υγεία	12/03/2012	Ανίχνευση με λέιζερ.	01014025894	ΟΓΑ	Πίση.	ΙΑΣΩ Θεσσαλίας	14/05/2015	Φαρμακευτική αγωγή.	01014025894	ΟΓΑ	Εκφύλιση αγγείων κηλίδας.	Υγεία	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!	Κωδικός Ιατρού: 1
ΑΜΚΑ Ασθενούς		Ασφάλεια Ασθενούς	Αιτία Παραπομπής	Όνομασία Νοσοκομείου	Ημερομηνία εξέτασης	Αποτέλεσμα εξέτασης																										
01014025894		ΟΓΑ	Μυωπία.	Υγεία	02/02/1980	Ανίχνευση με χαρτοφργείο.																										
01014025894		ΟΓΑ	Καταρράκτης.	Υγεία	12/03/2012	Ανίχνευση με λέιζερ.																										
01014025894		ΟΓΑ	Πίση.	ΙΑΣΩ Θεσσαλίας	14/05/2015	Φαρμακευτική αγωγή.																										
01014025894	ΟΓΑ	Εκφύλιση αγγείων κηλίδας.	Υγεία	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!																											
Στοιχεία Παραπεμπτικού		Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος																														
Ιστορικό Ασθενούς		Ειδικότητα Ιατρού: Οφθαλμίατρος																														
Ιστορικό Ιατρού		Αποσύνδεση																														

ΛΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

43

Ιατρός-Εμφάνιση Ιστορικού Ασθενή- 2^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενούς	Χρήστης
Σύνταξη Παραπεμπτικού	<p>Για τον ασθενή με ΑΜΚΑ 04055536363 δεν έχει συνταχθεί κανένα παραπεμπτικό και συνεπώς δεν έχει πραγματοποιήσει καμία εξέταση!!!</p>	Κωδικός Ιατρού: 1
Στοιχεία Παραπεμπτικού		Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος
Ιστορικό Ασθενούς		Ειδικότητα Ιατρού: Οφθαλμίατρος
Ιστορικό Ιατρού		Αποσύνδεση

ΛΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

44

Ιατρός-Αναζήτηση Ιστορικού Ιδίου

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ιατρού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Προκειμένου να αναζητήσετε το ιστορικό των παραπεμπτικών τα οποία αποκλειστικά εσείς έχετε συντάξει, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;">Αναζήτηση</p>	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p style="text-align: center;">Αποσύνδεση</p>

ΛΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

45

Ιατρός-Εμφάνιση Ιστορικού Ιδίου- 1^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ιατρού	Χρήστης																																				
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Το ιστορικό σας φαίνεται στον ακόλουθο πίνακα:</p> <table border="1"> <thead> <tr> <th>ID Ιατρού</th> <th>ID Παραπεμπτικού</th> <th>ΑΜΚΑ Ασθενούς</th> <th>Ασφάλεια Ασθενούς</th> <th>Όργανο/Νοσοκομείο</th> <th>Αιτία Παραπομπής</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1202016</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Υγεία</td> <td>Μυωπία.</td> </tr> <tr> <td>1</td> <td>4377946</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Υγεία</td> <td>Εκφύλιση αμφύς κηλίδας.</td> </tr> <tr> <td>1</td> <td>6124477</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Υγεία</td> <td>Κερατοκonus.</td> </tr> <tr> <td>1</td> <td>6427958</td> <td>1408321543</td> <td>ΙΚΑ</td> <td>Υγεία</td> <td>Πισση.</td> </tr> <tr> <td>1</td> <td>7627958</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Υγεία</td> <td>Γλαύκωμα.</td> </tr> </tbody> </table> <p style="text-align: center;">Εντάξη</p>	ID Ιατρού	ID Παραπεμπτικού	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Όργανο/Νοσοκομείο	Αιτία Παραπομπής	1	1202016	01014025894	ΟΓΑ	Υγεία	Μυωπία.	1	4377946	01014025894	ΟΓΑ	Υγεία	Εκφύλιση αμφύς κηλίδας.	1	6124477	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Κερατοκonus.	1	6427958	1408321543	ΙΚΑ	Υγεία	Πισση.	1	7627958	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Γλαύκωμα.	<p>Κωδικός Ιατρού: 1</p> <p>Όνοματεπώνυμο Ιατρού: Νικόλαος Παπαδόπουλος</p> <p>Ειδικότητα Ιατρού: Οφθαλμίατρος</p> <p style="text-align: center;">Αποσύνδεση</p>
ID Ιατρού	ID Παραπεμπτικού	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Όργανο/Νοσοκομείο	Αιτία Παραπομπής																																	
1	1202016	01014025894	ΟΓΑ	Υγεία	Μυωπία.																																	
1	4377946	01014025894	ΟΓΑ	Υγεία	Εκφύλιση αμφύς κηλίδας.																																	
1	6124477	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Κερατοκonus.																																	
1	6427958	1408321543	ΙΚΑ	Υγεία	Πισση.																																	
1	7627958	18759641223	ΔΗΜΟΣΙΟ	Υγεία	Γλαύκωμα.																																	

ΛΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

46

Ιατρός-Εμφάνιση Ιστορικού Ιδίου- 2^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ιατρού	Χρήστης
<ul style="list-style-type: none"> Σύνταξη Παραπεμπτικού Στοιχεία Παραπεμπτικού Ιστορικό Ασθενούς Ιστορικό Ιατρού 	<p>Δεν έχετε συντάξει κανένα παραπεμπτικό!!!</p> <p><input type="button" value="Επίστρ"/></p>	<p>Κωδικός Ιατρού: 4</p> <p>Όνοματεπώνυμο Ιατρού: Γεωργιος Καραγιαννης</p> <p>Ειδικότητα Ιατρού: Ειδικός Οφθαλμίατρος</p> <p><input type="button" value="Αποσύνδεση"/></p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΙΛΑΝΑΓΙΩΤΗΣ ΖΩΓΙΑΣ (ΑΜ 679)

47

Αρχικό Παράθυρο Νοσοκομείου

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αρχικό Παράθυρο Νοσοκομείου	Χρήστης
<ul style="list-style-type: none"> Όνομας Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Παρακαλούμε επιλέξτε μία από τις κατηγορίες...!!!</p>	<p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεια</p> <p><input type="button" value="Αποσύνδεση"/></p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΙΛΑΝΑΓΙΩΤΗΣ ΖΩΓΙΑΣ (ΑΜ 679)

48

Νοσοκομείο-Αναζήτηση Εκκρεμούς Παραπεμπτικού ως προς τον Ορισμό Ραντεβού

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ορισμός Ραντεβού	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Προκειμένου να αναζητήσετε τα εκκρεμή παραπεμπτικά ως προς τον ορισμό ραντεβού του νοσοκομείου σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":</p> <p style="text-align: center;">Αναζήτηση</p>	<p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p style="text-align: center;">Αποσύνδεση</p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

49

Νοσοκομείο-Επιλογή Εκκρεμούς Παραπεμπτικού ως προς τον Ορισμό Ραντεβού- 1^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ορισμός Ραντεβού	Χρήστης																														
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Τα παραπεμπτικά σε αναμνήμη ορισμού ραντεβού του νοσοκομείου σας φαίνονται στον ακόλουθο πίνακα:</p> <table border="1"> <thead> <tr> <th>ID Παραπεμπτικό</th> <th>ΑΜΚΑ Ασθενός</th> <th>Ασφάλεια Ασθενός</th> <th>Αιτία Παραπεμπτικής</th> <th>ID Νοσοκομείου</th> <th>Ημερομηνία Εξέτασης</th> </tr> </thead> <tbody> <tr> <td>4377946</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Εκφύλιση ισθμίου κηλόςας.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> <tr> <td>6427958</td> <td>14058321543</td> <td>ΙΚΑ</td> <td>Πίση.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> <tr> <td>7627958</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Γλαύκωμα.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> <tr> <td>8827958</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Εξασθένηση οπτικών νεύρων.</td> <td>10000</td> <td>Δεν έχει οριστεί!!!</td> </tr> </tbody> </table> <p>Προκειμένου να ορίσετε ραντεβού για ένα από τα εκκρεμή παραπεμπτικά του παραπάνω πίνακα, παρακαλούμε πληκτρολογήστε το ID του στο αμέσως επόμενο πλαίσιο κειμένου:</p> <p style="text-align: center;"><input type="text"/></p> <p>και κατόπιν πατήστε στο κουμπί "Επιλογή":</p> <p style="text-align: center;">Επιλογή</p>	ID Παραπεμπτικό	ΑΜΚΑ Ασθενός	Ασφάλεια Ασθενός	Αιτία Παραπεμπτικής	ID Νοσοκομείου	Ημερομηνία Εξέτασης	4377946	01014025894	ΟΓΑ	Εκφύλιση ισθμίου κηλόςας.	10000	Δεν έχει οριστεί!!!	6427958	14058321543	ΙΚΑ	Πίση.	10000	Δεν έχει οριστεί!!!	7627958	18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	10000	Δεν έχει οριστεί!!!	8827958	18759641223	ΔΗΜΟΣΙΟ	Εξασθένηση οπτικών νεύρων.	10000	Δεν έχει οριστεί!!!	<p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υγεία</p> <p style="text-align: center;">Αποσύνδεση</p>
ID Παραπεμπτικό	ΑΜΚΑ Ασθενός	Ασφάλεια Ασθενός	Αιτία Παραπεμπτικής	ID Νοσοκομείου	Ημερομηνία Εξέτασης																											
4377946	01014025894	ΟΓΑ	Εκφύλιση ισθμίου κηλόςας.	10000	Δεν έχει οριστεί!!!																											
6427958	14058321543	ΙΚΑ	Πίση.	10000	Δεν έχει οριστεί!!!																											
7627958	18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	10000	Δεν έχει οριστεί!!!																											
8827958	18759641223	ΔΗΜΟΣΙΟ	Εξασθένηση οπτικών νεύρων.	10000	Δεν έχει οριστεί!!!																											

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

50

Νοσοκομείο-Επιλογή Εκκρεμούς Παραπεμπτικού ως προς τον Ορισμό Ραντεβού- 2^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ορισμός Ραντεβού	Χρήτης
Ορισμός Ραντεβού	Δεν υπάρχουν εκκρεμή παραπεμπτικά ως προς τον ορισμό ραντεβού στο νοσοκομείο σας!!! <input type="button" value="Επιστροφή"/>	Κωδικός Νοσοκομείου: 10001
Αποτέλεσμα Εξέτασης		
Ιστορικό Ασθενούς		
Ιστορικό Νοσοκομείου		
		Όνομασία Νοσοκομείου: ΙΑΣΩ Θεσσαλίας <input type="button" value="Αποδοκιμασία"/>

ΛΥΣΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

51

Νοσοκομείο-Ανεπιτυχήs Επιλογή Εκκρεμούς Παραπεμπτικού ως προς τον Ορισμό Ραντεβού

Το ID που πληκτρολογήσατε δεν αντιστοιχεί σε εκκρεμές παραπεμπτικό ως προς τον ορισμό ραντεβού του νοσοκομείου σας!!!

52

Νοσοκομείο-Ορισμός Ραντεβού

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ορισμός Ραντεβού	Χρήστης
Ορισμός Ραντεβού	<p>Προκειμένου να ορίσετε ραντεβού για το εκκρεμές παραπεμπτικό με ID 4377946 του νοσοκομείου σας, παρακαλούμε συμπληρώστε τόσο την ημερομηνία όσο και την ώρα εξέτασης στα πεδία "Ημερομηνία Εξέτασης" και "Ωρα Εξέτασης"; αντίστοιχα, που ακολουθούν και κατόπιν πατήστε στο κουμπί "Υποβολή" στο τέλος της τρέχουσας φόρμας:</p> <p>ID Παραπεμπτικό: 4377946</p> <p>ID Ιατρού: 1</p> <p>ΑΜΚΑ Ασθενούς: 0104025894</p> <p>Ασφάλεια Ασθενούς: ΟΓΑ</p> <p>Αιτία Παραπομπής: <input type="text" value="Εμφύλιση ισχιάς κνήκιδας."/></p> <p>ID Νοσοκομείου: 10000</p> <p>Ημερομηνία Εξέτασης: <input type="text"/></p> <p>Ωρα Εξέτασης: <input type="text"/></p> <p><input type="button" value="Υποβολή"/></p>	Κωδικός Νοσοκομείου: 10000
Αποτέλεσμα Εξέτασης		Όνοματία Νοσοκομείου: Υγεια
Ιστορικό Ασθενούς		<input type="button" value="Αποσύνδεση"/>
Ιστορικό Νοσοκομείου		

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΗΓΩΓΑΣ (ΑΜ 679)

53

Νοσοκομείο-Υποβολή Ραντεβού- 1^η Περίπτωση

Ανεπιτυχής ορισμός ραντεβού: Το ραντεβού που επιλέξατε δεν είναι διαθέσιμο!!! Παρακαλούμε δοκιμάστε ξανά, επιλέγοντας διαφορετική ώρα ή (και) ημερομηνία εξέτασης!!!

54

Νοσοκομείο-Υποβολή Ραντεβού- 2^η Περίπτωση

Επιτυχής ορισμός ραντεβού!!!

Εντάξει

55

Νοσοκομείο-Αναζήτηση Εκκρεμών Παραπτεμπτικών ως προς την Καταχώρηση Αποτελέσματος Εξέτασης

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αποτέλεσμα Εξέτασης	Χρήστης	
Ορισμός Ραντεβού	Προκειμένου να αναζητήσετε τα εκκρεμή παραπτεμπτικά ως προς την καταχώρηση αποτελέσματος εξέτασης του νοσοκομείου σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση":	Κωδικός Νοσοκομείου: 11000	
Αποτελέσματα Εξέτασης		Όνομασία Νοσοκομείου: Υγεια	
Ιστορικό Λαβών		Αναζήτηση	Αποστολή
Ιστορικό Νοσοκομείου			

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (AM 679)

56

Νοσοκομείο-Επιλογή Εκκρεμούς Παραπεμπτικού ως προς την Καταχώρηση Αποτελέσματος Εξέτασης- 1^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αποτέλεσμα Εξέτασης	Χρήστης																		
Ορισμός Ραντεβού	<p>Τα παραπεμπτικά του νοσοκομείου σας για τα οποία έχει οριστεί ραντεβού, έχει εξεταστεί ο κατάλληλος κάθε φορά ασθενής, αλλά δεν έχει καταχωρηθεί στον ηλεκτρονικό φάκελο του ασθενούς το αποτέλεσμα της εξέτασης του, φαίνονται στον ακόλουθο πίνακα:</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>ID Παραπεμπτικό</th> <th>ΑΜΚΑ Ασθενός</th> <th>Ασφάλεια Ασθενός</th> <th>ID Νοσοκομείου</th> <th>Ημερομηνία Εξέτασης</th> <th>Αποτέλεσμα Εξέτασης</th> </tr> </thead> <tbody> <tr> <td>437946</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>10000</td> <td>10/05/2016</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> <tr> <td>6124477</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>10000</td> <td>12/06/2016</td> <td>Δεν έχει καταχωρηθεί!!!</td> </tr> </tbody> </table> <p>Προκειμένου να καταχωρήσετε το αποτέλεσμα της εξέτασης για ένα από τα εκκρεμή παραπεμπτικά του παραπάνω πίνακα, παρακάτωμε, πληκτρολογήστε το ID του στο αμέσως επόμενο πλαίσιο κειμένου:</p> <div style="text-align: center; margin: 10px 0;"> <input style="width: 50px; height: 20px;" type="text"/> </div> <p>και κατόπιν πατήστε στο κουμπί "Επιλογή":</p> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="Επιλογή"/> </div>	ID Παραπεμπτικό	ΑΜΚΑ Ασθενός	Ασφάλεια Ασθενός	ID Νοσοκομείου	Ημερομηνία Εξέτασης	Αποτέλεσμα Εξέτασης	437946	01014025894	ΟΓΑ	10000	10/05/2016	Δεν έχει καταχωρηθεί!!!	6124477	18759641223	ΔΗΜΟΣΙΟ	10000	12/06/2016	Δεν έχει καταχωρηθεί!!!	Κωδικός Νοσοκομείου: 10000
ID Παραπεμπτικό		ΑΜΚΑ Ασθενός	Ασφάλεια Ασθενός	ID Νοσοκομείου	Ημερομηνία Εξέτασης	Αποτέλεσμα Εξέτασης														
437946		01014025894	ΟΓΑ	10000	10/05/2016	Δεν έχει καταχωρηθεί!!!														
6124477		18759641223	ΔΗΜΟΣΙΟ	10000	12/06/2016	Δεν έχει καταχωρηθεί!!!														
Αποτέλεσμα Εξέτασης		Ονομασία Νοσοκομείου: Υγεία																		
Ιστορικό Ασθενούς	<input type="button" value="Αποσύνδεση"/>																			
Ιστορικό Νοσοκομείου																				

ΑΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΑΣ (ΑΜ 679)

57

Νοσοκομείο-Επιλογή Εκκρεμούς Παραπεμπτικού ως προς την Καταχώρηση Αποτελέσματος Εξέτασης- 2^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αποτέλεσμα Εξέτασης	Χρήστης
Ορισμός Ραντεβού	<p>Δεν υπάρχουν εκκρεμή παραπεμπτικά ως προς την καταχώρηση αποτελέσματος εξέτασης στο νοσοκομείο σας!!!</p> <div style="text-align: center; margin-top: 20px;"> <input type="button" value="Εντάξει"/> </div>	Κωδικός Νοσοκομείου: 10001
Αποτέλεσμα Εξέτασης		Ονομασία Νοσοκομείου: ΑΣΣΠ Φιλοθέης
Ιστορικό Ασθενούς		<input type="button" value="Αποσύνδεση"/>
Ιστορικό Νοσοκομείου		

ΑΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΓΑΣ (ΑΜ 679)

58

Νοσοκομείο-Ανεπιτυχής Επιλογή Εκκρεμούς Παραπεμπτικού ως προς την Καταχώρηση Αποτελέσματος Εξέτασης

Το ID που πληκτρολογήσατε δεν αντιστοιχεί σε εκκρεμές παραπεμπτικό ως προς την καταχώρηση αποτελέσματος εξέτασης του νοσοκομείου σας!!!

[Παραπομπή](#)

59

Νοσοκομείο-Καταχώρηση Αποτελέσματος Εξέτασης

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Αποτέλεσμα Εξέτασης	Χρήστης
<ul style="list-style-type: none"> Ορατός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Προκειμένου να καταχωρήσετε το αποτέλεσμα της εξέτασης για το παραπεμπτικό με ID 4377946 του νοσοκομείου σας, παρακαλούμε καταγράψτε το συγκεκριμένο αποτέλεσμα στο πεδίο "Αποτέλεσμα Εξέτασης" που ακολουθεί και κατόπιν πατήστε στο κουμπί "Υποβολή" στο τέλος της τρέχουσας φόρμας:</p> <p>ID Παραπεμπτικό: <input type="text" value="4377946"/></p> <p>ID Ιατρού: <input type="text" value="1"/></p> <p>ΑΜΚΑ Ασθενούς: <input type="text" value="01014025894"/></p> <p>Ασφάλεια Ασθενούς: <input type="text" value="ΟΓΑ"/></p> <p>ID Νοσοκομείου: <input type="text" value="10001"/></p> <p>Αιτία Παραπομπής: <input type="text" value="Εκπόλιση αχρούς κελύφους."/></p> <p>Ημερομηνία Εξέτασης: <input type="text" value="10/05/2016"/></p> <p>Ωρα Εξέτασης: <input type="text" value="10:30"/></p> <p>Αποτέλεσμα Εξέτασης: <input type="text"/></p> <p>Υποβολή</p>	<p>Κωδικός Νοσοκομείου: <input type="text" value="10001"/></p> <p>Όνομασία Νοσοκομείου: <input type="text" value="ΙΑΣΩ Θεσσαλίας"/></p> <p>Αποσύνδεση</p>

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΟΥΡΑΣ (ΑΜ 679)

60

Νοσοκομείο-Υποβολή Αποτελέσματος Εξέτασης

Επιτυχής καταχώρηση αποτελέσματος εξέτασης!!!

Εντάξει

61

Νοσοκομείο-Αναζήτηση Ιστορικού Ασθενή

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενούς	Χρήστης
Οφθαλμός Παιδιού	Προκειμένου να αναζητήσετε το ιστορικό της οφθαλμολογικής υγείας ενός ασθενούς, παρακαλούμε πληκτρολογήστε τον ΑΜΚΑ του στο αμέσως επόμενο πλαίσιο κειμένου: και κατόπιν πατήστε στο κουμπί "Αναζήτηση":	Κωδικός Νοσοκομείου: 10000
Αποτελέσματα Εξέτασης		Όνομασία Νοσοκομείου: Υγεία
Ιστορικό Ασθενούς	<input type="text"/>	<input type="button" value="Αναζήτηση"/>
Ιστορικό Νοσοκομείου		

ΑΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΙΩΑΝΝΙΝΟΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

62

Νοσοκομείο-Ανεπιτυχής Αναζήτηση Ιστορικού Ασθενή

Ο ΑΜΚΑ που πληκτρολογήσατε δεν αντιστοιχεί σε κάποιον ασθενή!!!

Δοκιμάστε ξανά

63

Νοσοκομείο-Εμφάνιση Ιστορικού Ασθενή- 1^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορία	Ιστορικό Ασθενούς						Χρήστης
Ομάδα Ραντεβού	Το ιστορικό του ασθενούς με ΑΜΚΑ 01014025894 φαίνεται στον ακόλουθο πίνακα:						Κωδικός Νοσοκομείου: 10000
Αποτέλεσμα Εξέτασης	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Αιτία Παραπομπής	Όνομασία Νοσοκομείου	Ημερομηνία εξέτασης	Αποτέλεσμα εξέτασης	Όνομασία Νοσοκομείου: Υγεία
Ιστορικό Ασθενούς	01014025894	ΟΓΑ	Μυωπία	Υγεία	02/02/1980	Αφαίρεση με χειρουργείο.	Αποσύνδεση
Ιστορικό Νοσοκομείου	01014025894	ΟΓΑ	Καταρράκτης	Υγεία	12/03/2012	Αφαίρεση με λέιζερ.	
	01014025894	ΟΓΑ	Πίση	ΙΑΣΩ Θεσσαλονίκη	14/05/2015	Φαρμακευτική αγωγή.	
	01014025894	ΟΓΑ	Εκπίεση σκληρ. κελύφου.	Υγεία	10/05/2016	Ενδοφθάλμια.	
	Επίστ.						

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

64

Νοσοκομείο-Εμφάνιση Ιστορικού Ασθενή- 2^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Ασθενούς	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Για τον ασθενή με ΑΜΚΑ 04055536363 δεν έχει συνταχθεί κάποιο παραπεμπτικό και συνεπώς δεν έχει πραγματοποιήσει κάποια εξέταση!!!</p> <p style="text-align: center;">Επίλεξε</p>	<ul style="list-style-type: none"> Κωδικός Νοσοκομείου: 10000 Όνομαστη Νοσοκομείου: Υγεία Αποσύνδεση

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

65

Νοσοκομείο-Αναζήτηση Ιστορικού Ιδίου

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Νοσοκομείου	Χρήστης
<ul style="list-style-type: none"> Ορισμός Ραντεβού Αποτέλεσμα Εξέτασης Ιστορικό Ασθενούς Ιστορικό Νοσοκομείου 	<p>Προκείμενοι να αναζητήσετε το ιστορικό των οφθαλμολογικών εξετάσεων που έχουν πραγματοποιηθεί ή πρόκειται να πραγματοποιηθούν στο νοσοκομείο σας, παρακαλούμε πατήστε στο κουμπί "Αναζήτηση".</p> <p style="text-align: center;">Αναζήτηση</p>	<ul style="list-style-type: none"> Κωδικός Νοσοκομείου: 10000 Όνομαστη Νοσοκομείου: Υγεία Αποσύνδεση

ΑΤΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

66

Νοσοκομείο-Εμφάνιση Ιστορικού Ιδίου- 1^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Νοσοκομείων	Χρήστης																																																						
Οφθαλμ. Ραντίφθο	<p>Το ιστορικό σας φαίνεται στον ακόλουθο πίνακα:</p> <table border="1"> <thead> <tr> <th>ID Νοσοκομείου</th> <th>ΑΜΚΑ Ασθενούς</th> <th>Ασφάλεια Ασθενούς</th> <th>Αιτία Παραπομπής</th> <th>Ημερομηνία Εξέτασης</th> <th>Αποτελεσμα Εξέτασης</th> </tr> </thead> <tbody> <tr> <td>10000</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Μυωπία.</td> <td>02/02/1980</td> <td>Ανίχνευση με χειροσκόπιο.</td> </tr> <tr> <td>10000</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Καταρράκτης.</td> <td>12/03/2012</td> <td>Ανίχνευση με λήζερ.</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Ασπτηματιοσ.</td> <td>24/11/2013</td> <td>Σύσταση για χρήση φακών επαφής.</td> </tr> <tr> <td>10000</td> <td>01014025894</td> <td>ΟΓΑ</td> <td>Εκφύλιση ωχρούς κηλίδας.</td> <td>10/05/2016</td> <td>Ενσωσφρακτία.</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Κερατόκωνος.</td> <td>12/06/2016</td> <td>Δεν έχει καταγραφεί!!!</td> </tr> <tr> <td>10000</td> <td>14058321543</td> <td>ΙΚΑ</td> <td>Πίση.</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταγραφεί!!!</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Γλαύκωμα.</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταγραφεί!!!</td> </tr> <tr> <td>10000</td> <td>18759641223</td> <td>ΔΗΜΟΣΙΟ</td> <td>Επιθρόνηση οπτικών νεύρων.</td> <td>Δεν έχει οριστεί!!!</td> <td>Δεν έχει καταγραφεί!!!</td> </tr> </tbody> </table>	ID Νοσοκομείου	ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Αιτία Παραπομπής	Ημερομηνία Εξέτασης	Αποτελεσμα Εξέτασης	10000	01014025894	ΟΓΑ	Μυωπία.	02/02/1980	Ανίχνευση με χειροσκόπιο.	10000	01014025894	ΟΓΑ	Καταρράκτης.	12/03/2012	Ανίχνευση με λήζερ.	10000	18759641223	ΔΗΜΟΣΙΟ	Ασπτηματιοσ.	24/11/2013	Σύσταση για χρήση φακών επαφής.	10000	01014025894	ΟΓΑ	Εκφύλιση ωχρούς κηλίδας.	10/05/2016	Ενσωσφρακτία.	10000	18759641223	ΔΗΜΟΣΙΟ	Κερατόκωνος.	12/06/2016	Δεν έχει καταγραφεί!!!	10000	14058321543	ΙΚΑ	Πίση.	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!	10000	18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!	10000	18759641223	ΔΗΜΟΣΙΟ	Επιθρόνηση οπτικών νεύρων.	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!	<p>Κωδικός Νοσοκομείου: 10000</p> <p>Όνομασία Νοσοκομείου: Υπο</p> <p>Αποσύνδεση</p>
ID Νοσοκομείου		ΑΜΚΑ Ασθενούς	Ασφάλεια Ασθενούς	Αιτία Παραπομπής	Ημερομηνία Εξέτασης	Αποτελεσμα Εξέτασης																																																		
10000		01014025894	ΟΓΑ	Μυωπία.	02/02/1980	Ανίχνευση με χειροσκόπιο.																																																		
10000		01014025894	ΟΓΑ	Καταρράκτης.	12/03/2012	Ανίχνευση με λήζερ.																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Ασπτηματιοσ.	24/11/2013	Σύσταση για χρήση φακών επαφής.																																																		
10000		01014025894	ΟΓΑ	Εκφύλιση ωχρούς κηλίδας.	10/05/2016	Ενσωσφρακτία.																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Κερατόκωνος.	12/06/2016	Δεν έχει καταγραφεί!!!																																																		
10000		14058321543	ΙΚΑ	Πίση.	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Γλαύκωμα.	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!																																																		
10000		18759641223	ΔΗΜΟΣΙΟ	Επιθρόνηση οπτικών νεύρων.	Δεν έχει οριστεί!!!	Δεν έχει καταγραφεί!!!																																																		
Αποτελεσμα Εξέτασης																																																								
Ιστορικό Ασθενούς																																																								
Ιστορικό Νοσοκομείων																																																								

ΑΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

67

Νοσοκομείο-Εμφάνιση Ιστορικού Ιδίου- 2^η Περίπτωση

Καλώς ήρθατε στο Σύστημα Οφθαλμιατρικών Εξετάσεων

Κατηγορίες	Ιστορικό Νοσοκομείων	Χρήστης
Οφθαλμ. Ραντίφθο	<p>Στο νοσοκομείο σας δεν έχει παραπεμφθεί κανένας ασθενής και συνεπώς δεν έχει πραγματοποιηθεί καμία εξέταση!!!</p>	<p>Κωδικός Νοσοκομείου: 10002</p> <p>Όνομασία Νοσοκομείου: METROPOLITAN Hospital</p> <p>Αποσύνδεση</p>
Αποτελεσμα Εξέτασης		
Ιστορικό Ασθενούς		
Ιστορικό Νοσοκομείων		

ΑΓΕΙ ΚΡΗΤΗΣ - ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ - ΠΑΝΑΓΙΩΤΗΣ ΖΙΩΓΑΣ (ΑΜ 679)

68

ΤΕΛΟΣ ΠΑΡΟΥΣΙΑΣΗΣ

- ΣΑΣ ΕΥΧΑΡΙΣΤΩ ΓΙΑ ΤΗΝ ΠΡΟΣΟΧΗ ΚΑΙ ΤΟΝ ΧΡΟΝΟ ΣΑΣ

