



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

Τίτλος: Κατασκευή εφαρμογής για καταχώρηση και παρακολούθηση ασθενών του ογκολογικού τμήματος καρκίνου του μαστού.

Μπούκλας Ιωάννης (ΑΜ: 225)

Στυλιανού Σταύρος (ΑΜ: 301)

Επιβλέπων καθηγητής: Παπαδουράκης Γεώργιος

Επιτροπή Αξιολόγησης:

Ημερομηνία παρουσίασης:

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της πτυχιακής εργασίας θα θέλαμε να ευχαριστήσουμε όλους όσους μας βοήθησαν και μας στήριξαν κατά τη διάρκεια εκπόνησης της πτυχιακής αυτής και ιδιαίτερα στον καθηγητή μας κύριο Παπαδουράκη Γεώργιο για τις συμβουλές, την καθοδήγηση και το χρόνο του.

Θα θέλαμε επίσης να ευχαριστήσουμε την κυρία Μανωλακάκη Δήμητρα, διευθύντρια του ιατρείου του καρκίνου του μαστού του νοσοκομείου Κατερίνης και τον κύριο Σταματίδη Ιορδάνη, υπεύθυνο τεχνολογικού εξοπλισμού του νοσοκομείου Κατερίνης που μας παρήχαν τις απαραίτητες πληροφορίες και διευκολύνσεις για το ξεκίνημα της εργασίας.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Ευχαριστίες	ii
Πίνακας περιεχομένων	iii
Πίνακας εικόνων	vi
Λίστα πινάκων.....	vii
Abstract περίληψη στα Αγγλικά.....	viii
Abstract περίληψη στα Ελληνικά.....	ix
1. Εισαγωγή.....	1
1.1 Περίληψη.....	1
1.2 Κίνητρο για την Διεξαγωγή της Εργασίας	1
1.3 Σκοπός και Στόχοι Εργασίας.....	1
1.4 Δομή Εργασίας.....	1
2. Θεμελιώδεις έννοιες	3
2.1 Λίγα λόγια για τον καρκίνο του μαστού	3
2.1.1 Ανατομία μαστού	3
2.1.2 Στατιστικά στοιχεία.....	4
2.1.3 Παράγοντες κινδύνου.....	4
2.1.4 Διάγνωση.....	5
2.1.5 Θεραπεία	6
2.2 Ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE)	7
2.2.1 Σκοπός των IDE	7
2.2.2 Ιστορία των IDE	7
2.2.3 Βασικές λειτουργίες IDE.....	8
2.2.4 Αναφορά στα πιο γνωστά IDE	10
2.3 Αντικειμενοστραφής προγραμματισμός.....	12
2.3.1 Κλάσεις (Classes).....	13
2.3.2 Κληρονομικότητα (Inheritance)	14
2.3.3 Σύνθεση (Composition).....	14
2.3.4 Γλώσσες προγραμματισμού	15
2.4 Εισαγωγή στις Βάσεις Δεδομένων	17
2.4.1 Τα Δεδομένα και οι Πληροφορίες.....	17
2.4.2 Η Οργάνωση Αρχείων.....	18
2.4.3 Προβλήματα της Οργάνωσης Αρχείων	19
2.4.4 Οι Βάσεις Δεδομένων και τα ΣΔΒΔ (DBMS).....	19
2.4.5 Η Αρχιτεκτονική των ΣΔΒΔ	20
2.4.6 Βασικά μοντέλα Βάσεων Δεδομένων	21
3. Εργαλεία ανάπτυξης της εφαρμογής	26
3.1 Visual Studio	26
3.1.1 Περιβάλλον	26
3.1.2 Στοιχεία ελέγχου (Components) του toolbox	27
3.1.3 Επιμέρους στοιχεία ελέγχου.....	29
3.1.4 Αντικειμενοστραφής Προγραμματισμός.....	34
3.2 Microsoft Access.....	36
3.2.1 Περιβάλλον	37
3.2.2 Πίνακες.....	37
3.2.3 Φόρμες	38
3.2.4 Ερωτήματα	39
3.2.5 Αντικείμενα της Access	39
3.2.6 Πρωτεύοντα και Ξένα Κλειδιά.....	40
3.2.7 Δημιουργία μιας Βάσης Δεδομένων στην Access	41
3.2.8 Αναφορά σε τύπους δεδομένων μιας βάσης Access.	41

3.3 OLE DB τεχνολογία διασύνδεσης Βάσης Δεδομένων με εφαρμογή.....	43
3.4 Photoshop.....	44
3.4.1 Το περιβάλλον.....	45
3.4.2 Εικόνα.....	46
3.5 Microsoft HTML help workshop.....	47
3.6 Inno Setup.....	47
4. Αρχές υλοποίησης εφαρμογής.....	49
4.1 Αρχές σχεδίασης διεπαφής.....	49
4.1.1 Ανθρώπινοι παράγοντες στο σχεδιασμό διεπαφής.....	50
4.1.2 Αρχές σχεδιασμού διεπαφής.....	50
4.1.3 Οπτικοποίηση δεδομένων.....	51
4.1.4 Προβολές με χρώματα.....	51
4.1.5 Μηνύματα σφαλμάτων.....	51
4.1.6 Γνωρίσματα χρηστικότητας.....	52
4.1.7 Γραμματοσειρά.....	52
4.1.8 Πλοήγηση.....	52
4.2 Αρχές σχεδίασης Βάσεων δεδομένων.....	53
4.2.1 Η διαδικασία σχεδίασης.....	54
5. Ανάπτυξη εφαρμογής.....	58
5.1 Προετοιμασία.....	58
5.2 Δημιουργία interface.....	59
5.2.1 Το κεντρικό παράθυρο της εφαρμογής.....	59
5.2.2 Παράθυρα διαλόγου μηνυμάτων.....	67
5.3 Δημιουργία εικονιδίων και γραφικών της εφαρμογής.....	69
5.4 Δημιουργία αρχείου βοήθειας.....	70
5.5 Ανάπτυξη Βάσης Δεδομένων.....	71
5.5.1 Σκοπός δημιουργίας της βάσης δεδομένων και ερωτήματα που θέλουμε αυτή να απαντά.....	71
5.5.2 Πίνακες, Γνωρίσματα (Στήλες) που περιέχονται σε αυτούς και περιορισμοί Γνωρισμάτων.....	71
5.5.3 Διάγραμμα σχέσεων-οντοτήτων.....	77
5.5.4 Επεξήγηση των Σχέσεων των Πινάκων της Βάσης.....	78
5.5.5 SQL queries.....	80
5.6 Υλοποίηση κώδικα.....	81
5.6.1 Δημιουργία κώδικα για interface.....	82
5.6.2 Έλεγχοι εγκυρότητας δεδομένων και περιορισμοί.....	83
5.6.3 Καταχώρηση, αποθήκευση και διαγραφή δεδομένων από τη βάση.....	85
5.6.4 Προβολή δεδομένων.....	87
5.6.5 Αντίγραφα ασφαλείας.....	91
5.6.6 Υλοποίηση κώδικα διασύνδεσης με τη βάση δεδομένων.....	91
5.7 Δημιουργία αρχείου εγκατάστασης.....	93
5.8 Διαδικασία ελέγχου.....	93
6. Αποτελέσματα.....	94
6.1.Συμπεράσματα.....	94
6.2.Χρήση.....	94
6.3.Μελλοντική εργασία και επεκτάσεις.....	95
Βιβλιογραφία.....	96
Πηγές από το διαδίκτυο.....	96
Παραρτήματα.....	97
Παράρτημα Α-Πηγαίος κώδικας.....	97
Παράρτημα Β-Παρουσίαση.....	157
Παράρτημα Γ-Περίληψη πτυχιακής σε στυλ δημοσίευσης.....	165
1. Εισαγωγή.....	165
2. Στόχοι Εργασίας.....	165
3. Ανάπτυξη της εφαρμογής.....	165

4. Σχεδιασμός διεπαφής	166
5. Σχεδίαση Βάσεων δεδομένων	167
6. Ανάλυση της εφαρμογής.....	167
7. Βάση Δεδομένων.....	168
8. Επεκτάσεις	169
Βιβλιογραφία.....	169
Πηγές από το διαδίκτυο.....	170

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Ανατομία του μαστού	3
Εικόνα 2: Είδη χειρουργικών επεμβάσεων του μαστού.....	6
Εικόνα 3 Λογότυπο του Microsoft Visual Studio	10
Εικόνα 4 Λογότυπο Microsoft Visual Studio.....	26
Εικόνα 5 Περιβάλλον λειτουργίας Microsoft Visual Studio	26
Εικόνα 6: Διάφορες προβολές των Ms Charts	30
Εικόνα 7: Μοντέλο Pull	32
Εικόνα 8: Μοντέλο Push	32
Εικόνα 9 ReportDocument Object Model Diagram	33
Εικόνα 10: Περιβάλλον της ACCESS.....	37
Εικόνα 11: Περιβάλλον εργασίας Adobe Photoshop	45
Εικόνα 12: Περιβάλλον εργασίας HTML Workshop.....	47
Εικόνα 13 Αρχική μορφή στοιχείων	58
Εικόνα 14 Το περιβάλλον εργασίας της εφαρμογής	59
Εικόνα 15: Tab page Νέος Ασθενής	61
Εικόνα 16 Tab page Υπάρχων Ασθενής.....	61
Εικόνα 17: Tab Control Καρτελών	62
Εικόνα 18: Data grid view Κλινικής Εξέτασης.....	62
Εικόνα 19: Tab page Παρούσα νόσος/Οικογενειακό ιστορικό	63
Εικόνα 20: Tab page Παράγοντες κινδύνου.....	63
Εικόνα 21: Tab page Κλινική εξέταση.....	64
Εικόνα 22: Tab page Διαγνωστικός Έλεγχος.....	64
Εικόνα 23: Tab Page Σταδιοποίηση καρκίνου	65
Εικόνα 24: Tab page Παρακολούθηση Ασθενών με καρκίνο μαστού	65
Εικόνα 25: Form Στατιστικών.....	66
Εικόνα 26: Form Αντιγράφων ασφαλείας.....	66
Εικόνα 27: Form Εκτύπωσης	67
Εικόνα 28: Παράθυρο επιλογών εκτύπωσης.....	67
Εικόνα 29: Παράθυρο δημιουργίας κενής εικόνας	69
Εικόνα 30: Σχέδιο εικονιδίου	69
Εικόνα 31: Πάχος σχεδίου εικονιδίου	69
Εικόνα 32: Σελίδες του HTML Workshop project.....	70
Εικόνα 33: Index του αρχείου βοήθειας.....	71
Εικόνα 34: Πίνακας BasicData	72
Εικόνα 35: Διάγραμμα σχέσεων οντοτήτων	77
Εικόνα 36: Ερωτήματα SQL της Access.....	80

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

Πίνακας 1 Συχνότητα καρκίνου στις γυναίκες συνόλου 2004-2008.....	4
Πίνακας 2 Συγκριτικά IDE.....	12
Πίνακας 3: Βασικά χαρακτηριστικά Ms charts.....	29
Πίνακας 4 Στοιχεία OOP.....	35
Πίνακας 5 Παράδειγμα πίνακα βάσης δεδομένων.....	38
Πίνακας 6 Ιδιότητες της Access.....	38
Πίνακας 7 Τύποι δεδομένων της ACCESS.....	41
Πίνακας 8 Ιδιότητες του Autonumber.....	42
Πίνακας 9 Ιδιότητες κειμένου.....	42
Πίνακας 10 Ιδιότητες number.....	43
Πίνακας 11 Αντικείμενα του OLE DB.....	44
Πίνακας 12 Στυλ αλληλεπιδράσεων.....	49
Πίνακας 13 Παράγοντες.....	51
Πίνακας 14 Πίνακας BasicData.....	73
Πίνακας 15 Πίνακας DeseasesData.....	73
Πίνακας 16 Πίνακας DangersData.....	74
Πίνακας 17 Πίνακας DiagnosisData.....	74
Πίνακας 18 Πίνακας ClinicalData.....	75
Πίνακας 19 Πίνακας StadioData.....	75
Πίνακας 20 Πίνακας ObserveData.....	76
Πίνακας 21 Πίνακας CoordinatesData.....	77

ABSTRACT ΠΕΡΙΛΗΨΗ ΣΤΑ ΑΓΓΛΙΚΑ

The topic of this project is to develop a windows application that can be used by a medical breast cancer lab .The purpose of this application is to record and file the basic data of the examined as their condition. The application was developed with the programming language Visual Basic on Visual Studio environment along with other tools found necessary such as Crystal Reports, Adobe Photoshop and others, in conjunction with Microsoft access database. The application is easy to use, lightweight and has sufficient management and fast installation. It can fully replace the amount of paper used in such medical lab and fasten the data retrieval of a patient record. In addition, it can provide statistical data that it would be almost impossible to extract out of paper files.

ABSTRACT ΠΕΡΙΛΗΨΗ ΣΤΑ ΕΛΛΗΝΙΚΑ

Η πτυχιακή αυτή έχει να κάνει με όλες τις παραμέτρους σχεδιασμού και υλοποίησης μιας εφαρμογής η οποία καταγράφει και αρχειοθετεί τα συμπεράσματα και ευρήματα εξετάσεων σε ένα ιατρείο καρκίνου του μαστού. Τα δεδομένα αποθηκεύονται και ανακαλούνται από την εφαρμογή, σε και από, μια βάση δεδομένων, παρέχοντας έτσι όλα τα πλεονεκτήματα της ψηφιακής διαχείρισης δεδομένων εκεί που παλιότερα χρησιμοποιούταν χαρτί. Στην πορεία υλοποίησης της εφαρμογής αυτής, είχαμε την ευκαιρία να εντρυφήσουμε στον προγραμματισμό στην πιο ουσιαστική του μορφή που είναι η παραγωγή μιας ολοκληρωμένης και λειτουργικής εφαρμογής καθώς και σε πολλά εργαλεία και αρχές που περιβάλλουν την ανάπτυξη μιας τέτοιας εφαρμογής.

1. Εισαγωγή

1.1 Περίληψη

Το πρόγραμμα BCP είναι μια εφαρμογή για windows που χρησιμοποιεί βάση δεδομένων της Microsoft Access και δίνει την δυνατότητα στο χρήστη να διαχειριστεί τα δεδομένα ασθενών του καρκίνου του μαστού μέσω μιας γνώριμης και φιλικής προς το χρήστη διεπαφής. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment-I.D.E) Visual studio 2008 με γλώσσα προγραμματισμού τη Visual Basic. Οι απαιτήσεις υλοποίησης της εφαρμογής κατέστησαν αναγκαία τη χρήση επιπλέον προγραμμάτων και βιβλιοθηκών πέραν των προεπιλεγμένων του περιβάλλοντος που χρησιμοποιήθηκε (Visual Studio). Αυτά είναι τα Microsoft HTML Help Workshop ,Photoshop, Ms Chart, και Crystal Reports. Επιπρόσθετα χρησιμοποιήθηκαν συμπληρωματικά οι γλώσσες SQL, HTML και Crystal Syntax.

Οι βασικές λειτουργίες του προγράμματος είναι η καταχώρηση, η αναζήτηση, η προβολή και επεξεργασία, η διαγραφή και η εκτύπωση των δεδομένων του ασθενούς. Το πρόγραμμα διαθέτει εγχειρίδιο χρήσης για σχετικά άπειρους με windows εφαρμογές χρήστες καθώς επίσης και λειτουργία προβολής στατιστικών και συγκεντρωτικών στοιχείων με βάση τα δεδομένα που έχουν καταχωρηθεί με στόχο τη διευκόλυνση του ιατρού να οδηγηθεί σε ωφέλιμα συμπεράσματα όπως επίσης να αποτελέσουν την αφορμή για περεταίρω έρευνα. Η εγκατάσταση της εφαρμογής πραγματοποιείται μέσω ενός αρχείου setup που περιλαμβάνει όλα τα στοιχεία που χρειάζονται για την ομαλή εκτέλεσή της (crystal runtime , jet drivers , net framework 3.5)

Πριν ξεκινήσει η ανάπτυξη της εφαρμογής πραγματοποιήθηκε καταγραφή του τεχνικού εξοπλισμού και λειτουργικού λογισμικού του ογκολογικού τμήματος καρκίνου του μαστού έτσι ώστε οι απαιτήσεις του προγράμματος να προσαρμοστούν σύμφωνα με τα δεδομένα που περισυλλέγησαν. Για την δημιουργία της διεπαφής λήφθηκαν υπόψη οι γνώμες χρηστών καθώς και οι βασικές αρχές σχεδίασης διεπαφής και παρέχεται στο χρήστη η δυνατότητα εναλλακτικών μεθόδων εργασίας. Η υλοποίηση της αρχιτεκτονικής της βάσης δεδομένων που χρησιμοποιεί το πρόγραμμα έγινε σύμφωνα με τις βασικές αρχές βάσεων δεδομένων για την καλύτερη οργάνωση και ταχύτερη προσπέλαση των δεδομένων.

1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Τυχαίνει να βρισκόμαστε κοντά στα ιατρικά επαγγέλματα και η τριβή μας με αυτόν τον τομέα μας έκανε να συνειδητοποιήσουμε ότι υπάρχουν κενά στην ηλεκτρονική διαχείριση των δεδομένων των ασθενών. Η ακρίβεια που απαιτείται για την υλοποίηση ενός συστήματος διαχείρισης αυτών των δεδομένων καθώς και η προσοχή για την αποφυγή λαθών, αποτρέπει την ενασχόληση με την δημιουργία προγραμμάτων ιατρικής παρακολούθησης

1.3 Σκοπός και Στόχοι Εργασίας

Η ανάπτυξη της εφαρμογής βασίστηκε σε έξι καρτέλες που χρησιμοποιούν ιατροί και ασθενείς για την συμπλήρωση των στοιχείων των ασθενών και των εξετάσεων. Στόχος της εφαρμογής είναι αρχικά η ηλεκτρονική αρχειοθέτηση των ασθενών και των εξετάσεων τους και ίσως η οριστική αντικατάσταση των καρτελών. Η εφαρμογή καθίσταται συμφέρουσα προς το περιβάλλον και το ογκολογικό τμήμα μαστού καθώς συντελεί στην εξοικονόμηση αποθηκευτικού χώρου και επειδή καθιστά την κατανάλωση αποθεμάτων χάρτου μη αναγκαία. Η ανάκτηση των δεδομένων γίνεται κατά πολύ ευκολότερη, παρέχοντας παράλληλα υπηρεσίες που η έντυπη αρχειοθέτηση καθιστά σχεδόν αδύνατες όπως είναι η εξαγωγή στατιστικών και η σύγκριση κλινικών ευρημάτων. Παράλληλα διευκολύνεται η φορητότητα των δεδομένων μέσω αποθηκευτικών μονάδων.

1.4 Δομή Εργασίας

Η συγγραφή του βιβλίου ολοκληρώθηκε σε 6 ενότητες. Οι κύριες ενότητες είναι η εισαγωγή, οι Θεμελιώδεις έννοιες, τα εργαλεία ανάπτυξης της εφαρμογής, οι αρχές ανάπτυξης εφαρμογής, η υλοποίηση εφαρμογής και συμπεράσματα. Κατά την ενότητα εισαγωγή, γίνεται μια περίληψη της πτυχιακής και αναφορά στο κίνητρο υλοποίησης της και στους στόχους της εργασίας. Στην ενότητα Θεμελιώδεις έννοιες παρατίθενται κάποιες έννοιες για τον καρκίνο του μαστού όπως επίσης και οι

θεμελιώδεις έννοιες για τα ολοκληρωμένα περιβάλλοντα ανάπτυξης, του αντικειμενοστραφούς προγραμματισμού και των βάσεων δεδομένων και αναλύονται τα δομικά στοιχεία τους. Στην ενότητα Εργαλεία ανάπτυξης της εφαρμογής γίνεται ανάλυση όλων των εργαλείων και πρόσθετων που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής όπως του Visual Studio και των controls Ms Charts και Crystal Reports, της Ms Access , του HTML Workshop και του Adobe Photoshop. Στην ενότητα Αρχές υλοποίησης εφαρμογής γίνεται εκτενής αναφορά στις αρχές που βασίστηκε η υλοποίηση του interface και η βάση δεδομένων της εφαρμογής. Στην ενότητα Ανάπτυξη εφαρμογής γίνεται ανάλυση της εφαρμογής και των βασικών βημάτων υλοποίησής της. Στο 6ο κεφάλαιο, γίνεται μια ανασκόπηση των αποτελεσμάτων της εργασίας και αναφέρονται τα συμπεράσματα της διεξαγωγής της. Ακόμη αναφέρονται οι επεκτάσεις που είναι δυνατές να γίνουν για μελλοντική εργασία επάνω στην εφαρμογή.

2. Θεμελιώδεις έννοιες

Σε αυτό το κεφάλαιο γίνεται αναφορά στο θεματικό αντικείμενο της εφαρμογής που είναι ο καρκίνος του μαστού. Επίσης αναλύονται οι θεμελιώδεις έννοιες και τα δομικά στοιχεία τους που αφορούν τα ολοκληρωμένα περιβάλλοντα ανάπτυξης, τον αντικειμενοστραφή προγραμματισμό και τις βάσεις δεδομένων.

2.1 Λίγα λόγια για τον καρκίνο του μαστού

Ο **καρκίνος του μαστού** αναφέρεται σε καρκίνο που αναπτύσσεται στους γαλακτοπαραγωγούς αδένες του μαστού ή στους γαλακτοφόρους πόρους που μεταφέρουν το γάλα. Διακρίνονται διαφορετικοί τύποι. Ανιχνεύεται με τη μέθοδο της μαστογραφίας. Οι θεραπείες για τον καρκίνο του μαστού περιλαμβάνουν χειρουργική επέμβαση, ορμονοθεραπεία, χημειοθεραπεία και ακτινοθεραπεία. Ο όρος «καρκίνος του μαστού» αναφέρεται στην ανάπτυξη κακοήθους όγκου στην περιοχή του μαστού. Αποτελεί μία από τις συχνότερα εμφανιζόμενες μορφές καρκίνου παγκοσμίως και είναι η πρώτη σε αριθμό κρουσμάτων στο γυναικείο πληθυσμό. Προκαλείται από ανεξέλεγκτο πολλαπλασιασμό παθολογικών κυττάρων που ως αποτέλεσμα προκαλούν το σχηματισμό κακοήθους όγκου στην περιοχή του μαστού και ουσιαστικά αποτελεί κυτταρική νόσο. Τα παθολογικά αυτά κύτταρα έχουν τη δυνατότητα εξάπλωσης σε γειτονικούς ιστούς σε δυσάρεστες συνέπειες για ολόκληρο τον οργανισμό. Όσον αφορά στις γυναίκες όλες αντιμετωπίζουν τον κίνδυνο εμφάνισης της νόσου αλλά όχι στον ίδιο βαθμό.

2.1.1 Ανατομία μαστού

Η εξωτερική μορφολογία του μαστού περιλαμβάνει την θηλή, την θηλαία άλω και τα αλωαία οζίδια. Η **θηλή** αποτελεί έπαρμα του δέρματος του μαστού που βρίσκεται λίγο πιο κάτω και έξω από το μέσο του μαστού. Στην κορυφή της υπάρχουν 15-20 στόμια όπου καταλήγουν οι γαλακτοφόροι πόροι. Το ύψος της θηλής είναι περίπου 1-1,5 cm και αυξάνει κατά την γαλουχία (θηλασμός). Το καστανέρυθρο χρώμα της θηλής οφείλεται στην άφθονη παρουσία μελανίνης ουσίας. Η **θηλαία άλω** αποτελεί υποστρόγγυλη και ελαφρά επηρμένη περιοχή γύρω από την θηλή με διάμετρο 1,5-6 cm. Στην επιφάνεια της θηλαίας άλω υπάρχουν μικρά έπαρματα, τα θηλαία οζίδια τα οποία έχουν ως υπόθεμα τους αλωαίους αδένες. Οι αλωαίοι αδένες είναι κυρίως οσμηγόνοι, αλλά και σμηγματογόνοι και υποτυπώδεις γαλακτικοί αδένες.

Εσωτερικά ο μαστός αποτελείται από τον μαστικό ή μαζικό αδένα και το περιμαστικό λίπος. Το περιμαστικό λίπος είναι συνέχεια του υποδόριου λίπους το οποίο όμως είναι αφθονότερο στην πρόσθια περιοχή του μαστού, ανάμεσα στο δέρμα και τον μαστικό αδένα. Ο **μαστικός αδένας** σε γυναίκα μη κυοφορούσα βρίσκεται πίσω από την θηλαία άλω και μόλις που υπερβαίνει τα όριά της. Ο μαστικός αδένας αποτελείται από τους λοβούς όπου παράγεται το γάλα και τους γαλακτοφόρους πόρους που μεταφέρουν το γάλα στους γαλακτοφόρους κόλπους. Οι γαλακτοφόροι κόλποι είναι ανευρύσματα των πόρων τα οποία λειτουργούν ως αποθήκη του γάλακτος, το οποίο και απελευθερώνουν μετά από πίεση της θηλής από το βρέφος.



Εικόνα 1: Ανατομία του μαστού

2.1.2 Στατιστικά στοιχεία

Σύμφωνα με τη Διεθνή Έκθεση για τις καρκινικές νόσους που παρουσιάστηκε στη Γενεύη της Ελβετίας τον Απρίλιο του 2003 από την IARC (International Agency for Research on Cancer), ο καρκίνος του μαστού αποτελεί την πιο συχνή μορφή καρκίνου μεταξύ των γυναικών, με περίπου 1.000.000 νέα κρούσματα παγκοσμίως. Ορισμένοι αριθμοί δίνουν τη κοινωνική διάσταση του καρκίνου του μαστού. Στις ΗΠΑ το 2004 οι νέες περιπτώσεις καρκίνου του μαστού στις γυναίκες ήταν 217.000 περίπου. Στην Ελλάδα αναφέρονται 4.500 περίπου νέες περιπτώσεις το χρόνο, ενώ υπολογίζεται ότι 1 στις 8 γυναίκες παγκοσμίως θα παρουσιάσει καρκίνο μαστού σε κάποια φάση της ζωής της. Στην Ευρώπη, το 60% των κρουσμάτων καρκίνου του μαστού διαγιγνώσκεται σε πρώιμο στάδιο. Το αντίστοιχο ποσοστό στην Ελλάδα είναι μόλις 5%. Τα στοιχεία αυτά καταδεικνύουν πόσο ελλιπής είναι η σχετική ενημέρωση μεταξύ των Ελληνίδων.

Είναι ένα από τα είδη καρκίνου που επιφέρουν τους περισσότερους θανάτους ετησίως (519.000 θανάτους το 2004). Ωστόσο, τα ποσοστά θανάτου από καρκίνο του μαστού χαρακτηρίζονται από πτωτική τάση από τις αρχές του 1990, με τις μεγαλύτερες μειώσεις να εντοπίζονται στις γυναίκες κάτω των 50. Οι ερευνητές αποδίδουν την πτώση αυτή στην έγκαιρη διάγνωση μέσω μαστογραφιών καθώς και στις βελτιώσεις που έχουν επέλθει στις σχετικές θεραπευτικές αγωγές. Ο αριθμός των ατόμων που έχουν αντιμετωπίσει με επιτυχία τον καρκίνο του μαστού αυξάνεται συνεχώς - από τον Ιανουάριο του 2006, υπήρξαν περίπου 2,5 εκατομμύρια γυναίκες στις ΗΠΑ που, βάσει της έκθεσης, έχουν ξεπεράσει με επιτυχία την περιπέτεια του καρκίνου του μαστού.

Έξι συχνότεροι καρκίνοι στις γυναίκες, % συνόλου, 2004-2008		
C50 (Μαστού)	2120	34,9%
C18-C20 (Ορθοκοιλιακού)	691	11,4%
C54 (Σώματος μήτρας)	362	6,0%
C73 (Θρεοειδούς)	357	5,9%
C33-C34 (Τραχείας, Βρόγχου και πνεύμονα)	242	4,0%
C82-C85;C96 (Λέμφωμα μη Hodgkin)	229	3,8%
Άλλοι	2076	34,2%
Σύνολο όλων των καρκίνων	6077	100,0%

Πίνακας 1: Συχνότητα καρκίνου στις γυναίκες συνόλου 2004-2008

2.1.3 Παράγοντες κινδύνου

Ελάχιστα είναι γνωστά σχετικά με τα αίτια που προκαλούν καρκίνο του μαστού, παρά το γεγονός ότι έχουν εντοπιστεί αρκετοί παράγοντες κινδύνου οι οποίοι αναφέρονται παρακάτω:

- **Ηλικία:** ο καρκίνος του μαστού μπορεί να προκύψει σε οποιαδήποτε ηλικία μετά την εφηβεία αλλά τα ποσοστά αυξάνονται όσο αυξάνονται και οι ηλικιακές κλίμακες. Οι περισσότερες περιπτώσεις παρουσιάζονται μετά από την ηλικία των 50 ετών, ενώ είναι σπάνιος σε γυναίκες ηλικίας κάτω των 35 ετών (5% των περιπτώσεων), με εξαίρεση τις γυναίκες που έχουν κληρονομική προδιάθεση.
- **Κληρονομικότητα:** Υπολογίζεται ότι μόλις το 5 - 10% των κρουσμάτων καρκίνου του μαστού σχετίζεται όντως με παράγοντες κληρονομικότητας. Ωστόσο, δύο γονίδια, γνωστά ως BRCA 1 και BRCA 2, έχουν προσδιοριστεί ως παράγοντες που συμβάλλουν στην εμφάνιση καρκίνου του μαστού. Επίσης, γυναίκες με εξ αίματος συγγενείς που έχουν νοσήσει αντιμετωπίζουν αυξημένο κίνδυνο εμφάνισης καρκίνου στο μαστό.
- **Διαταραχές της εμμηνου ρύσης:** Στοιχεία υποδηλώνουν πως γυναίκες με πρώιμη έναρξη της εμμηνου ρύσης (πριν από το 12ο έτος της ηλικίας τους) ή με καθυστερημένη εμμηνόπαυση (μετά τα 55) αντιμετωπίζουν αυξημένο κίνδυνο εμφάνισης καρκίνου στο μαστό. Επίσης, η λήψη οιστρογόνων μετά την εμμηνόπαυση έχει συσχετιστεί με αυξημένα ποσοστά εμφάνισης της νόσου, με τον κίνδυνο να είναι ανάλογος του διαστήματος λήψης των οιστρογόνων. Ανάλογος συσχετισμός έχει προκύψει και για γυναίκες που παρέμειναν άτεκνες, που δεν είχαν πλήρεις κυήσεις (διάρκεια εννέα μηνών) ή που γέννησαν μετά τα τριάντα πέντε τους χρόνια.
- **Αλκοόλ:** Τα οινοπνευματώδη ποτά αυξάνουν τη συγκέντρωση των οιστρογόνων στο αίμα. Σύμφωνα με μελέτη που παρουσιάστηκε το Δεκέμβριο του 2009 στο Διεθνές Συνέδριο

ογκολογίας του Σαν Αντόνιο, άτομα που νόσησαν από καρκίνο του μαστού και καταναλώνουν με μετριοπάθεια αλκοόλ διατρέχουν μεγαλύτερο κίνδυνο επανεμφάνισής του από εκείνα που πίνουν λίγο ή καθόλου οινοπνευματώδη.

- **Παχυσαρκία:** Η παχυσαρκία αυξάνει τον κίνδυνο καρκίνου του μαστού καθώς αυξάνει τα επίπεδα των οιστρογόνων. Η παραγωγή των οιστρογόνων στις γυναίκες μετά την εμμηνόπαυση γίνεται κυρίως μέσα σε λιπώδη ιστό (μετατροπή των επινεφριδικών ανδρογόνων σε οιστρογόνα από την αρωματάση, ένα ένζυμο που βρίσκεται κυρίως στο λίπος). Τον Ιούνιο του 2009 ανακοινώθηκε από το Αμερικανικό Ίδρυμα για την έρευνα του Καρκίνου (AICR) ότι η συσσώρευση σωματικού λίπους σε ποσοστά άνω του κανονικού ευθύνεται για το 17% των κρουσμάτων καρκίνου του μαστού στις Η.Π.Α.
- **Κάπνισμα:** πρόσφατες μελέτες απέδειξαν ότι η κατανάλωση ενός πακέτου τσιγάρων ημερησίως, από γυναίκες προ της εμμηνόπαυσης για εννέα περίπου χρόνια, αυξάνει δραστικά τον κίνδυνο εμφάνισης καρκίνου του μαστού κατά σχεδόν 60%.
- **Ιστορικό Καρκίνου:** γυναίκες που έχουν εμφανίσει προηγουμένως καρκίνο της μήτρας, των ωοθηκών ή του μαστού έχουν αυξημένες πιθανότητες να εμφανίσουν έναν 2ο καρκίνο στο μαστό.
- **Καθιστική Ζωή:** Η τακτική άσκηση πριν την έναρξη της εμμηνου ρύσης μπορεί να μειώσει τον κίνδυνο καρκίνου του μαστού μιας γυναίκας, κυρίως διότι μπορεί να καθυστερήσει την έναρξη της εμμηνου ρύσεως, να επιμηκύνει τον χρόνο μεταξύ των περιόδων ή να ελαττώσει τον αριθμό των εμμηνορυσιακών κύκλων, μειώνοντας έτσι την έκθεση της γυναίκας στα οιστρογόνα.
- **Έκθεση σε ακτινοβολία**
- **Ατεκνία**
- **Θεραπεία Ορμονικής Υποκατάστασης:** προσφέρει ανακούφιση από τα συμπτώματα της εμμηνόπαυσης, ωστόσο, η μακροχρόνια χρήση της μετά την εμμηνόπαυση αυξάνει τον κίνδυνο ανάπτυξης καρκίνου του μαστού.

2.1.4 Διάγνωση

Ο καρκίνος του μαστού δεν προλαμβάνεται ακόμα πρωτογενώς διότι δεν γνωρίζουμε ακόμα τον παράγοντα δημιουργίας του αλλά δευτερογενώς με έγκαιρη διάγνωση, πρόληψη και θεραπεία. Η δευτερογενής πρόληψη περιλαμβάνει αυτοεξέταση, κλινική εξέταση από ιατρό και μαστογραφία. Η διάγνωση ξεκινά όταν ανακαλυφθεί μια μάζα στο μαστό μετά από ψηλάφηση ή μαστογραφία ή υπέρηχο και στη συνέχεια ακολουθούν εξετάσεις ώστε να διευκρινιστεί η διαφορά ανάμεσα σε ένα συμπαγή όγκο και μια κύστη γεμάτη υγρό. Υπάρχουν 4 διαγνωστικές μέθοδοι:

1. **Κυτταρολογική βιοψία:** Γίνεται με μια λεπτή βελόνα (FNA - Fine Needle Aspiration Biopsy) που προσαρμόζεται σε μια σύριγγα και επιτρέπει την αναρρόφηση κυττάρων από την ύποπτη περιοχή του μαστού.
2. **Ιστολογική βιοψία:** η βελόνα σε αυτή την περίπτωση είναι μεγαλύτερη, απαιτείται τοπική αναισθησία και τομή και το υλικό που μελετάται είναι ιστολογικό.
3. **Στερεοτακτική βιοψία:** η λήψη του ιστολογικού υλικού γίνεται με την καθοδήγηση ειδικών μηχανημάτων
4. **Ανοιχτή χειρουργική βιοψία:** η ταυτοποίηση της φύσης της ύποπτης περιοχής γίνεται με ταχεία βιοψία κατά τη διάρκεια του χειρουργείου πριν την αφαίρεση του όγκου.

Καλοήθεις παθήσεις. Οι καλοήθεις παθήσεις μπορούν να είναι:

- Ινοαδένωμα
- Κύστη
- Λίπωμα
- Φυλλοειδής όγκος
- Θήλωμα
- Φλεγμονή-Απόστημα
- Ινοκυστική μαστοπάθεια
- Γαλακτοφορεκτασία
- Μασταλγία
- Μαστίτιδα

- Γαλακτοκήλη
- Αδενολίπωμα

Κακοήθεις Παθήσεις. Οι κακοήθεις παθήσεις μπορούν να είναι:

- Επί τόπου Πορογενές Καρκίνωμα (Ductal carcinoma in situ [DCIS])
- Πορογενές διηθητικό καρκίνωμα (Invasive Ductal Carcinoma [IDC])
- Σωληνώδες Καρκίνωμα του μαστού [Tubular carcinoma]
- Μυελώδες Καρκίνωμα στο στήθος
- Βλεννοπαράγωγο Μαστικό Καρκίνωμα
- Θηλώδες (Θηλοειδές) Καρκίνωμα στο μαστό (Papillary Carcinoma of the Breast)
- Διάτρητο Καρκίνωμα του Μαστού [Cribriform Carcinoma of the Breast]
- Διηθητικό Λοβιακό Καρκίνωμα (Invasive lobular carcinoma [ILC])
- Φλεγμονώδης Καρκίνος του Μαστού (Inflammatory breast cancer)

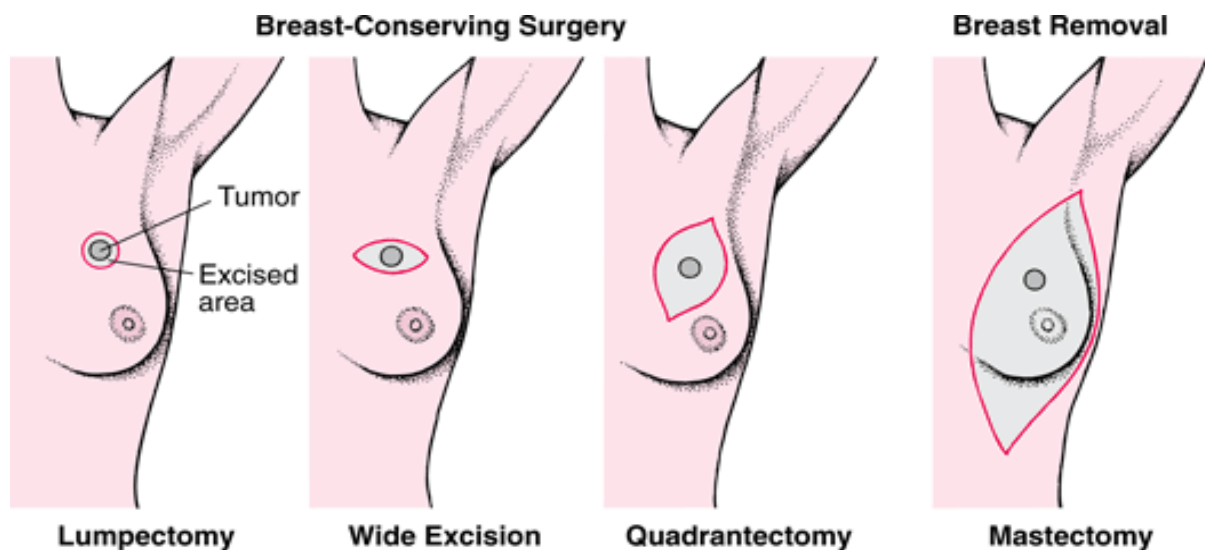
2.1.5 Θεραπεία

Το αρχικό βήμα για την αντιμετώπιση της νόσου είναι η χειρουργική επέμβαση για την αφαίρεση του όγκου. Στη συνέχεια η μετεγχειρητική θεραπεία μπορεί να περιλαμβάνει Ακτινοθεραπεία, Χημειοθεραπεία, Ορμονοθεραπεία. Τα στοιχεία που λαμβάνονται υπόψη στην επιλογή της μετεγχειρητικής θεραπείας είναι το μέγεθος και η διαφοροποίηση του όγκου, η ύπαρξη ορμονικών υποδοχέων και η γενική κατάσταση της υγείας.

- Χημειοθεραπεία: καταστρέφει τα κύτταρα που έχουν μεταναστεύσει από τον αρχικό όγκο. Χορηγείται ενδοφλεβίως, με συνδυασμό 2-3 φαρμάκων, σε κύκλους θεραπειών χωρίς να απαιτείται εισαγωγή στο νοσοκομείο. Επειδή τα φάρμακα είναι χημικά έχουν δυσάρεστες παρενέργειες για τον οργανισμό κυρίως στο αίμα (πτώση αιματοκρίτη και λευκών), στο γαστρεντερικό σωλήνα (εμετοί), στα μαλλιά (πτώση) και στα νύχια (μαύρισμα) οι οποίες όμως σήμερα περιορίζονται με τη χορήγηση βοηθητικών φαρμακευτικών σκευασμάτων.

- Ακτινοθεραπεία: μπορεί να χορηγηθεί προεγχειρητικά με σκοπό την ελάττωση του μεγέθους του όγκου ή ως συμπλήρωμα των "συντηρητικών" χειρουργικών επεμβάσεων όπως η μερική μαστεκτομή ή σε ορισμένες περιπτώσεις ριζικής μαστεκτομής.

- Ορμονοθεραπεία: Εάν στον όγκο υπάρχουν ορμονικοί υποδοχείς μπορεί να χρησιμοποιηθεί Ορμονοθεραπεία μόνη ή σε συνδυασμό με Χημειοθεραπεία και /ή Ακτινοθεραπεία. Δεν είναι όλες οι αλλοιώσεις, διογκώσεις ή σκληρύνσεις καρκινικές και κακοήθεις. Υπολογίζεται πως το 90% των μαστικών όγκων σε γυναίκες ηλικίας 20 – 50 ετών προκαλούνται από κάποια πάθηση καλοήθους φύσης.



Εικόνα 2: Είδη χειρουργικών επεμβάσεων του μαστού

2.2 Ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE)

Το λογισμικό ενός υπολογιστή μπορεί να διαιρεθεί σε δυο κατηγορίες . Τα προγράμματα συστήματος που διαχειρίζονται την λειτουργία του ίδιου του υπολογιστή και τα προγράμματα των εφαρμογών που επιλύουν τα προβλήματα των χρηστών του. Το βασικότερο από τα προγράμματα συστήματος είναι το λειτουργικό σύστημα που ελέγχει όλους τους πόρους του υπολογιστή και παρέχει τη βάση πάνω στην οποία θα αναπτυχθούν τα προγράμματα των εφαρμογών. Το πλέον διαδεδομένο λειτουργικό σύστημα είναι τα Windows της Microsoft πάνω στο οποίο αναπτύχθηκε και δοκιμάστηκε και η εφαρμογή μας.

Με την καθιέρωση των Windows αναπτύχθηκαν γλώσσες όπως η visual basic , το Delphi και η visual c. Οι οποίες καθιέρωσαν τον λεγόμενο οπτικό προγραμματισμό (visual programming). Ο οπτικός προγραμματισμός δίνει την δυνατότητα στον προγραμματιστή με πολύ μικρότερη προσπάθεια (σε σχέση με παλαιότερες γλώσσες) , να δημιουργήσει ισχυρές εφαρμογές που να διαθέτουν πολύ καλό περιβάλλον διασύνδεσης με τον χρήστη. Εκτεταμένη χρήση του οπτικού προγραμματισμού κάνουν τα ολοκληρωμένα περιβάλλοντα ανάπτυξης .

Ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE) είναι μία σουίτα λογισμικού που διευκολύνει την ανάπτυξη προγραμμάτων παρέχοντας στον χρήστη διάφορα εργαλεία και γραφικό περιβάλλον. Οι λειτουργίες που παρέχονται στο χρήστη τον διευκολύνουν να τελέσει τις ενέργειες του χωρίς να χρειαστεί να χρησιμοποιήσει δευτερεύουσες εφαρμογές. Συνήθως ένα IDE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή, συνδέτη , δομή δεδομένων για γρήγορη προσπέλαση κώδικα, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές.

2.2.1 Σκοπός των IDE

Τα IDE έχουν σχεδιαστεί για να μεγιστοποιήσουν την παραγωγικότητα του προγραμματιστή με την παροχή σχετικών συστατικών με παρόμοια interface. Το IDE είναι ένα ενιαίο πρόγραμμα, στο οποίο συντελείτε όλη η ανάπτυξη του λογισμικού. Αυτό το πρόγραμμα παρέχει συνήθως πολλαπλές δυνατότητες για τη συγγραφή, την τροποποίηση, την κατάρτιση, την ανάπτυξη και τον εντοπισμό σφαλμάτων λογισμικού. Διαφορετικά η ανάπτυξη λογισμικού μπορεί να επιτευχθεί χρησιμοποιώντας διαφορετικά εργαλεία.

Ένας από τους στόχους του IDE είναι να ελαττώσει τις απαραίτητες ρυθμίσεις για να ενσωματώσει πολλά βοηθητικά προγράμματα ανάπτυξης και να παρέχουν ένα σύνολο δυνατοτήτων ως συνεκτική μονάδα. Με τη μείωση του χρόνου εγκατάστασης μπορεί να αυξήσει την παραγωγικότητα του έργου, και σε ορισμένες περιπτώσεις το να μάθει κάποιος να χρησιμοποιεί το IDE είναι ταχύτερο από ότι να συγκεντρώσει και να κάνει τις ρυθμίσεις για όλα τα εργαλεία που θα χρειαστεί. Η ενσωμάτωση όλων των εργασιών ανάπτυξης μπορεί να βελτιώσει τη συνολική παραγωγικότητα πέρα από την απλή βοήθεια που παρέχει με την εύκολη εγκατάσταση. Για παράδειγμα, ο κώδικας μπορεί συνεχώς να αναλύεται ενώ ταυτόχρονα να γίνεται η επεξεργασία του , παρέχοντας άμεση ανάδραση τη στιγμή που εισάγονται συντακτικά λάθη. Αυτό μπορεί να επιταχύνει την εκμάθηση μιας νέας γλώσσας προγραμματισμού και των συναφών βιβλιοθηκών.

Μερικά IDEs χρησιμοποιούν μια συγκεκριμένη γλώσσα προγραμματισμού, παρέχοντας ένα χαρακτηριστικό σετ εργαλείων που ταιριάζει περισσότερο με τα πρότυπα προγραμματισμού της γλώσσας. Ωστόσο, υπάρχουν πολλά IDEs, που χρησιμοποιούν πολλαπλές γλώσσες προγραμματισμού όπως τα Eclipse, ActiveState Komodo, IntelliJ IDEA, Oracle JDeveloper, NetBeans, Microsoft Visual Studio, Genuitec MyEclipse και WinDev.

Ενώ τα περισσότερα σύγχρονα IDEs διαθέτουν γραφικό περιβάλλον ,πολλά που βασίζονται σε κείμενο όπως το Turbo Pascal είχαν ευρεία χρήση πριν από την διαθεσιμότητα των Microsoft Windows και του συστήματος X Window (X11). Χρησιμοποιούν συνήθως συντομεύσεις πλήκτρων ή αλλιώς hotkeys προκειμένου να εκτελέσουν τις συχνά χρησιμοποιούμενες εντολές ή αλλιώς μακροεντολές.

2.2.2 Ιστορία των IDE

Τα IDEs έγινε αρχικά δυνατή η δημιουργία τους όταν ακόμα η ανάπτυξη των προγραμμάτων γινόταν μέσω μιας κονσόλας ή τερματικού. Τα πρώτα συστήματα δεν θα μπορούσαν να τα

υποστηρίζουν, δεδομένου ότι τα προγράμματα αυτά προετοιμάζονταν με χρήση διαγραμμάτων ροής, είτε εισάγοντας προγράμματα με διάτρητες κάρτες (punched cards) πριν από την υποβολή τους σε compiler. Η Dartmouth BASIC ήταν η πρώτη γλώσσα που δημιουργήθηκε με IDE (και ήταν επίσης η πρώτη σχεδιασμένη για χρήση σε κονσόλα ή τερματικό). Αυτό το IDE του (μέρος του συστήματος κατανομής χρόνου Dartmouth) ήταν βασισμένο σε εντολές, και ως εκ τούτου δεν έμοιαζε καθόλου με τα IDEs με το γραφικό περιβάλλον και τα πολλαπλά μενού που είναι διαδεδομένα σήμερα. Ωστόσο η ενσωματωμένη επεξεργασία, η διαχείριση αρχείων, η ανάλυση, το debugging και η εκτέλεση κατά τρόπο ήταν παρόμοια με ένα σύγχρονο IDE.

Το Maestro I είναι ένα προϊόν της Softlab Μονάχου δημιουργημένο το 1975 και ήταν το πρώτο ολοκληρωμένο περιβάλλον ανάπτυξης στον κόσμο. Το Maestro I είχε εγκατασταθεί για 22.000 προγραμματιστές σε όλο τον κόσμο. Μέχρι το 1989, 6.000 εγκαταστάσεις του υπήρχαν στην Ομοσπονδιακή Δημοκρατία της Γερμανίας. Το Maestro I ήταν αναμφισβήτητο ο παγκόσμιος ηγέτης σε αυτόν τον τομέα κατά τη διάρκεια της δεκαετίας του 1970 και του 1980. Σήμερα, ένα από τα τελευταία Maestro I μπορούν να βρεθούν στο Μουσείο της Πληροφορικής στο Arlington. Ένα από τα πρώτα IDEs που υποστήριζε ένθετα (plug-in) ήταν το Softbench. Το 1995 το Computerwoche σχολίασε ότι η χρήση ενός IDE δεν θα τύχαινε ευνοϊκής υποδοχής από τους προγραμματιστές, επειδή θα έβαζε φραγμό στη δημιουργικότητά τους.

2.2.3 Βασικές λειτουργίες IDE

Κωδικοποίηση

Η διαδικασία της κωδικοποίησης είναι περίπλοκη και χρονοβόρα. Για τη διευκόλυνσή της έχουν σχεδιαστεί και αναπτυχθεί περιβάλλοντα υποστήριξης της κωδικοποίησης (programming support environments). Σε ένα περιβάλλον υποστήριξης της κωδικοποίησης όλος ο κύκλος της κωδικοποίησης από την αρχική εισαγωγή του προγράμματος μέχρι την τελική αποσφαλμάτωση υποστηρίζεται από ένα ολοκληρωμένο σύστημα. Έτσι ένα περιβάλλον υποστήριξης της κωδικοποίησης μπορεί να περιέχει σε ένα ολοκληρωμένο περιβάλλον έξυπνο διορθωτή, μεταγλωττιστή, συνδέτη, αποσφαλματωτή, ψηφιακή τεκμηρίωση, σύστημα υπολογισμού της σειράς μεταγλώττισης, σύστημα σχεδιασμού της διεπαφής με το χρήστη, φυλλομετρητή του πηγαίου κώδικα (source code browser) και φυλλομετρητή των κλάσεων (class browser), υπολογιστή κατανομής (profiler) του κόστους της επεξεργασίας μέσα στο πρόγραμμα.

Υπερσύνολο του περιβάλλοντος υποστήριξης της κωδικοποίησης αποτελεί το περιβάλλον τεχνολογίας λογισμικού (software engineering environment) το οποίο παρέχει υποστήριξη για όλο το κύκλο ζωής του λογισμικού συμπεριλαμβάνοντας δηλαδή υποστήριξη για την ανάλυση των απαιτήσεων, το σχεδιασμό, την κωδικοποίηση, τη διοίκηση του έργου, τον έλεγχο των εκδόσεων και του σχηματισμού, τη διαχείριση εξαρτημάτων λογισμικού (software components), τη διασφάλιση ποιότητας και τη συντήρηση.

Η τάση που εμφανίζεται είναι τα περιβάλλοντα υποστήριξης της κωδικοποίησης να περιέχουν όλο και περισσότερα στοιχεία από περιβάλλοντα τεχνολογίας λογισμικού ενώ παράλληλα περιβάλλοντα τεχνολογίας λογισμικού να υποστηρίζουν όλο και περισσότερο την κωδικοποίηση με τη χρήση γεννητριών κώδικα.

Μεταγλώττιση

Η διαδικασία της μεταγλώττισης περιλαμβάνει αρκετές ευκολίες σε ένα ολοκληρωμένο περιβάλλον. Υπολογίζονται αυτόματα τα αρχεία που απαιτούν μεταγλώττιση ανάλογα με τις αλλαγές που έγιναν σε αρχεία επικεφαλίδων. Η εμφάνιση ενός λάθους επιτρέπει την άμεση σύνδεσή του με τη γραμμή του κώδικα που το δημιούργησε. Επιτρέπει επίσης την εμφάνιση τεκμηρίωσης σχετικά με τα λάθη που εμφανίζονται.

Διορθωτής

Ο διορθωτής σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης περιλαμβάνει ισχυρές δυνατότητες εισαγωγής του κώδικα καθώς και δυνατότητες που σχετίζονται με τη γλώσσα προγραμματισμού όπως αυτόματη στοίχιση, συντακτικό χρωματισμό του κειμένου, εμφάνιση του ορίσματος των συναρτήσεων, συντακτικό έλεγχο, αναίρεση πολλαπλών επιπέδων, σελιδοδείκτες,

σύνδεση με το σύστημα τεκμηρίωσης, ταυτόχρονη διόρθωση πολλών αρχείων, εύρεση κειμένου σε αρχεία και αυτόματη μορφοποίηση του προγράμματος

Σε περιβάλλοντα που στοχεύουν στην ανάπτυξη προγραμμάτων με γραφικό σύστημα διεπαφής με το χρήστη συχνά υπάρχει και διορθωτής ο οποίος επιτρέπει το σχεδιασμό του συστήματος διεπαφής. Ο διορθωτής αυτό επιτρέπει την εισαγωγή στοιχείων του γραφικού περιβάλλοντος. Μερικά στοιχεία που απαντώνται συχνά είναι κουμπί εντολής (command button), πλαίσιο (frame), εισαγωγή κειμένου (edit box), μπάρα κύλισης (scrollbar) και κείμενο δείκτης προόδου (progress bar).

Το σύστημα βοήθειας

Το σύστημα βοήθειας περιλαμβάνει συχνά σε ψηφιακή μορφή τεκμηρίωση για τη γλώσσα προγραμματισμού, τη βιβλιοθήκη, το περιβάλλον ανάπτυξης και τα εργαλεία που το απαρτίζουν, τεκμηρίωση για τη διαδικασία ανάπτυξης εφαρμογών, οδηγίες συμβατότητας, απαντήσεις σε συχνές ερωτήσεις και διαπιστωμένα σφάλματα του περιβάλλοντος.

Το σύστημα βοήθειας είναι τις περισσότερες φορές παρουσιασμένο σε μορφή υπερκειμένου με πίνακες περιεχομένων και συνδέσεις ανάμεσα σε τμήματα. Μερικές φορές το σύστημα βοήθειας συμπληρώνεται από οδηγούς (wizards) που επιτρέπουν με διαλογικό τρόπο τη βήμα με βήμα ανάπτυξη μιας εφαρμογής. Συχνά υπάρχει άμεση σύνδεση του διορθωτή με το σύστημα βοήθειας έτσι ώστε την ώρα που π.χ. πληκτρολογούμε την κλήση σε μια συνάρτηση της βιβλιοθήκης να μπορούμε να δούμε τον ορισμό της.

Αποσφαλμάτωση

Ο αποσφαλματωτής επιτρέπει τον πλήρη έλεγχο της ροής εκτέλεσης και των δεδομένων του προγράμματος που εκτελείται. Περιλαμβάνει δυνατότητες όπως τη γραμμή προς γραμμή εκτέλεση του προγράμματος, την εμφάνιση της τιμής των μεταβλητών, τη διακοπή της εκτέλεσης του προγράμματος σε μια συγκεκριμένη γραμμή, τη διακοπή της εκτέλεσης του προγράμματος όταν αλλάξει τιμή μια μεταβλητή, την εκτέλεση μια ολόκληρης συνάρτησης, την εμφάνιση της τιμής μιας έκφρασης, την εκτέλεση του προγράμματος μέχρι ένα ορισμένο σημείο, την εμφάνιση της στοίβας κλήσεων (call stack) των συναρτήσεων, την εμφάνιση των καταχωρητών και της μνήμης του επεξεργαστή, την εμφάνιση του πηγαίου κώδικα καθώς και του συμβολικού κώδικα και τη δυναμική αλλαγή του πηγαίου κώδικα κατά τη διάρκεια της εκτέλεσης.

Φυλλομέτρηση πηγαίου κώδικα και κλάσεων

Συνδεδεμένο με το διορθωτή είναι συχνά ένα εργαλείο που επιτρέπει τη φυλλομέτρηση και την εμφάνιση της δομής του κώδικα και των κλάσεων που τον απαρτίζουν. Δυνατότητες του φυλλομετρητή μπορεί να είναι η εμφάνιση ορισμών και παραπομπών (references), η εμφάνιση της δομής ενός αρχείου, η εμφάνιση των βασικών κλάσεων και των μελών τους, η εμφάνιση του δένδρου της δομής των κλάσεων, η εμφάνιση του γράφου των καλούντων (call graph) συναρτήσεων και η εμφάνιση του γράφου των καλούμενων (callers graph) συναρτήσεων,

GUI builder

Ένας κατασκευαστής γραφικής διεπαφής χρήστη (graphical user interface builder, GUI builder) γνωστός και σαν σχεδιαστής διεπαφής (GUI designer) είναι ένα εργαλείο ανάπτυξης λογισμικού που απλοποιεί την δημιουργία διεπαφών επιτρέποντας τον χρήστη να τακτοποιήσει τα στοιχεία της διεπαφής οπτικά σέρνοντας τα από την εργαλειοθήκη και τοποθετώντας τα εκεί που ο ίδιος επιθυμεί. Χωρίς ένα GUI builder, για την δημιουργία ενός GUI πρέπει χειροκίνητα να καταχωρηθούν οι παράμετροι για κάθε στοιχείο του μέσω κώδικα χωρίς προεσκόπηση μέχρι να εκτελεστεί το πρόγραμμα. Καθώς οι διεπαφές χρήστη συνήθως προγραμματίζονται χρησιμοποιώντας event driven αρχιτεκτονική, οι GUI builders βοηθούν επιπλέον στο να απλοποιήσουν την δημιουργία event driven κώδικα. Αυτός ο απλοποιημένος κώδικας συνδέει αυτόματα τα στοιχεία της διεπαφής με εισερχόμενα και εξερχόμενα γεγονότα που ενεργοποιούν τις διάφορες λειτουργίες παρέχοντας στην εφαρμογή κατανοητή λογική.

Profiler

Στην ανάπτυξη λογισμικού η διαδικασία profiling είναι ένα είδος δυναμικής ανάλυσης προγραμμάτων που προσμετρά διάφορες ιδιότητες και στατιστικά του προγράμματος όπως το χώρο που καταλαμβάνει, την πολυπλοκότητα του ως προς το χρόνο εκτέλεσης ή τη συχνότητα και διάρκεια των κλήσεων των λειτουργιών (function calls) και απώτερος σκοπός της είναι η βελτιστοποίηση της εφαρμογής.

2.2.4 Αναφορά στα πιο γνωστά IDE

Τα τελευταία χρόνια έχουν αναπτυχθεί πολλά ολοκληρωμένα περιβάλλοντα ανάπτυξης λογισμικού. Μερικά από τα διασημότερα και ευρύτερα χρησιμοποιούμενα είναι το Visual Studio και το Visual J++ της Microsoft, το Eclipse, το NetBeans της Sun, το IntelliJ IDEA της JetBrains, ο Jbuilder της εταιρίας Borland και το Visual Café από την Symantec. Διαφέρουν μεταξύ τους στα πακέτα λειτουργιών που προσφέρουν, στις γλώσσες προγραμματισμού που υποστηρίζουν, στο τομέα εξειδίκευσης καθώς και στον τρόπο και κόστος διανομής τους.



Εικόνα 3: Λογότυπο του Microsoft Visual Studio

Microsoft Visual Studio

Το Visual Studio είναι ένα εύχρηστο περιβάλλον ανάπτυξης εφαρμογών, που αναπτύχθηκε από την εταιρεία Microsoft Corporation. Υποστηρίζει την ανάπτυξη προγραμμάτων σε κονσόλα, οπτικές εφαρμογές, ιστοσελίδες, υπηρεσίες WEB και άλλα.

Το περιβάλλον ανάπτυξης Visual Studio βοηθά τον προγραμματιστή να αναπτύξει τα προγράμματά του με σχετική ευκολία, καθώς η τεχνολογία Microsoft IntelliSense βοηθά τον προγραμματιστή να κατανοήσει με ευκολία πιθανά λάθη του κώδικά του, υπογραμμίζοντας τα με κόκκινη γραμμή δυναμικά κατά το χρόνο συγγραφής του προγράμματος. Αυτή η τεχνολογία είναι ικανή να εντοπίζει λάθη τα οποία μπορεί να είναι είτε συντακτικά, όπως για παράδειγμα η χρήση μιας εντολής με εσφαλμένο τρόπο, είτε λογικά, όπως για παράδειγμα η δήλωση ενός αντικειμένου χωρίς να αυτό να χρησιμοποιείται. Το Visual Studio υποστηρίζει την ανάπτυξη προγραμμάτων με γλώσσες προγραμματισμού C++, C#, Visual Basic, F# και τη μεταφορά προγραμμάτων από την μία γλώσσα στην άλλη. Με άλλα λόγια συγγράφει ένα πρόγραμμα σε γλώσσα C++ μπορεί απλά και εύκολα να μετατραπεί αυτόματα σε κάποια εκ των γλωσσών που υποστηρίζει το Visual Studio.

Visual Café

Είναι ένα προϊόν που αναπτύχθηκε αρχικά από την εταιρία Symantec που είναι γνωστή για τα πακέτα επικοινωνίας και καταπολέμησης ιών που κατασκευάζει. Το περιβάλλον του προγράμματος είναι παρόμοιο με αυτό της Visual Basic. Διαθέτει παράθυρα έργου, σχεδιαστή φορμών, παλέτα συστατικών και παράθυρο ιδιοτήτων.

IntelliJ IDEA

Το IntelliJ είναι ένα αρκετά δημοφιλές εμπορικό ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού, κυρίως για Java εφαρμογές. Εκτός της Java παρέχει υποστήριξη και για άλλες γλώσσες προγραμματισμού όπως JavaScript, HTML/L/XHTML/CSS, XML/XSL, Ruby. Επιπλέον της βασικής έκδοσης της Java υποστηρίζεται η χρήση πολλών τεχνολογιών και περιβάλλοντων εργασίας όπως Spring, Hibernate, Web Services, Struts, JSP, JSF, EJB, AJAX. Στο πακέτο παρέχεται ένας αναπτυγμένος κειμενογράφος ο οποίος προσφέρει κάποιες πολύ χρήσιμες υπηρεσίες. Τέτοιες είναι η αυτόματη παραγωγή κώδικα, η ανάλυση κώδικα κατά την ώρα πληκτρολόγησης, ο αυτόματος σχολιασμός σύμφωνα με το πρότυπο της Java και ο έλεγχος λαθών και πιθανών παραλήψεων κατά την ώρα πληκτρολόγησης.

Το IntelliJ , όπως και τα περισσότερα IDEs, προσφέρει πολλά εργαλεία που διευκολύνουν τη συγγραφή κώδικα. Η αυτόματη παραγωγή κώδικα, όπως η δημιουργία μεθόδων δημιουργών σε κλάσεις, η ανάλυση και ολοκλήρωση κώδικα, ο έλεγχος λαθών κατά την διάρκεια της πληκτρολόγησης, ο αυτόματος σχολιασμός σύμφωνα με το πρότυπο της Java είναι λειτουργίες που βοηθούν τον προγραμματιστή να αυξήσει την παραγωγικότητα του και μειώνουν τον κίνδυνο λαθών.

Επιπλέον δίνεται η δυνατότητα χρήσης εργαλείων που βοηθούν την παρακολούθηση της πορείας ανάπτυξης του κώδικα καθ' όλη τη διαδικασία παραγωγής. Υποστηρίζεται η χρήση αρκετών γνωστών συστημάτων διαχείρισης εκδόσεων (Version Control Systems), ενώ παράλληλα παρέχεται ένα ισχυρό εργαλείο παρακολούθησης της τοπικής ιστορίας. Επιπλέον, υπάρχουν ενσωματωμένα περιβάλλοντα για το χτίσιμο της εφαρμογής (Ant/Maven), για τον έλεγχο της εφαρμογής μέσω δοκιμών μονάδας JunIt και συστήματα διαχείρισης εκδόσεων, ενώ τέλος προσφέρεται ένα πολύ εύχρηστο περιβάλλον μεταγλώττισης, εκτέλεσης και αποσφαλμάτωσης (debugging).

NetBeans IDE

Το NetBeans IDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών που βασίζεται στην πλατφόρμα NetBeans. Η πλατφόρμα NetBeans IDE επιτρέπει την ανάπτυξη της εφαρμογής NetBeans IDE από ένα σύνολο συστατικών μονάδων λογισμικού που ονομάζονται modules. Ένα module είναι ένα συμπιεσμένο αρχείο Java το οποίο περιέχει κλάσεις, που είναι προορισμένες να αλληλεπιδρούν με NetBeans Open APIs, και ένα αρχείο που το χαρακτηρίζει ως module. Εφαρμογές που έχουν αναπτυχθεί πάνω σε modules μπορούν να επεκταθούν προσθέτοντας νέα modules.

Το NetBeans IDE διανέμεται δωρεάν και είναι λογισμικό ανοιχτού κώδικα. Υποστηρίζει την ανάπτυξη όλων των ειδών εφαρμογών Java (Java SE, web, EJB και mobile εφαρμογές). Επιπλέον, υποστηρίζει την ανάπτυξη εφαρμογών με την χρήση του Ant τον έλεγχο εκδόσεων, με την χρήση των CVS, Subversion, Mercurial και Clearcase, και την επεξεργασία κώδικα (refactoring).

Όλες οι λειτουργίες του IDE παρέχονται από τα modules. Κάθε module παρέχει μία απαιτούμενη λειτουργία όπως υποστήριξη ανάπτυξης εφαρμογών Java, επεξεργασία κώδικα, υποστήριξη συστημάτων διαχείρισης εκδόσεων CVS και SVN. Το NetBeans IDE παρέχει όλα τα απαιτούμενα modules για την ανάπτυξη εφαρμογών Java στο αρχικό πακέτο που κατεβάζει κανείς, το οποίο επιτρέπει στον χρήστη να αρχίσει άμεσα την ανάπτυξη μιας εφαρμογής. Τα modules επιτρέπουν την επέκταση του IDE. Νέες λειτουργίες, όπως η υποστήριξη επιπλέον γλωσσών προγραμματισμού μπορούν να προστεθούν μετά την αρχική εγκατάσταση.

Jbuilder

Είναι προϊόν της εταιρίας Borland και είναι πολύ δημοφιλές μεταξύ των προγραμματιστών της java . Με τον jbuilder μπορεί κανείς να δημιουργήσει επαγγελματικές εφαρμογές Java που να περιλαμβάνουν και προηγμένη διαχείριση βάσεων δεδομένων. Όπως όλα τα προγράμματα IDE ο Jbuilder περιλαμβάνει ενσωματωμένο συντάκτη (editor), debugger και compiler. Στο διαδίκτυο κυκλοφορούν διαφορετικές εκδόσεις του jbuilder όπως “lite “ ή “university” οι οποίες διατίθενται δωρεάν και περιλαμβάνουν διαφορετικά πακέτα λειτουργιών ανάλογα με τις απαιτήσεις του χρήστη.

Eclipse

Το Eclipse είναι μία πλατφόρμα ανάπτυξης λογισμικού η οποία διατίθεται δωρεάν, είναι ανοιχτού κώδικα και αποτελείται από ένα IDE και ένα plug-in σύστημα ώστε να είναι εύκολη η επέκτασή της. Υποστηρίζει πολλές γλώσσες προγραμματισμού κάποιες από τις οποίες είναι: C, C++, COBOL, Python, Perl, PHP. Η ανάπτυξή του έχει γίνει σε γλώσσα Java και χρησιμοποιείται για την ανάπτυξη εφαρμογών σε Java, αλλά με την χρήση διάφορων plug-ins, και σε άλλες γλώσσες όπως: C, C++, COBOL, Python, PHP.

Το Eclipse χρησιμοποιεί plug-ins για να παρέχει περισσότερες λειτουργίες πέρα από το σύστημα runtime, σε αντίθεση με άλλες εφαρμογές που η λειτουργικότητα τους είναι καθορισμένη. Ο μηχανισμός plug-in που χρησιμοποιεί είναι ένα ελαφρύ πλαίσιο εργασίας που αποτελείται από πολλές συνιστώσες. Εκτός από την δυνατότητα που δίνει στο Eclipse να επεκτείνει τη χρήση του και σε άλλες γλώσσες προγραμματισμού, επιτρέπει την εργασία με: στοιχειοθετικές γλώσσες όπως η LaTeX, διαδικτυακές εφαρμογές όπως η telnet, και συστήματα διαχείρισης βάσεων δεδομένων. Η αρχιτεκτονική του plug-in υποστηρίζει την συγγραφή αρχείων οποιαδήποτε κατάληξης στο περιβάλλον του Eclipse. Το Eclipse υποστηρίζει από την αρχή τη γλώσσα Java και το σύστημα

διαχείρισης εκδόσεων CVS, με την υποστήριξη του Subversion να είναι διαθέσιμη από εξωτερικά plug-ins.

Το Eclipse SDK περιλαμβάνει τα Eclipse Development Tools, τα οποία παρέχουν ένα IDE με ενσωματωμένο μεταφραστή της Java και με ένα πλήρες μοντέλο των αρχείων πηγαίου κώδικα της Java. Αυτό επιτρέπει την χρήση προχωρημένων τεχνικών αλλαγής και ανάλυσης κώδικα. Επιπλέον, το IDE χρησιμοποιεί ένα workspace, το οποίο είναι ένα σύνολο μεταδεδομένων για ένα καθορισμένο αριθμό αρχείων, επιτρέποντας τις τροποποιήσεις των αρχείων έξω από το IDE εφόσον το αντίστοιχο workspace ενημερωθεί αμέσως μετά τις αλλαγές.

Visual J++

Η Visual J++ είναι η απάντηση της εταιρίας Microsoft στα IDE εφαρμογών Java. Περιλαμβάνει εργαλεία οδηγών για την ανάπτυξη applet, class browser, εξελιγμένο συντάκτη πηγαίου κώδικα, οπτικό εκσφαλματωτή, και άλλα. Η έκδοση professional περιλαμβάνει μεταξύ άλλων και υποστήριξη βάσεων δεδομένων. Η Visual J++ αποτελεί ταυτόχρονα υπερσύνολο και υποσύνολο της καθαρής Java δηλαδή δεν περιλαμβάνει όλα τα χαρακτηριστικά της καθαρής Java ενώ περιέχει ένα σύνολο δικών της χαρακτηριστικών.

IDE	Άδεια χρήσης	Υποστηριζόμενα Λειτουργικά			Λειτουργίες		
		Windows	Linux	Mac	Profiler	GUI builder	Debugger
IntelliJ IDEA	Αγορά	Ναί	Ναί	Ναί	Ναί	Όχι	Ναί
Eclipse	Δωρεάν	Ναί	Ναί	Ναί	Όχι	Ναί	Ναί
Ms Visual Studio	Αγορά	Ναί	Όχι	Όχι	Ναί	Ναί	Ναί
Jbuilder	Αγορά	Ναί	Ναί	Ναί	Όχι	Ναί	Όχι
Netbeans	Δωρεάν	Ναί	Ναί	Ναί	Ναί	Ναί	Ναί

Πίνακας 2: Συγκριτικά IDE

2.3 Αντικειμενοστραφής προγραμματισμός

Ο Αντικειμενοστραφής Προγραμματισμός (Object-Oriented Programming ή OOP για συντομία), όπως λέει και το όνομά του συνδέεται άμεσα με τα αντικείμενα και κατά συνέπεια, όπως θα δούμε και στη συνέχεια, και με τις κλάσεις. Κατ' ουσίαν, ο OOP είναι ένας τρόπος οργάνωσης των προγραμμάτων που γράφουμε. Ο τρόπος αυτός οργάνωσης, δεν είναι φυσικά ούτε μοναδικός, ούτε καν ο βέλτιστος. Άλλοι τρόποι οργάνωσης είναι ο διαδικαστικός (procedural ή imperative) και ο συναρτησιακός (functional). Συνηθίζεται, αυτοί οι τρόποι οργάνωσης ή τεχνικές οργάνωσης των προγραμμάτων να ονομάζονται προγραμματιστικά παραδείγματα (programming paradigms). Η επιλογή του προγραμματιστικού παραδείγματος το οποίο θα χρησιμοποιήσουμε εξαρτάται από το πρόβλημα το οποίο καλούμαστε να επιλύσουμε, αλλά και από τη γλώσσα προγραμματισμού την οποία χρησιμοποιούμε. Δεν επιτρέπουν όλες οι γλώσσες προγραμματισμού τη χρήση όλων των προγραμματιστικών παραδειγμάτων. Για παράδειγμα υπάρχουν γλώσσες που επιτρέπουν τη δημιουργία μόνο διαδικαστικών προγραμμάτων όπως η Fortran. Υπάρχουν άλλες που επιτρέπουν τη δημιουργία μόνο (ή κυρίως) συναρτησιακών προγραμμάτων, όπως η Lisp και οι πολλές παραλλαγές της. Ενώ υπάρχουν και γλώσσες που επιτρέπουν, με περισσότερη ή λιγότερη ευκολία, να εφαρμόσεις περισσότερα από ένα προγραμματιστικά παραδείγματα.

Το πιο χαρακτηριστικό ίσως παράδειγμα προβλήματος που προσφέρεται για λύση μέσω αντικειμενοστραφούς προγραμματισμού είναι ο σχεδιασμός GUI. Μία από τις πλέον συνήθειες εκφράσεις είναι ότι “τα πάντα είναι αντικείμενα” (everything is an object). Τα αντικείμενα (objects) δημιουργούνται από τις κλάσεις (classes). Για την ακρίβεια μάλιστα, οι κλάσεις ορίζονται ως εργοστάσια αντικειμένων (object factories). Είναι δηλαδή, τα στοιχεία εκείνα της γλώσσας, τα οποία κατασκευάζουν αντικείμενα (objects). Προκειμένου να εξηγήσουμε τον όρο αντικείμενο (object) συχνά βοηθάει να σκεφτόμαστε τα αντικείμενα (objects) ως κάτι ανάλογο με αυτό που ονομάζουμε στη γραμματική της φυσικής γλώσσας “ουσιαστικό”. Για να γίνει πιο σαφές αυτό ας σκεφτούμε ορισμένα ουσιαστικά. Καρέκλα, τραπέζι, σκύλος, γάτα, κύκλος, ορθογώνιο, οδηγός είναι ορισμένα

που πιθανά έρχονται στο μυαλό. Αν προσπαθήσουμε να δούμε το κοινό σημείο όλων των παραπάνω θα δούμε ότι πρόκειται για οντότητες (έμψυχες ή άψυχες) οι οποίες έχουν συγκεκριμένες ιδιότητες ή/και μπορούν να εκτελούν συγκεκριμένες ενέργειες. Μερικά παραδείγματα:

- Μία καρέκλα έχει τις ιδιότητες χρώμα, ύψος, υλικό κτλ.
 - Ένα ορθογώνιο αντίστοιχα έχει τις ιδιότητες πλάτος, ύψος, εμβαδόν, περίμετρος κτλ.
 - Μία γάτα έχει τις ιδιότητες όνομα, ηλικία, φύλλο αλλά επίσης μπορεί και να εκτελεί διάφορες ενέργειες όπως για παράδειγμα να νιαουρίσει, να τρέξει, να περπατήσει, να σκαφαλώσει, να φάει κτλ.
 - Ένας οδηγός μπορεί να προβεί σε διάφορες ενέργειες, όπως πχ να στρίψει, να φρενάρει, να επιταχύνει, να βάλει μπροστά τη μηχανή κτλ.
- Όπως γίνεται σαφές από τα παραπάνω, σε όρους φυσικής γλώσσας, οι ιδιότητες των αντικειμένων είναι άλλα ουσιαστικά, ενώ οι ενέργειες που μπορούν να εκτελέσουν είναι ρήματα.

Ο παραπάνω “ορισμός” των αντικειμένων είναι φυσικά πολύ γενικός και δύσκολα θα άντεχε σε ενδελεχή εξέταση από έναν φιλόλογο. Παρόλα αυτά, είναι ένας ορισμός που μας βολεύει ιδιαίτερα όταν επιστρέφουμε στον κόσμο του Αντικειμενοστραφούς Προγραμματισμού και αυτό γιατί τα αντικείμενα (objects) είναι εκείνες οι “οντότητες” οι οποίες μας επιτρέπουν να περιγράψουμε κάθε τι που έχει ιδιότητες και μπορεί να εκτελεί ενέργειες. Με λίγα λόγια δηλαδή, μέσω των αντικειμένων μπορούμε να περιγράψουμε πρακτικά σχεδόν οτιδήποτε συναντάμε στον υλικό κόσμο. Οι ιδιότητες ενός αντικειμένου ονομάζονται attributes, ενώ οι ενέργειές που μπορεί να εκτελέσει ονομάζονται methods (μέθοδοι).

2.3.1 Κλάσεις (Classes)

Μια κλάση δεν είναι τίποτα άλλο παρά ένας ορισμός. Αυτό που ορίζει είναι το ποιες ιδιότητες (attributes) και ποιες μεθόδους (methods) θα έχει ένα αντικείμενο. Η πρώτη γραμμή είναι αυτή στην οποία ορίζεται το όνομα της κλάσης. Οι γραμμές που ακολουθούν, αποτελούν το σώμα της κλάσης (class body). Στη συγκεκριμένη περίπτωση, ορίσαμε ότι οι γάτες που θα δημιουργηθούν από την κλάση αυτή, θα έχουν 3 ιδιότητες (όνομα, ηλικία και φύλο) και θα μπορούν να εκτελούν 2 ενέργειες (θα έχουν 2 μεθόδους δηλαδή), να τρώνε και να κοιμούνται. Όλες οι γάτες που θα δημιουργηθούν από την κλάση αυτή θα έχουν μόνο αυτές τις ιδιότητες (attributes) και θα μπορούν να εκτελούν μόνο τις συγκεκριμένες ενέργειες. Θα μπορούσαμε φυσικά να κάνουμε την κλάση μας πολύ πιο σύνθετη. Μια γάτα εξάλλου είναι ένας πολύ σύνθετος οργανισμός.

Η σύνταξη για τη δημιουργία των νέων αντικειμένων είναι πολύ απλή. Απλά χρησιμοποιούμε το όνομα της κλάσης και μέσα στις παρενθέσεις δίνουμε τις τιμές των ιδιοτήτων (attributes) της κλάσης. Με τον τρόπο αυτό, δίνοντας δηλαδή διαφορετικές τιμές στα attributes της κλάσης, μπορούμε να δημιουργήσουμε πολλές, διαφορετικές μεταξύ τους γάτες. Δεν υπάρχει κανένας περιορισμός στον αριθμό των νέων αντικειμένων που θα δημιουργήσουμε από μία κλάση. Μπορούμε να δημιουργήσουμε όσα νέα αντικείμενα θέλουμε.

Όλα τα νέα αντικείμενα θα έχουν ακριβώς τις ίδιες ιδιότητες (attributes) και τις ίδιες μεθόδους. Οι τιμές των ιδιοτήτων τους όμως θα είναι διαφορετικές μεταξύ τους. Μπορούμε φυσικά να δώσουμε ακριβώς τις ίδιες τιμές στις ιδιότητες, οπότε θα δημιουργηθούν δύο όμοια αντικείμενα, αλλά συνήθως δεν έχουμε λόγο να κάνουμε κάτι τέτοιο. Υπενθυμίζουμε ότι προηγουμένως ορίσαμε τις κλάσεις σαν εργοστάσιο αντικειμένων (factory object). Ακριβώς όπως ένα εργοστάσιο που φτιάχνει καρέκλες μπορεί να κατασκευάσει καρέκλες διαφορετικού σχεδίου, διαστάσεων και χρώματος σε όποια ποσότητα επιθυμεί, έτσι και οι κλάσεις μπορούν να κατασκευάσουν αντικείμενα με διαφορετικές ιδιότητες σε όποια ποσότητα επιθυμούμε. Στην ορολογία του Αντικειμενοστραφούς Προγραμματισμού, όλα τα νέα αντικείμενα που δημιουργούνται από μία κλάση ονομάζονται στιγμιότυπα (instances).

Όπως μία φωτογραφία ενός αθλητή που τρέχει αποτελεί την απεικόνιση μίας μόνο από όλες τις θέσεις που πέρασε κατά τη διάρκεια του αγώνα του, έτσι και μία instance αποτελεί μία μόνο αποτύπωση όλων των δυνατών συνδυασμών που μπορούν να έχουν οι ιδιότητες μιας κλάσης. Αν προσέξουμε τον ορισμό των μεθόδων μέσα στο σώμα της κλάσης, θα δούμε ότι δε διαφέρουν από τον

ορισμό των συναρτήσεων. Μία μέθοδος δεν είναι τίποτα άλλο παρά μία συνάρτηση που ανήκει σε ένα αντικείμενο. Μπορούμε να δώσουμε έξτρα ορίσματα (arguments) σε μία μέθοδο .

2.3.2 Κληρονομικότητα (Inheritance)

Ίσως η πλέον κεφαλαιώδης έννοια του Αντικειμενοστραφούς Προγραμματισμού είναι η κληρονομικότητα (inheritance). Με τον όρο κληρονομικότητα, εννοούμε τη δυνατότητα που έχει μια κλάση να κληρονομεί όλες τις ιδιότητες και τις μεθόδους μιας άλλης κλάσης. Χρησιμοποιώντας λίγο πιο επίσημη ορολογία, λέμε ότι η κλάση που ορίζεται πρώτη είναι η βασική κλάση (base class) και η κλάση που κληρονομεί τη βασική κλάση ονομάζεται παράγωγη κλάση (derived class). Εναλλακτικά οι βασικές κλάσεις ονομάζονται και υπερκλάσεις ενώ οι παράγωγες κλάσεις ονομάζονται υποκλάσεις.

Απλή vs Πολλαπλή Κληρονομικότητα (Single vs Multiple Inheritance)

Η διαφορά μεταξύ της απλής και της πολλαπλής κληρονομικότητας έγκειται στον αριθμό των υπερκλάσεων που έχει μία συγκεκριμένη υποκλάση. Αν κληρονομεί από μία μόνο υπερκλάση τότε μιλάμε για απλή κληρονομικότητα. Αν κληρονομεί από δύο ή περισσότερες υπερκλάσεις, τότε μιλάμε για πολλαπλή. Η πολλαπλή κληρονομικότητα είναι περίπλοκη. Η χρήση της πολλαπλής κληρονομικότητας είναι συνήθως ένδειξη κακού design. Αυτός είναι και ο λόγος που υπάρχουν πολλές γλώσσες προγραμματισμού που υποστηρίζουν μόνο απλή κληρονομικότητα (πχ Java). Στην πλειοψηφία των περιπτώσεων, υπάρχει απλούστερος τρόπος να κάνουμε αυτό που θέλουμε από το να καταφύγουμε στην πολλαπλή κληρονομικότητα. Στο σημείο αυτό, οφείλουμε να διασαφηνίσουμε ότι μία αλληλουχία κλάσεων που διαδοχικά κληρονομούν η μία την άλλη δεν είναι πολλαπλή κληρονομικότητα.

Η (απλή) Κληρονομικότητα χρησιμοποιείται όταν έχουμε μια ακολουθία αντικειμένων τα οποία πηγάζουν από το γενικότερο στο ειδικότερο. Αν μελετήσουμε τις σχέσεις μεταξύ των διαδοχικών κλάσεων θα δούμε ότι η “κάθε υποκλάση είναι η υπερκλάση της”. Δηλαδή:

- Μία Γάτα είναι Αιλουροειδές.
- Ένα Αιλουροειδές είναι Θηλαστικό.
- Ένα Θηλαστικό είναι Ζώο.

Η σχέση αυτή ισχύει όχι μόνο για την αμέσως ανώτερη υπερκλάση, αλλά για κάθε υπερκλάση. Δηλαδή:

- Μία Γάτα είναι Αιλουροειδές.
- Μια Γάτα είναι και Θηλαστικό.
- Μια Γάτα είναι και Ζώο.

Κάθε φορά που έχουμε μια παρόμοια σχέση μεταξύ αντικειμένων, δηλαδή μπορούμε να πούμε ότι κάτι είναι κάτι άλλο (is-a relationship) τότε μπορούμε/πρέπει να χρησιμοποιήσουμε κληρονομικότητα. Αν έχουμε να ασχοληθούμε μόνο με αντικείμενα Γάτας τότε δεν μας χρειάζονται όλα αυτά. Αν όμως έχουμε να κατασκευάσουμε Γάτες, Λιοντάρια, Τίγρεις κτλ τότε τα οφέλη αρχίζουν να φαίνονται. Ορίζουμε αρχικά τα κοινά στοιχεία όλων αυτών των κλάσεων στην υπερκλάση Αιλουροειδές, την κληρονομούμε και την εξειδικεύουμε καταλλήλως στις υποκλάσεις. Αν μάλιστα εκτός από Αιλουροειδή έχουμε να κατασκευάσουμε και αντικείμενα Σκύλου, Αλεπούς, Λύκου, αλλά και Αρκούδας, Αλόγου, Φώκιας κτλ τότε η κλάση Θηλαστικό κατευθείαν αποκτάει νόημα. Με τον ίδιο τρόπο, αν στο πρόγραμμά μας χρειαζόμαστε και Ψάρια ή Έντομα κτλ τότε η υπερκλάση Ζώο βρίσκει και αυτή τη θέση της. Η κάθε μία από τις κλάσεις Ψάρι, Έντομο κτλ θα έχει φυσικά το δικό της ιεραρχικό δέντρο, στο οποίο θα εξειδικεύεται καταλλήλως.

2.3.3 Σύνθεση (Composition)

Οι ιδιότητες (attributes) ενός αντικείμενου είναι και αυτές με τη σειρά τους αντικείμενα. Η παρατήρηση αυτή είναι η βασική ιδέα πίσω από την τεχνική της Σύνθεσης. Τα πλεονεκτήματα του Αντικειμενοστραφούς Προγραμματισμού χωρίζονται σε οφέλη σε επίπεδο κώδικα και σε οφέλη σε επίπεδο οργάνωσης. Χάρης στην κληρονομικότητα αποφεύγουμε να επαναλάβουμε κώδικα (αρχή “Do not Repeat Yourself” - DRY). Αυτό γίνεται γιατί όλα τα αντικείμενα μας μοιράζονται τον ίδιο κώδικα. Οι κοινές μέθοδοι ορίζονται μία φορά στην υπερκλάση και όλες οι υποκλάσεις απλά τη

κληρονομούν. Με τον τρόπο αυτό μειώνεται το μέγεθος του κώδικα. Λιγότερος κώδικας σημαίνει λιγότερα bugs, πιο κατανοητός κώδικας και κώδικας ο οποίος είναι πιο εύκολο να συντηρηθεί. Ο OOP είναι μία προσέγγιση που μας επιτρέπει να φτάσουμε σε υψηλά επίπεδα αφαίρεσης (abstraction) μειώνοντας με αυτόν τον τρόπο την πολυπλοκότητα των προβλημάτων που καλούμαστε να λύσουμε.

2.3.4 Γλώσσες προγραμματισμού

Visual Basic

Η Visual Basic (VB) είναι γλώσσα προγραμματισμού τρίτης γενιάς, οδηγούμενη από συμβάντα (event driven) και έχει ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft για το μοντέλο προγραμματισμού COM. Η VB θεωρείται επίσης μία σχετικά εύκολη γλώσσα προγραμματισμού στην εκμάθηση και τη χρησιμοποίηση, λόγω των χαρακτηριστικών της, καθώς έχει Γραφικό Περιβάλλον Χρήστη και συγγένεια με την γλώσσα προγραμματισμού BASIC. Η Visual Basic προέρχεται από τη BASIC και επιτρέπει την ταχεία ανάπτυξη εφαρμογών (RAD) με Γραφικό Περιβάλλον Χρήστη (GUI), πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Data Access Objects, Remote Data Objects, ή ActiveX Data Objects), και τη δημιουργία στοιχείων ελέγχου ActiveX και αντικειμένων. Οι γλώσσες προγραμματισμού τύπου "scripting", όπως η VBA και VBScript συντακτικά είναι παρόμοιες με τη Visual Basic, αλλά έχουν διαφορετικές επιδόσεις. Ένας προγραμματιστής μπορεί να ολοκληρώσει μια εφαρμογή χρησιμοποιώντας τα στοιχεία που παρέχονται με την Visual Basic. Προγράμματα γραμμένα σε Visual Basic μπορούν, επίσης, να χρησιμοποιήσουν το Windows API, αλλά κάτι τέτοιο απαιτεί δηλώσεις εξωτερικών συναρτήσεων. Η τελική έκδοση 6 βγήκε το 1998. Η εκτεταμένη υποστήριξη της Microsoft έληξε το Μάρτιο του 2008 και ορίστηκε διάδοχος της η Visual Basic.NET (γνωστή απλά ως Visual Basic).

Όπως και η γλώσσα προγραμματισμού BASIC, η Visual Basic έχει σχεδιαστεί για να είναι εύκολη στην εκμάθηση και το χειρισμό. Η γλώσσα δεν επιτρέπει στους προγραμματιστές να δημιουργήσουν μόνο απλές εφαρμογές GUI, αλλά μπορούν, επίσης, να αναπτύξουν πολύπλοκες εφαρμογές. Ο προγραμματισμός σε VB συνίσταται από τον οπτικό συνδυασμό στοιχείων ή ελέγχων σε μια φόρμα, τον προσδιορισμό χαρακτηριστικών και ενεργειών αυτών των στοιχείων και την σύνταξη επιπλέον γραμμών κώδικα για αυξημένη λειτουργικότητα. Καθώς υπάρχουν προεπιλεγμένα χαρακτηριστικά και ενέργειες για τα επιμέρους στοιχεία, μπορεί να δημιουργηθεί ένα απλό πρόγραμμα χωρίς ο προγραμματιστής να γράψει πολλές γραμμές κώδικα. Στις προηγούμενες εκδόσεις υπήρχαν προβλήματα επιδόσεων, αλλά με τους ταχύτερους υπολογιστές και τη μεταγλώττιση εγγενούς κώδικα αυτό παύει να είναι ένα τόσο σημαντικό ζήτημα.

Αν και τα προγράμματα μπορούν να μετατραπούν σε εγγενή εκτελέσιμο κώδικα από την έκδοση 5 και μετά, αυτά εξακολουθούν να απαιτούν την παρουσία των βιβλιοθηκών χρόνου εκτέλεσης (runtime) με μέγεθος περίπου 1 MB. Οι βιβλιοθήκες runtime υπάρχουν στα Windows 2000 και αργότερα, αλλά στις παλαιότερες εκδόσεις των Windows όπως τα 95/98/NT πρέπει να διανέμονται μαζί με το εκτελέσιμο αρχείο.

Οι φόρμες δημιουργούνται χρησιμοποιώντας τεχνικές "σύρε κι άσε" (drag-and-drop). Χρησιμοποιείται ένα εργαλείο για την τοποθέτηση στοιχείων ελέγχου (π.χ. πλαίσια κειμένου, κουμπιά, κλπ.) στη φόρμα (παράθυρο). Τα στοιχεία ελέγχου έχουν χαρακτηριστικά και χειριστές συμβάντων συνδεδεμένους με αυτά. Οι προεπιλεγμένες τιμές παρέχονται όταν δημιουργείται το στοιχείο ελέγχου, αλλά μπορούν να τροποποιηθούν από τον προγραμματιστή. Πολλές τιμές χαρακτηριστικών είναι δυνατό να τροποποιηθούν κατά το χρόνο εκτέλεσης από ενέργειες του χρήστη ή αλλαγές του περιβάλλοντος, παρέχοντας έτσι μια δυναμική εφαρμογή. Για παράδειγμα, μπορεί να εισαχθεί κώδικας στον χειριστή συμβάντων αλλαγής διαστάσεων της φόρμας, ώστε ένα στοιχείο ελέγχου να παραμένει πάντα στο κέντρο της φόρμας ή να μεγαλώσει ώστε να την γεμίσει, κλπ. Με την προσθήκη κώδικα μέσα σε ένα χειριστή συμβάντων για το πάτημα των πλήκτρων σε ένα πλαίσιο κειμένου, το πρόγραμμα μπορεί αυτόματα να μετατρέψει το εισαγόμενο κείμενο σε κεφαλαία ή πεζά ή ακόμα και να εμποδίσει ορισμένους από τους χαρακτήρες να εμφανιστούν.

Με τη Visual Basic είναι δυνατή η δημιουργία εκτελέσιμων (EXE) αρχείων, στοιχείων ελέγχου ActiveX ή αρχείων DLL, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Πλαίσια διαλόγου με λιγότερες λειτουργίες μπορούν να χρησιμοποιηθούν για pop-up δυνατότητες. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισαγάγουν επιπλέον λογική

μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μια λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο. Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο, και στη συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεγθεί.

Εναλλακτικά, ένα συστατικό της Visual Basic μπορεί να μην έχει Γραφικό Περιβάλλον Χρήστη, αλλά αντ' αυτού, να παρέχει αντικείμενα ActiveX σε άλλα προγράμματα μέσω Component Object Model (COM). Αυτό επιτρέπει επεξεργασία στην πλευρά του διακομιστή (server-side processing) ή τη δημιουργία πρόσθετων μορφωμάτων (add-in module).

Η γλώσσα έχει αυτόματη διαχείριση μνήμης με την τεχνική της συλλογής απορριμμάτων χρησιμοποιώντας υπολογισμό αναφορών και έχει μια μεγάλη βιβλιοθήκη με βοηθητικά αντικείμενα καθώς και βασική αντικειμενοστραφή υποστήριξη. Από τα πιο κοινά στοιχεία που περιλαμβάνονται στο προεπιλεγμένο πρότυπο έργου, ο προγραμματιστής σπάνια χρειάζεται να καθορίσει πρόσθετες βιβλιοθήκες. Αντίθετα με πολλές άλλες γλώσσες προγραμματισμού η Visual Basic γενικά δεν διαχωρίζει τους πεζούς από τους κεφαλαίους χαρακτήρες, αν και θα μετατρέψει τις λέξεις-κλειδιά σε μία τυπική διαμόρφωση. Οι συγκρίσεις συμβολοσειρών διαχωρίζουν τα πεζά από τα κεφαλαία από προεπιλογή, αλλά μπορεί να αλλάξει αυτό, εφόσον το επιθυμείτε. Ο μεταγλωττιστής της Visual Basic είναι κοινός με τις άλλες γλώσσες του Visual Studio (C#, C++), αλλά οι περιορισμοί στον IDE δεν επιτρέπουν τη δημιουργία ορισμένων στόχων (μοντέλα Windows DLL) και σε μοντέλα νημάτων.

Χαρακτηριστικά της Visual Basic

Η Visual Basic έχει τα εξής χαρακτηριστικά που διαφέρουν από τις γλώσσες που προέρχονται από τη C:

- Η πολλαπλή ανάθεση που διατίθεται στη C δεν είναι δυνατή. Το $A = B = C$, δεν σημαίνει ότι οι τιμές των A, B και C είναι ίδιες. Το αποτέλεσμα της Boolean " $B = C$;" αποθηκεύεται στην μεταβλητή A. Το αποτέλεσμα που αποθηκεύεται στην A θα μπορούσε συνεπώς να είναι ψευδές (0) ή αληθές (-1).
- Οι λογικοί και δυαδικοί τελεστές είναι ενοποιημένοι. Αυτό έρχεται σε αντίθεση με ορισμένες γλώσσες προερχόμενες από τη C (όπως η γλώσσα Perl), οι οποίες έχουν ξεχωριστούς λογικούς και δυαδικούς τελεστές. Αυτό είναι ένα παραδοσιακό χαρακτηριστικό της γλώσσας BASIC.

HTML

Η HTML είναι το ακρωνύμιο των λέξεων HyperText Markup Language (γλώσσα μορφοποίηση υπερκειμένου) και είναι η βασική γλώσσα δόμηση σελίδων του World Wide Web (ή απλά ιστού: Web). Είναι μία γλώσσα προγραμματισμού. Χρησιμοποιείται για να σημαίνει ένα τμήμα κειμένου και να το κάνει να εμφανίζεται καλύτερα. Επιτρέπει την ενσωμάτωση ήχου και εικόνων στις web σελίδες. Αρχικά είχε κατασκευασθεί με σκοπό μόνο την μορφοποίηση κειμένου, αλλά μεγάλωσε και ενσωμάτωσε σχεδιαστικές τεχνικές κ.α. Η γλώσσα χρησιμοποιεί ένα αριθμό από tags για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (links) μετάβασης ανάμεσα των σελίδα, για την εισαγωγή εικόνων, ήχου κ.α. Όταν ένας Web Browser ανοίγει ένα αρχείο HTML τα στοιχεία (tags) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας.

Το 1990 ο Tim Berners-Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο θα μπορούσαν να μεταφέρονται κάθε είδος αρχείων και αντικειμένων μέσα από το Internet. Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW όπως το ξέρουμε σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσα HTML. Το 1994 αναπτύσσεται το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό (Internet Engineering Task Force). Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996). Η τελευταία περιελάμβανε πολλές από τις σημάνσεις (tags) που είχαν εισάγει οι δύο εταιρίες. Η έκδοση 4.0 παρουσιάστηκε τον Ιούνιο του 1997.

Οι ενδογενείς αδυναμίες της HTML, μιας κατεξοχήν γλώσσας μορφοποίησης υπερκειμένου, σύντομα οδήγησαν στην ανάπτυξη γλωσσών προγραμματισμού στο Internet. Η χρήση των γλωσσών προγραμματισμού κρίθηκε απαραίτητη αφού η εξάπλωση του Internet και η χρήση των σελίδων HTML για αλληλεπίδραση με τους χρήστες (δυναμική αποστολή και λήψη δεδομένων) δεν μπορούσε

να πραγματοποιηθεί μέσω της HTML. Αρχικά, αναπτύχθηκε από την εταιρεία Netscape η γλώσσα JavaScript η οποία, όπως και η HTML μεταφράζεται από τον Web Browser κατά την εμφάνιση της σελίδας. Η Microsoft ανέπτυξε απ' την πλευρά της μια δική της έκδοση της γλώσσας JavaScript την οποία ονόμασε JScript και μια έκδοση της γλώσσας Basic ειδικά για το Internet που ονόμασε VBScript.

Έτσι πλέον με την χρήση των παραπάνω γλωσσών είναι δυνατό να ελεγχθούν και να προγραμματιστούν όλα σχεδόν τα αντικείμενα που μπορεί να περιέχει μία ιστοσελίδα και με τη χρήση είτε εντολών διαδικαστικού χαρακτήρα, είτε εντολών διακλάδωσης, όπως για παράδειγμα η δομή [if ... then... else...]. Η χρήση τέτοιου είδους "προγραμμάτων" ή όπως ονομάζονται "scripts" είναι διαδικασία που έχει μεγάλες απαιτήσεις σε προγραμματισμό και απευθύνεται σε προγραμματιστές και μόνο. Η δημιουργία αρχείων HTML είναι πολύ απλή. Αρκεί να "τρέξουμε" έναν οποιοδήποτε διορθωτή κειμένου text, όπως το Notepad των Windows. Να γράψουμε τον κώδικα HTML που επιθυμούμε και να το αποθηκεύσουμε (σώσουμε) σε ένα αρχείο με κατάληξη .htm ή .html. Συνήθως αποθηκεύουμε με επέκταση .html όταν θα δημοσιεύσουμε τα αρχεία μας σε διακομιστή Unix. Αντίθετα ο εξυπηρετητής Windows NT καταλαβαίνει τα αρχεία και σαν .htm και σαν .html.

2.4 Εισαγωγή στις Βάσεις Δεδομένων

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Είναι κοινός τόπος σήμερα η εκτίμηση ότι το αγαθό της πληροφορίας είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους, ώστε να είναι πιο αποδοτικοί, ανταγωνιστικοί αλλά και παραγωγικοί στην εργασία τους.

Μια Βάση Δεδομένων (ΒΔ) είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, έτσι ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες ταυτόχρονα. Υπάρχει ένα ειδικό λογισμικό το οποίο μεσολαβεί ανάμεσα στις αρχεία δεδομένων και τις εφαρμογές που χρησιμοποιούν οι χρήστες και αποκαλείται Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ) ή DBMS (Data Base Management System). Το ΣΔΒΔ είναι στην ουσία ένα σύνολο από προγράμματα και υπορουτίνες που έχουν να κάνουν με τον χειρισμό της βάσης δεδομένων, όσον αφορά τη δημιουργία, τροποποίηση, διαγραφή στοιχείων, με ελέγχους ασφαλείας κ.ά.

Τα συστήματα βάσεων δεδομένων τα χρησιμοποιούμε για να μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά. Όσο η επιστήμη της Πληροφορικής εξελίσσεται τόσο περισσότερο απαραίτητες είναι οι Βάσεις Δεδομένων, την χρήση τους την βλέπουμε καθημερινά σε όλους τους τομείς της ζωής μας, όπως σε νοσοκομεία όπου καταχωρούνται οι ασθενείς και η συνταγογράφηση που τους γίνεται, στις δημόσιες υπηρεσίες, π.χ. εφορία που καταχωρούνται όλοι οι φορολογούμενοι πολίτες και τα οικονομικά τους στοιχεία, στα video clubs όπου καταχωρούνται σε μια Βάση Δεδομένων όλοι οι πελάτες καθώς και οι ταινίες του και οι ηθοποιοί που παίζουν.

2.4.1 Τα Δεδομένα και οι Πληροφορίες

Με τον όρο πληροφορία αναφερόμαστε συνήθως σε ειδήσεις, γεγονότα και έννοιες που αποκτάμε από την καθημερινή μας επικοινωνία και τα θεωρούμε ως αποκτηθείσα γνώση, ενώ τα δεδομένα μπορούν να είναι μη κατάλληλα επεξεργασμένα και μη ταξινομημένα σύνολα πληροφοριών.

Ένας αυστηρός ορισμός για το τι είναι δεδομένα και τι είναι πληροφορία έχει δοθεί σύμφωνα με την επιτροπή ANSI των ΗΠΑ. Δεδομένα (data) είναι μια παράσταση, όπως γράμματα, αριθμοί, σύμβολα κ.ά. στα οποία μπορούμε να δώσουμε κάποια σημασία (έννοια). Πληροφορία (information) είναι η σημασία που δίνουμε σ' ένα σύνολο από δεδομένα, τα οποία μπορούμε να επεξεργαστούμε βάσει προκαθορισμένων κανόνων και να βγάλουμε έτσι κάποια χρήσιμα συμπεράσματα. Με τις πληροφορίες περιορίζεται η αβεβαιότητα που έχουμε για διάφορα πράγματα και οδηγούμαστε έτσι στο να λάβουμε σωστές αποφάσεις.

Τα δεδομένα μπορούν να θεωρηθούν ως τρόποι αναπαράστασης εννοιών και γεγονότων που μπορούν να υποστούν διαχείριση και επεξεργασία. Η συλλογή και αποθήκευση ενός τεράστιου όγκου δεδομένων όπως απαιτούν οι κοινωνικές συνθήκες σήμερα, δεν λύνει τελείως το πρόβλημα της

σωστής οργάνωσης και ταξινόμησης των δεδομένων. Τα δεδομένα θα πρέπει να οργανωθούν με τέτοιο τρόπο έτσι ώστε να μπορούμε να τα εντοπίζουμε και να τα αξιοποιούμε εύκολα και γρήγορα και τη στιγμή που τα χρειαζόμαστε.

2.4.2 Η Οργάνωση Αρχείων

Ο πιο γνωστός τρόπος οργάνωσης δεδομένων με τη χρήση ηλεκτρονικών υπολογιστών είναι σε αρχεία εγγραφών. Για να κατανοήσουμε καλύτερα ορισμένες έννοιες, θα εξετάσουμε την περίπτωση ενός αρχείου πελατών και παραγγελιών μιας εμπορικής επιχείρησης. Για να οργανώσουμε σωστά το αρχείο μας, θα πρέπει να δημιουργήσουμε καρτέλες για τους πελάτες, αλλά και για τις παραγγελίες τους αργότερα, που θα πρέπει να περιέχουν τα εξής στοιχεία ανά πελάτη : Κωδικός, Επώνυμο, Όνομα, Διεύθυνση, ΤΚ, Πόλη, Τηλέφωνο, ΑΦΜ.

Ένα αρχείο (file) θα μπορούμε να το χαρακτηρίσουμε σαν ένα σύνολο που αποτελείται από οργανωμένα ομοειδή στοιχεία. Τα στοιχεία ενός αρχείου μπορούμε να τα οργανώσουμε σε λογικές ενότητες και το σύνολο των στοιχείων που περιέχει μια λογική ενότητα καλείται εγγραφή (record). Το κάθε στοιχείο της εγγραφής καλείται πεδίο (field). Το πεδίο αποτελεί και τη μικρότερη δυνατή υποδιαίρεση των στοιχείων ενός αρχείου. Ένα πεδίο χαρακτηρίζεται από τον μέγιστο αριθμό των χαρακτήρων (bytes) που απαιτούνται για την καταχώρησή του στη μνήμη του υπολογιστή και που αποκαλείται μήκος του πεδίου (field length).

Σε μια οργάνωση αρχείου όπως είναι οι πελάτες μιας εμπορικής επιχείρησης που είδαμε νωρίτερα, τα αντίστοιχα πεδία όλων των εγγραφών καταλαμβάνουν τον ίδιο αριθμό σε bytes που είναι αυτός που έχουμε ορίσει κατά τη δημιουργία του αρχείου. Για παράδειγμα, αν ορίσαμε ότι το πεδίο Επώνυμο θα έχει μήκος 15 χαρακτήρες, τότε το πεδίο της εγγραφής του πελάτη με επώνυμο Παπαδόπουλος, αλλά και το πεδίο της εγγραφής του πελάτη με επώνυμο Βες θα καταλαμβάνουν από 15 bytes στη μνήμη του υπολογιστή, ενώ αν ένας πελάτης ονομάζεται Παπαχριστοδουλόπουλος, τότε θα γίνει αποκοπή του επωνύμου του και θα καταχωρηθούν στη μνήμη του υπολογιστή μόνο τα 15 πρώτα γράμματα, δηλ. τα Παπαχριστοδουλό. Ένα πεδίο χαρακτηρίζεται ακόμη και από το είδος των δεδομένων που μπορεί να περιέχει, όπως:

- Αλφαριθμητικό (alphanumeric), μπορεί να περιέχει γράμματα, ψηφία ή και ειδικούς χαρακτήρες.
- Αριθμητικό (numeric), μπορεί να περιέχει μόνο αριθμούς.
- Αλφαβητικό (alphabetic), μπορεί να περιέχει μόνο γράμματα (αλφαβητικούς χαρακτήρες).
- Ημερομηνίας (date), μπορεί να περιέχει μόνο ημερομηνίες.
- Δυαδικό (binary), μπορεί να περιέχει ειδικού τύπου δεδομένα, όπως εικόνες, ήχους
- Λογικό (logical), μπορεί να περιέχει μόνο μία από δύο τιμές, οι οποίες αντιστοιχούν σε δύο διακριτές καταστάσεις και μπορούν να χαρακτηρισθούν σαν 0 και 1 ή σαν αληθές (true) και ψευδές (false).
- Σημειώσεων (memo), μπορεί να περιέχει κείμενο με μεταβλητό μήκος, το οποίο μπορεί να είναι και αρκετά μεγάλο και συνήθως αποθηκεύεται σαν ξεχωριστό αρχείο από το κύριο αρχείο.

Όσον αφορά τις εγγραφές, χρήσιμοι ορισμοί είναι οι εξής :

- Μήκος εγγραφής (record length) καλείται το άθροισμα που προκύπτει από τα μήκη των πεδίων που την αποτελούν.
- Δομή εγγραφής (record layout) ή γραμμογράφηση καλείται ο τρόπος με τον οποίο οργανώνουμε τα πεδία μιας εγγραφής.
- Διάβασμα (read) από αρχείο σημαίνει τη μεταφορά των δεδομένων του αρχείου, που γίνεται συνήθως ανά μία εγγραφή, από το μέσο αποθήκευσης (σκληρό δίσκο ή δισκέτα) στην κεντρική μνήμη του υπολογιστή για επεξεργασία.
- Γράψιμο (write) σε αρχείο σημαίνει μεταφορά των δεδομένων του αρχείου, που γίνεται συνήθως ανά μία εγγραφή, από την κεντρική μνήμη του υπολογιστή στο μέσο αποθήκευσης.

2.4.3 Προβλήματα της Οργάνωσης Αρχείων

Στα αρχικά στάδια της οργάνωσης αρχείων, ήταν πολύ συνηθισμένη πρακτική η δημιουργία ξεχωριστών εφαρμογών (προγραμμάτων) και ξεχωριστών αρχείων, όπως για παράδειγμα η δημιουργία ενός αρχείου πελατών και ενός άλλου ανεξάρτητου αρχείου για τις παραγγελίες των πελατών. Από την πρακτική αυτή προέκυψαν ορισμένα προβλήματα.

- **Πλεονασμός των δεδομένων** (data redundancy). Υπάρχει η περίπτωση να έχουμε επανάληψη των ίδιων δεδομένων σε αρχεία διαφορετικών εφαρμογών. Για παράδειγμα, αν έχουμε ένα αρχείο πελατών και ένα αρχείο παραγγελιών αυτών των πελατών, είναι σχεδόν σίγουρο ότι θα υπάρχουν κάποια στοιχεία των πελατών που θα υπάρχουν και στα δύο αρχεία.
- **Ασυνέπεια των δεδομένων** (data inconsistency). Αυτό μπορεί να συμβεί όταν υπάρχουν τα ίδια στοιχεία των πελατών (πλεονασμός) και στο αρχείο πελατών και στο αρχείο παραγγελιών και χρειασθεί να γίνει κάποια αλλαγή στη διεύθυνση ή στα τηλέφωνα κάποιου πελάτη, οπότε είναι πολύ πιθανό να γίνει η διόρθωση μόνο στο ένα αρχείο και όχι και στο άλλο.
- **Αδυναμία μερισμού δεδομένων** (data sharing). Μερισμός δεδομένων σημαίνει δυνατότητα για κοινή χρήση των στοιχείων κάποιων αρχείων. Για παράδειγμα, ο μερισμός δεδομένων θα ήταν χρήσιμος αν με την παραγγελία ενός πελάτη μπορούμε να έχουμε πρόσβαση την ίδια στιγμή στο αρχείο πελατών για να δούμε το υπόλοιπο του πελάτη και μετά στο αρχείο της αποθήκης για να δούμε αν είναι διαθέσιμα τα προϊόντα που παρήγγειλε ο συγκεκριμένος πελάτης. Η αδυναμία μερισμού δεδομένων δημιουργεί καθυστέρηση στη λήψη αποφάσεων και στην εξυπηρέτηση των χρηστών.
- **Αδυναμία προτυποποίησης**. Έχει να κάνει με την ανομοιομορφία και με την διαφορετική αναπαράσταση και οργάνωση των δεδομένων στα αρχεία των εφαρμογών. Η αδυναμία αυτή δημιουργεί προβλήματα προσαρμογής των χρηστών καθώς και προβλήματα στην ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων.

2.4.4 Οι Βάσεις Δεδομένων και τα ΣΔΒΔ (DBMS)

Για να δοθεί μια λύση σ' όλα τα παραπάνω προβλήματα, και με βάση το γεγονός ότι η χρήση των ηλεκτρονικών υπολογιστών και συνεπώς η ηλεκτρονική καταχώρηση και επεξεργασία δεδομένων αυξήθηκε κατακόρυφα ήδη από τη δεκαετία του '70 στις μεγάλες επιχειρήσεις και άρα είχαμε πάρα πολλές εφαρμογές να επεξεργάζονται δεδομένα σε πάρα πολλά αρχεία ταυτόχρονα, προτάθηκε η συνένωση όλων των αρχείων μιας εφαρμογής. Εκτός, όμως, από τη συνένωση των αρχείων, απαιτείται και μια σωστή οργάνωσή τους. Δημιουργήθηκαν έτσι οι Τράπεζες Πληροφοριών ή Βάσεις Δεδομένων (Data Bases).

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ' αυτήν. Το ΣΔΒΔ παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του ΣΔΒΔ μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα ΣΔΒΔ μπορεί να είναι εγκατεστημένο σ' έναν μόνο υπολογιστή ή και σ' ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν χρήστη ή και από πολλούς χρήστες.

Ένα Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή DBS (Data Base System) αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή.

- Το **υλικό** (hardware) αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το **λογισμικό** (software) είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.
- Η **βάση δεδομένων** (data base) αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των

αρχείων αυτών είναι ενοποιημένα (data integration), δηλ. δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και μερισμένα (data sharing), δηλ. υπάρχει δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.

- Οι **χρήστες** (users) μιας βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :
- Τελικοί χρήστες (end users). Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τούς επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
- **Προγραμματιστές εφαρμογών** (application programmers) αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
- **Διαχειριστής δεδομένων** (data administrator – DA). Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
- **Διαχειριστής βάσης δεδομένων** (database administrator – DBA). Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΔΒΔ.

2.4.5 Η Αρχιτεκτονική των ΣΔΒΔ

Όπως είδαμε νωρίτερα, ένα ΣΔΒΔ (Σύστημα Διαχείρισης Βάσης Δεδομένων) έχει σαν αποστολή τη διαχείριση των δεδομένων των αρχείων της βάσης, δηλ. την προσθήκη, διαγραφή, τροποποίηση εγγραφών, την αναζήτηση μέσα στις εγγραφές κ.ά.). Το ΣΔΒΔ δέχεται αιτήσεις από τους χρήστες των εφαρμογών και επικοινωνεί με τα αρχεία της βάσης δεδομένων για να τα διεκπεραιώσει.

Αυτή η κοινή διεπαφή (interface) των εφαρμογών με τα αρχεία αποκαλείται λογική διεπαφή. Οι εφαρμογές που δημιουργούμε δεν απασχολούνται με τον τρόπο που είναι αποθηκευμένα τα δεδομένα, πόσο χώρο καταλαμβάνουν και αυτή η ιδιότητα είναι γνωστή ως ανεξαρτησία δεδομένων. Αυτό σημαίνει πρακτικά ότι οποιαδήποτε αλλαγή στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων δεν θα συνεπάγεται και αλλαγή στις εφαρμογές: ένα πρόβλημα που ταλαιπωρούσε πολύ τους προγραμματιστές παλαιότερων εποχών. Ακόμη, η προσθήκη, η κατάργηση ή και η τροποποίηση κάποιων εφαρμογών δεν θα έχει καμία επίπτωση στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων. Στα ΣΔΒΔ έχει επικρατήσει η λεγόμενη αρχιτεκτονική των τριών επιπέδων (βαθμίδων), όπου τα τρία επίπεδα είναι τα εξής :

- **Εσωτερικό επίπεδο** (internal level), έχει να κάνει με την αποθήκευση των αρχείων στον σκληρό δίσκο, δηλ. την πραγματική ή φυσική κατάστασή τους.
- **Εξωτερικό επίπεδο** (external level), έχει να κάνει με τους χρήστες είτε αυτοί είναι απλοί χειριστές, είτε προγραμματιστές ή και οι διαχειριστές της βάσης δεδομένων
- **Εννοιολογικό επίπεδο** (conceptual level), είναι ένα ενδιάμεσο επίπεδο που διασυνδέει τα δύο άλλα επίπεδα και έχει να κάνει με τη λογική σχεδίαση των αρχείων της βάσης δεδομένων.

Οι Οντότητες (Entities)

Με τον όρο οντότητα (entity) εννοούμε ένα αντικείμενο, ένα πρόσωπο, μια κατάσταση και γενικά οτιδήποτε μπορεί να προσδιορισθεί σαν ανεξάρτητη ύπαρξη (αυτόνομη μονάδα του φυσικού κόσμου). Για παράδειγμα, σε μια βάση δεδομένων μιας εμπορικής εταιρείας, οντότητες μπορεί να είναι οι εργαζόμενοι, οι πελάτες, οι προμηθευτές, οι παραγγελίες, τα είδη της αποθήκης (προϊόντα) κ.ά.

Το Μοντέλο Οντοτήτων Συσχετίσεων (Entity Relationship Model, ER Model) είναι μια διαγραμματική αναπαράσταση της δομής μιας βάσης δεδομένων και χρησιμοποιείται κατά τη φάση του λογικού σχεδιασμού της βάσης. Δηλαδή, δεν ασχολείται με τον τρόπο που αποθηκεύονται τα δεδομένα της βάσης, αλλά με την ταυτοποίηση των δεδομένων και με τον τρόπο με τον οποίο αυτά συσχετίζονται μεταξύ τους. Για παράδειγμα μια εταιρεία η οποία περιέχει δεδομένα που αφορούν τους υπαλλήλους της (employees), τα τμήματά της (departments) και τα έργα (projects) που έχουν

αναλάβει αυτά τα τμήματα. Ένα τμήμα της εταιρείας μπορεί να εποπτεύει ένα ή περισσότερα έργα (projects) και ένας υπάλληλος ανήκει σ' ένα μόνο τμήμα της εταιρείας αλλά μπορεί να απασχολείται ταυτόχρονα σε πολλά έργα, τα οποία δεν είναι υποχρεωτικό να παρακολουθούνται από το ίδιο τμήμα.

Οι Ιδιότητες (Attributes)

Με τον όρο ιδιότητα ή χαρακτηριστικό ή και πεδίο (attribute) μιας οντότητας, αναφερόμαστε σ' ένα από τα συστατικά της στοιχεία που την περιγράφουν και την κάνουν να ξεχωρίζει από τα άλλα στοιχεία της ίδιας οντότητας. Για παράδειγμα, η οντότητα ΠΕΛΑΤΗΣ μπορεί να έχει ως ιδιότητες (χαρακτηριστικά) τον κωδικό, το επώνυμο, το όνομα, τη διεύθυνση, το τηλέφωνο, το ΑΦΜ κ.ά., με τη βοήθεια των οποίων μπορούμε να ξεχωρίσουμε τους πελάτες μεταξύ τους.

Επίσης, η οντότητα ΠΑΡΑΓΓΕΛΙΑ μπορεί να έχει ως ιδιότητες (χαρακτηριστικά) τον κωδικό, τον αριθμό παραστατικού, την ημερομηνία, τον κωδικό πελάτη, το προϊόν κ.ά., με τη βοήθεια των οποίων μπορούμε να ξεχωρίσουμε τις παραγγελίες μεταξύ τους. Στο παράδειγμα της εταιρείας, μπορούμε να ορίσουμε έναν τύπο οντότητας για τους υπαλλήλους της εταιρείας (EMPLOYEE), έναν τύπο οντότητας για τα τμήματα που έχει η εταιρεία (DEPARTMENT) και έναν τύπο οντότητας για τα έργα που έχει αναλάβει η εταιρεία (PROJECT). Καθένας από τους παραπάνω τύπους οντοτήτων περιγράφεται από ένα όνομα και από το σύνολο των πεδίων που περιέχει. Οι πληροφορίες αυτές αποτελούν το σχήμα (schema) της οντότητας.

Το Πρωτεύον Κλειδί (Primary Key)

Πρωτεύον κλειδί ή πεδίο κλειδί (primary key) μιας οντότητας καλείται εκείνη η ιδιότητα (ή ο συνδυασμός ιδιοτήτων) που έχει μοναδική τιμή για όλα τα στιγμιότυπα (εμφανίσεις) της οντότητας. Για παράδειγμα, στην οντότητα ΠΕΛΑΤΗΣ πρωτεύον κλειδί είναι ο κωδικός πελάτη, στην οντότητα ΠΑΡΑΓΓΕΛΙΑ πρωτεύον κλειδί μπορεί να είναι ο κωδικός παραγγελίας ή ο αριθμός παραστατικού κοκ. Υπάρχουν περιπτώσεις όπου το πεδίο κλειδί ενός τύπου οντότητας μπορεί να μην είναι απλό αλλά σύνθετο, να αποτελείται δηλαδή από πολλά απλά πεδία και τότε η συνθήκη της μοναδικότητας για την τιμή του κλειδιού δεν εφαρμόζεται σε κάθε πεδίο του σύνθετου κλειδιού αλλά στο σύνολο του συνδυασμού αυτών των πεδίων.

Οι Συσχετίσεις (Relationships)

Με τον όρο συσχέτιση (relationship) αναφερόμαστε στον τρόπο σύνδεσης (επικοινωνίας) δύο ξεχωριστών οντοτήτων, ώστε να μπορούμε να αντλούμε στοιχεία (πληροφορίες) από τον συνδυασμό τους. Για παράδειγμα, η οντότητα ΓΙΑΤΡΟΣ συσχετίζεται με την οντότητα ΑΣΘΕΝΗΣ αλλά και με την οντότητα ΚΛΙΝΙΚΗ στη βάση δεδομένων ενός νοσοκομείου. Μπορούμε να δεχθούμε ότι ένας γιατρός παρακολουθεί (συσχετίζεται με) πολλούς ασθενείς, αλλά ένας ασθενής παρακολουθείται από (συσχετίζεται με) έναν μόνο γιατρό και επίσης ένας γιατρός συσχετίζεται με (ανήκει σε) μία μόνο κλινική, αλλά μια κλινική συσχετίζεται με (απασχολεί) πολλούς γιατρούς.

Στο παράδειγμα της εταιρείας, η οντότητα EMPLOYEE συσχετίζεται με την οντότητα DEPARTMENT και η οντότητα DEPARTMENT συσχετίζεται με την οντότητα PROJECTS. Ένας υπάλληλος ανήκει σ' ένα μόνο τμήμα και ένα τμήμα μπορεί να έχει πολλούς υπαλλήλους. Επίσης, ένα τμήμα εποπτεύει πολλά έργα αλλά ένα έργο εποπτεύεται από ένα μόνο τμήμα.

2.4.6 Βασικά μοντέλα Βάσεων Δεδομένων

Υπάρχουν τρία βασικά μοντέλα που έχουν επικρατήσει στις βάσεις δεδομένων, το ιεραρχικό, το δικτυωτό και το σχεσιακό, και τα οποία αναπτύχθηκαν με βάση αντίστοιχες δομές.

Το Ιεραρχικό Μοντέλο Βάσεων Δεδομένων

Το ιεραρχικό μοντέλο (hierarchical) έχει μια ιεραρχική δομή που θυμίζει δένδρο. Οι οντότητες μοιάζουν με απολήξεις από κλαδιά δένδρων και τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες. Από μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο εκκινούν πολλά κλαδιά, καθένα από τα οποία καταλήγει σε μια οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο. Αλλά, σε κάθε οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο αντιστοιχεί μία και μόνο μία οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο. Το μοντέλο αυτό ήταν το πρώτο που εμφανίσθηκε αλλά σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.

Το Σχεσιακό Μοντέλο Βάσεων Δεδομένων

Το σχεσιακό (relational) μοντέλο έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα μ' αυτό αποκαλούνται σχεσιακές (relational databases). Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Υστερούν μόνο σε ταχύτητα υπολογισμών και σε χώρο αποθήκευσης, αλλά μόνο όταν έχουμε να κάνουμε με πολύ μεγάλες βάσεις δεδομένων.

Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων (relations) αντίστοιχα. Ένας πίνακας (table) αποτελείται από γραμμές (rows) και στήλες (columns), όπου τοποθετούμε τα στοιχεία σε οριζόντια και κάθετη μορφή. Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται χαρακτηριστικό (attribute) ή πεδίο (field), ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες (στήλες) που αφορούν ένα στοιχείο της οντότητας και αποκαλείται πλειάδα (tuple) ή εγγραφή (record). Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ, για παράδειγμα. Το σύνολο των αποδεκτών τιμών μιας οντότητας αποκαλείται πεδίο ορισμού (domain).

Τα Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ) ή RBMS (Relational DataBase Management Systems) αναπτύχθηκαν με βάση το σχεσιακό μοντέλο και έχουν επικρατήσει πλήρως στον χώρο. Κατά τον σχεδιασμό και τη δημιουργία μιας σχεσιακής βάσης δεδομένων, οι πίνακες αποτελούν το μοναδικό δομικό και απαραίτητο στοιχείο για μπορέσουν να αναπαρασταθούν οι πληροφορίες που περιέχονται στη βάση δεδομένων.

Για να μπορέσουμε να προσθέσουμε, διαγράψουμε ή τροποποιήσουμε τα στοιχεία που περιέχονται σε μια βάση δεδομένων, χρησιμοποιούμε ειδικές γλώσσες προγραμματισμού που αποκαλούνται γλώσσες ερωταπαντήσεων (query languages). Η γλώσσα που αποτελεί σήμερα ένα διεθνές πρότυπο για την επικοινωνία των χρηστών με τα Σχεσιακά ΣΔΒΔ είναι η SQL (Structured Query Language) ή Δομημένη Γλώσσα Ερωτημάτων. Μπορεί να λειτουργήσει αυτόνομα αλλά και σε συνεργασία μ' άλλες γλώσσες προγραμματισμού.

Μια άλλη, φιλική προς τον χρήστη γλώσσα προγραμματισμού για να μπορούμε να υποβάλουμε ερωτήματα σε σχεσιακές βάσεις δεδομένων και να λαμβάνουμε απαντήσεις είναι η QBE (Query By Example), η οποία χρησιμοποιεί φόρμες για τη γραφική απεικόνιση των ερωτημάτων μας.

Σήμερα, υπάρχουν εξελιγμένα εργαλεία διαχείρισης σε γραφικό και φιλικό προς τον χρήστη περιβάλλον για δημιουργία πινάκων, φορμών, ερωτημάτων και εκθέσεων (αναφορών). Τα Σχεσιακά ΣΔΒΔ τα διακρίνουμε στα μεγάλα, τα οποία αφορούν κυρίως μεγάλους οργανισμούς και επιχειρήσεις, έχουν τεράστιο όγκο δεδομένων και πολλούς χρήστες ταυτόχρονα, και τέτοια συστήματα είναι τα Oracle, Ingres, Informix, SQL Server κ.ά. και τα μικρά, τα οποία αφορούν κυρίως απλούς χρήστες, όπως είναι η Microsoft Access, η Paradox, η FoxPro κ.ά.

Το Μοντέλο Οντοτήτων– Συσχετίσεων

Το μοντέλο που έχει επικρατήσει σήμερα για να παραστήσει τις έννοιες ή τη δομή μιας βάσης δεδομένων είναι το Μοντέλο Οντοτήτων–Συσχετίσεων (ΟΣ). Οι βασικές (θεμελιώδεις) έννοιες του μοντέλου αυτού είναι οι οντότητες, οι ιδιότητες ή χαρακτηριστικά και οι συσχετίσεις. Για να αναπαραστήσουμε ένα Μοντέλο Οντοτήτων – Συσχετίσεων χρησιμοποιούμε ειδικά διαγράμματα, όπου τα ορθογώνια συμβολίζουν τις οντότητες, οι ρόμβοι τις συσχετίσεις και οι ελλείψεις τις ιδιότητες. Με ευθείες γραμμές συνδέουμε τις οντότητες που συσχετίζονται με κάποιο τρόπο μεταξύ τους. Όλα τα παραπάνω αποτελούν τη λογική δομή μιας βάσης δεδομένων, μια εργασία που είναι απαραίτητο να γίνει πριν από την καταχώριση και την επεξεργασία των στοιχείων (πληροφοριών) της βάσης δεδομένων. Το μοντέλο οντοτήτων – συσχετίσεων αποτελεί μια γενική περιγραφή των γενικών στοιχείων που απαρτίζουν μια βάση δεδομένων και απεικονίζει την αντίληψη που έχουμε για τα δεδομένα (εννοιολογικό), χωρίς να υπεισέρχεται σε λεπτομέρειες υλοποίησης.

Οι Οντότητες

Με τον όρο οντότητα (entity) αναφερόμαστε σε κάθε αντικείμενο, έννοια, πρόσωπο ή κατάσταση που έχει μια ανεξάρτητη ύπαρξη. Είναι κάτι που ξεχωρίζει και μπορούμε να

συγκεντρώσουμε πληροφορίες (στοιχεία) γι' αυτό. Η οντότητα είναι αντίστοιχη με την έννοια της εγγραφής που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια του αντικειμένου στις σύγχρονες αντικειμενοστραφείς γλώσσες προγραμματισμού. Μια βάση δεδομένων μπορεί να περιέχει πολλές διαφορετικές οντότητες, οι οποίες απεικονίζονται με ορθογώνια παραλληλόγραμμα και συσχετίζονται μεταξύ τους ανά δύο.

Οι Ιδιότητες (Χαρακτηριστικά) των Οντοτήτων

Με τον όρο ιδιότητες (properties) ή χαρακτηριστικά (attributes) αναφερόμαστε στα συστατικά (δομικά) στοιχεία που προσδιορίζουν (αποτελούν) μια οντότητα. Η ιδιότητα είναι αντίστοιχη με την έννοια του πεδίου που συναντάμε στα αρχεία και στους πίνακες αλλά και με την έννοια της μεταβλητής στις γλώσσες προγραμματισμού.

Για παράδειγμα, η οντότητα ΓΙΑΤΡΟΣ μπορεί να αποτελείται από τις ιδιότητες (χαρακτηριστικά) ΑριθμόςΜητρώου, Επώνυμο, Όνομα, Πατρώνυμο, Ειδικότητα, Βαθμός, ΈτοςΓέννησης, Διεύθυνση, ΑΦΜ, Τηλέφωνο, Κινητό κ.ά., ενώ η οντότητα ΑΘΛΗΤΗΣ μπορεί να αποτελείται από τις ιδιότητες (χαρακτηριστικά) ΚωδικόςΑθλητή, Επώνυμο, Όνομα, Πατρώνυμο, Αγώνισμα, Επίδοση, Σύλλογος, ΈτοςΓέννησης, Διεύθυνση, Τηλέφωνο, Κινητό κ.ά.

Απ' όλες τις ιδιότητες μιας οντότητας, υπάρχει μία μόνο ιδιότητα, και σπανιότερα ένας συνδυασμός δύο ή και περισσότερων ιδιοτήτων, η τιμή της οποίας είναι μοναδική και προσδιορίζει την κάθε εμφάνιση (στιγμιότυπο) της οντότητας και αποκαλείται πρωτεύον κλειδί (primary key). Το πρωτεύον κλειδί εμφανίζεται στα διαγράμματα με υπογράμμιση ή με έντονη γραφή ή έχει ως πρόθεμα τον χαρακτήρα #. Στο διάγραμμα οντοτήτων–συσχετίσεων οι ιδιότητες απεικονίζονται με σχήματα ελλειπτικής μορφής, τα οποία ενώνονται με ευθείες γραμμές με την οντότητα στην οποία ανήκουν.

Τα Κλειδιά

Όπως είδαμε και νωρίτερα, με τον όρο κλειδί (key) ή πιο σωστά πρωτεύον κλειδί (primary key) αναφερόμαστε σε μια ιδιότητα (πεδίο), ή σπανιότερα σ' ένα σύνολο ιδιοτήτων (πεδίων), η τιμή της οποίας είναι μοναδική σ' ολόκληρη την οντότητα (πίνακας). Στην πράξη, το πρωτεύον κλειδί έχει διαφορετική τιμή για κάθε εμφάνιση της οντότητας ή για κάθε γραμμή (εγγραφή) του πίνακα και ποτέ δεν μπορεί να έχει μηδενική (κενή) τιμή (null). Προσοχή, άλλο πράγμα είναι ο αριθμός 0 και άλλο πράγμα είναι η κενή τιμή (null), δηλ. η μη ύπαρξη τιμής.

Ο συνδυασμός δύο ή και περισσότερων ιδιοτήτων (πεδίων) για τη δημιουργία ενός πρωτεύοντος κλειδιού αποκαλείται σύνθετο κλειδί. Ένα παράδειγμα σύνθετου κλειδιού θα μπορούσε να είναι ο συνδυασμός των ιδιοτήτων Επώνυμο, Όνομα και Πατρώνυμο, εφόσον φυσικά είμαστε απολύτως βέβαιοι ότι δεν υπάρχουν δύο ή και περισσότερα άτομα με κοινές τιμές στις παραπάνω ιδιότητες.

Ξένο κλειδί αποκαλείται μια ιδιότητα (πεδίο) που είναι πρωτεύον κλειδί σε μια οντότητα (πίνακας) αλλά που υπάρχει και σε μια άλλη οντότητα (πίνακας) σαν απλή ιδιότητα. Τα ξένα κλειδιά είναι απαραίτητα για να μπορέσουμε να κάνουμε τις συσχετίσεις (συνδέσεις, επικοινωνίες) ανάμεσα στις οντότητες (πίνακες).

Για παράδειγμα, στην οντότητα ΣΥΛΛΟΓΟΣ, το πεδίο ΚωδικόςΣυλλόγου είναι πρωτεύον κλειδί, ενώ στην οντότητα ΑΘΛΗΤΗΣ, το πεδίο ΚωδικόςΣυλλόγου είναι ξένο κλειδί και είναι απαραίτητο για να μπορέσουμε να υλοποιήσουμε τη συσχέτιση ΑΝΗΚΕΙ, δηλ. να αντλήσουμε την πληροφορία ποιοι αθλητές ανήκουν σε ποιους συλλόγους. Προφανώς, στην οντότητα ΣΥΛΛΟΓΟΣ, το πεδίο Κωδικός Συλλόγου θα έχει μοναδικές τιμές, ενώ στην οντότητα ΑΘΛΗΤΗΣ, το πεδίο Κωδικός Συλλόγου θα έχει επαναλαμβανόμενες τιμές και αυτό γιατί πολλοί αθλητές θα ανήκουν στον ίδιο σύλλογο, αλλά ένας αθλητής ανήκει υποχρεωτικά σ' έναν και μόνο έναν σύλλογο.

Σ' ένα άλλο παράδειγμα, στην οντότητα ΓΙΑΤΡΟΣ, το πεδίο ΚωδικόςΓιατρού είναι πρωτεύον κλειδί, ενώ στην οντότητα ΑΣΘΕΝΗΣ, το πεδίο ΚωδικόςΓιατρού είναι ξένο κλειδί και είναι απαραίτητο για να μπορέσουμε να υλοποιήσουμε τη συσχέτιση ΠΑΡΑΚΟΛΟΥΘΕΙΤΑΙ, δηλ. να αντλήσουμε την πληροφορία ποιοι ασθενείς παρακολουθούνται από ποιους γιατρούς. Προφανώς, στην οντότητα ΓΙΑΤΡΟΣ, το πεδίο ΚωδικόςΓιατρού θα έχει μοναδικές τιμές, ενώ στην οντότητα ΑΣΘΕΝΗΣ, το πεδίο ΚωδικόςΓιατρού θα έχει επαναλαμβανόμενες τιμές και αυτό γιατί πολλοί ασθενείς θα παρακολουθούνται από τον ίδιο γιατρό, αλλά ένας ασθενής παρακολουθείται μόνο από έναν γιατρό.

Αυτό αποτελεί βέβαια μια παραδοχή που κάνουμε για να μπορέσουμε να υλοποιήσουμε μια συσχέτιση σαν την παραπάνω σε μια βάση δεδομένων ενός Νοσοκομείου, αλλά μπορεί να θεωρήσει κάποιος ότι ένας ασθενής μπορεί να παρακολουθείται από πολλούς γιατρούς ταυτόχρονα, οπότε θα πρέπει να μεταβάλλουμε και τον τρόπο συσχέτισης των παραπάνω οντοτήτων.

Οι Συσχετίσεις Μεταξύ Οντοτήτων

Ο σωστός σχεδιασμός και προσδιορισμός των οντοτήτων και των ιδιοτήτων τους αποτελούν τα θεμελιώδη βήματα για τη σωστή σχεδίαση και υλοποίηση μιας βάσης δεδομένων. Μια συσχέτιση συνδέει δύο ή και περισσότερες οντότητες μεταξύ τους και παριστάνεται στο διάγραμμα οντοτήτων-συσχετίσεων μ' έναν ρόμβο. Οι συσχέτισεις είναι απαραίτητες για να μπορέσουμε να αντλήσουμε πληροφορίες που αφορούν δύο ή και περισσότερες οντότητες, όπως για παράδειγμα ποιοι πελάτες έκαναν παραγγελίες κάποια συγκεκριμένη χρονική περίοδο (συσχέτιση ΠΑΡΑΓΓΕΛΝΕΙ) ή ποιοι αθλητές ανήκουν σε ποιους συλλόγους (συσχέτιση ΑΝΗΚΕΙ) ή ποιοι αθλητές έλαβαν μέρος σε αγωνίσματα μια συγκεκριμένη χρονιά (συσχέτιση ΣΥΜΜΕΤΕΧΕΙ) κ.ο.κ.

Όταν οι οντότητες που συμμετέχουν σε μια συσχέτιση είναι δύο, η συσχέτιση αποκαλείται διμελής ή δυαδική. Ο βαθμός μιας συσχέτισης είναι ίσος με το πλήθος των οντοτήτων που συμμετέχουν σ' αυτήν. Μια συσχέτιση μπορεί και η ίδια να έχει ιδιότητες που να περιγράφουν ορισμένα χαρακτηριστικά της, όπως για παράδειγμα η συσχέτιση ΠΑΡΑΓΓΕΛΙΑ ανάμεσα στις οντότητες ΠΕΛΑΤΗΣ και ΠΡΟΪΟΝ μπορεί να περιέχει τις ιδιότητες (πεδία) ΚωδικόςΠελάτη, ΚωδικόςΠροϊόντος, ΚωδικόςΠαραγγελίας, ΗμερομηνίαΠαραγγελίας, Ποσότητα κ.ά.

Στην περίπτωση αυτή το σωστό είναι να δημιουργήσουμε μια ακόμα οντότητα, την οντότητα ΠΑΡΑΓΓΕΛΙΑ, η οποία και θα περιέχει όλες τις παραπάνω ιδιότητες, και να μετονομάσουμε την προηγούμενη συσχέτιση από ΠΑΡΑΓΓΕΛΙΑ σε ΣΥΝΑΛΛΑΓΗ, που δεν θα περιέχει τώρα ιδιότητες. Έτσι, η παραπάνω συσχέτιση θα μετατραπεί από διμελή σε τριμελή. Όταν σχεδιάζουμε μια βάση δεδομένων, θα πρέπει να εκχωρούμε ιδιότητες μόνο στις οντότητες και να έχουμε τις συσχέτισεις απλά και μόνο για να κατανοούμε τις λογικές συνδέσεις ανάμεσα στις οντότητες.

Οι Διμελείς Συσχετίσεις

Ένα-προς-ένα (1:1), όπου μια εμφάνιση της μιας οντότητας συνδέεται με μία και μόνο μία εμφάνιση της άλλης οντότητας. Για παράδειγμα, η οντότητα ΣΥΛΛΟΓΟΣ έχει έναν μόνο προπονητή, ενώ η οντότητα ΠΡΟΠΟΝΗΤΗΣ συνδέεται μ' έναν και μόνο έναν σύλλογο. Σ' ένα άλλο παράδειγμα, η οντότητα ΝΟΜΟΣ έχει μία μόνο πόλη σαν πρωτεύουσα, ενώ η οντότητα ΠΡΩΤΕΥΟΥΣΑ αντιστοιχεί σ' έναν και μόνο έναν νομό. Στην περίπτωση των διμελών συσχετίσεων του τύπου ένα-προς-ένα, μπορούμε να ενώσουμε τα στοιχεία και των δύο ιδιοτήτων και να δημιουργήσουμε μια μοναδική οντότητα (πίνακα).

Ένα-προς-πολλά (1:M), όπου μια εμφάνιση της μιας οντότητας συνδέεται με πολλές εμφανίσεις της άλλης οντότητας αλλά κάθε εμφάνιση της δεύτερης οντότητας συνδέεται με μία και μόνο μία εμφάνιση της πρώτης οντότητας. Για παράδειγμα, ένας ΠΕΛΑΤΗΣ κάνει πολλές παραγγελίες, αλλά μια ΠΑΡΑΓΓΕΛΙΑ αντιστοιχεί σ' έναν και μόνο έναν πελάτη. Σ' ένα άλλο παράδειγμα, ένας ΣΥΛΛΟΓΟΣ έχει πολλούς αθλητές, αλλά ένας ΑΘΛΗΤΗΣ ανήκει σ' έναν και μόνο έναν σύλλογο. Οι διμελείς συσχετίσεις του τύπου ένα-προς-ένα είναι οι πιο συχνά συναντώμενες και οι πιο βολικές στη διαχείριση.

Πολλά-προς-πολλά (M:N), όπου σε μια εμφάνιση της μιας οντότητας αντιστοιχούν πολλές εμφανίσεις της άλλης οντότητας και σε κάθε εμφάνιση της δεύτερης οντότητας αντιστοιχούν πολλές εμφανίσεις της πρώτης οντότητας. Για παράδειγμα, ένας ΑΘΛΗΤΗΣ συμμετέχει σε πολλούς αγώνες αλλά και σ' έναν ΑΓΩΝΑ λαμβάνουν μέρος πολλοί αθλητές. Σ' ένα άλλο παράδειγμα, ένας ΚΑΘΗΓΗΤΗΣ διδάσκει σε πολλούς μαθητές αλλά και ένας ΜΑΘΗΤΗΣ διδάσκεται από πολλούς καθηγητές. Για να μπορέσουμε να διαχειριστούμε μια διμελή σχέση του τύπου πολλά-προς-πολλά, θα πρέπει να δημιουργήσουμε έναν τρίτο πίνακα που θα περιέχει δύο μόνο ιδιότητες (πεδία), δηλ. τα πεδία κλειδιά των δύο οντοτήτων, οπότε ο συνδυασμός τους θα είναι και το πεδίο κλειδί (σύνθετο κλειδί) του νέου πίνακα.

SQL

Η SQL (Structured Query Language) είναι μια γλώσσα προγραμματισμού, που έχει ως σκοπό την δημιουργία, την διαχείριση, την καταχώρηση δεδομένων και την άντληση δεδομένων από Βάσεις Δεδομένων και Σχεσιακές Βάσεις Δεδομένων. Η SQL διαφέρει από τις διαδικαστικές γλώσσες στο ότι επικεντρώνεται στον ορισμό και χειρισμό των δεδομένων και οι εντολές τις είναι πολύ κατανοητές μια και γράφονται σχεδόν σε απλά Αγγλικά. Σήμερα υπάρχουν διαφορετικές εκδόσεις της γλώσσας αυτής (διάλεκτοι) με μερικές διαφοροποιήσεις η καθεμία από την αρχική όπως η MySQL, Oracle, MS Sql Server. Η έκδοση που χρησιμοποιούν οι περισσότερο σήμερα, από μη ειδικευμένους σε SQL χρήστες, είναι η MS Access που εν αγνοία μας χρησιμοποιούμε την γλώσσα SQL σε ένα γραφικό περιβάλλον διεπαφής χρήστη.

3. Εργαλεία ανάπτυξης της εφαρμογής

Σε αυτό το κεφάλαιο περιγράφουμε όλα τα προγράμματα που χρησιμοποιήσαμε προκειμένου να αναπτύξουμε την εφαρμογή μας. Το IDE πάνω στο οποίο στηρίζαμε την ανάπτυξη της εφαρμογής είναι το Visual Studio με πρόσθετα τα Ms Charts και Crystal Reports. Η ανάπτυξη της βάσης δεδομένων έγινε με την Microsoft Access και η διασύνδεση της με την εφαρμογή, με τεχνολογία Ole db. Το αρχείο βοήθειας που εμπεριέχεται στην εφαρμογή δημιουργήθηκε με το εργαλείο Microsoft Help Workshop και τα γραφικά της εφαρμογής με το Adobe Photoshop. Το αρχείο της εγκατάστασης έγινε με το πρόγραμμα Inno Setup.



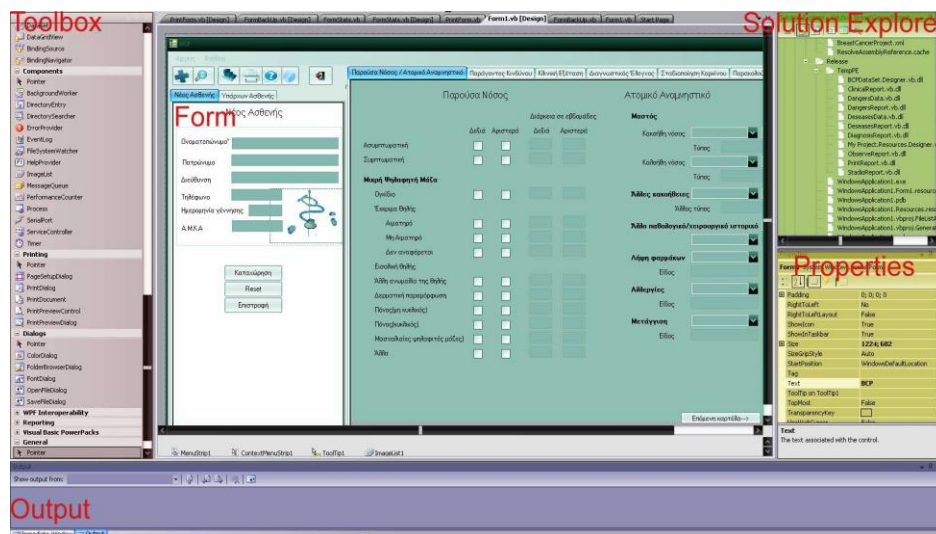
Εικόνα 4: Λογότυπο Microsoft Visual Studio

3.1 Visual Studio

Το Visual Studio, είναι ένα εύχρηστο περιβάλλον ανάπτυξης εφαρμογών, που αναπτύχθηκε από την εταιρεία Microsoft Corporation. Υποστηρίζει την ανάπτυξη προγραμμάτων σε κονσόλα, οπτικές εφαρμογές, ιστοσελίδες, υπηρεσίες WEB και άλλα. Το περιβάλλον ανάπτυξης Visual Studio βοηθά τον προγραμματιστή να αναπτύξει τα προγράμματά του με σχετική ευκολία, καθώς η τεχνολογία Microsoft IntelliSense οδηγεί τον προγραμματιστή να κατανοήσει με ευκολία πιθανά λάθη του κώδικά του, υπογραμμίζοντας τα με κόκκινη γραμμή δυναμικά κατά το χρόνο συγγραφής του προγράμματος. Αυτή η τεχνολογία είναι ικανή να εντοπίσει λάθη τα οποία μπορεί να είναι είτε συντακτικά, όπως για παράδειγμα η χρήση μιας εντολής με εσφαλμένο τρόπο, είτε λογικά, όπως για παράδειγμα η δήλωση ενός αντικειμένου χωρίς να αυτό να χρησιμοποιείται. Το Visual Studio υποστηρίζει την ανάπτυξη προγραμμάτων στις C++, C#, Visual Basic, F# και τη μεταφορά προγραμμάτων από την μία γλώσσα στην άλλη. Με άλλα λόγια συγγράφει ένα πρόγραμμα σε γλώσσα C++ μπορεί απλά και εύκολα να μετατραπεί αυτόματα σε κάποια εκ των γλωσσών που υποστηρίζει το Visual Studio.

3.1.1 Περιβάλλον

Καθώς εισερχόμαστε στο κυρίως περιβάλλον του Visual Studio παρατηρούμε πως είναι διαιρεμένο σε υπό-περιοχές. Κάθε μια απ' αυτές, έχει έναν συγκεκριμένο ρόλο στην υποβοήθηση του προγραμματιστή για τη σύνταξη ενός ολοκληρωμένου project.



Εικόνα 5: Περιβάλλον λειτουργίας Microsoft Visual Studio

Project Explorer

Βρίσκεται δεξιά της οθόνης και εξυπηρετεί στο να μπορέσουμε να πλοηγηθούμε σε όλα τα αρχεία του προγράμματός μας. Μας δείχνει από ποια συστατικά αποτελείται το project που φτιάχνουμε. Περιεχόμενα του εξερευνητή είναι οι φόρμες, τα modules, τα class modules κτλ.

Output

Στο κάτω μέρος του κώδικα εμφανίζεται ένα παράθυρο με όνομα “Output”, στο οποίο εμφανίζονται τυχόν πληροφορίες όπως είναι σφάλματα που έχουμε κάνει ή τυχόν προειδοποιήσεις σχετικά με τον κώδικά μας, ώστε να τον βελτιώσουμε, είτε πληροφορίες σχετικά με την διαδικασία build του προγράμματος.

ToolBox (Γραμμή εργαλείων)

Βρίσκεται στα αριστερά της κεντρικής οθόνης. Περιλαμβάνει τα «στοιχεία ελέγχου» πάνω στα οποία βασίζεται το «χτίσιμο» ενός project. Τα αντικείμενα αυτά καθοδηγούν τον χρήστη στη σύνταξη ενός λογισμικού. Σε αυτά οφείλεται ο όρος «αντικειμενοστραφής προγραμματισμός». Η γραμμή εργαλείων περιλαμβάνει τα βασικά στοιχεία ελέγχου. ωστόσο, ο χρήστης μπορεί εύκολα να προσθέσει επιπλέον όσα προαιρετικά εργαλεία.

Properties Window (Παράθυρο ιδιοτήτων)

Όλα τα στοιχεία ελέγχου έχουν ιδιότητες. Αυτές είναι παράμετροι που σχετίζονται με την εμφάνιση ενός αντικειμένου (χρώμα, γραμματοσειρά, τίτλος κτλ), με τη θέση του αντικειμένου σε μια φόρμα (απόσταση από το αριστερό και το πάνω άκρο της φόρμας), με το αν είναι προσβάσιμα από το χρήστη, αν είναι ορατά, με τον τύπο εμφάνισής τους και πολλά άλλα. Τα στοιχεία ελέγχου, εκτός από τις ιδιότητες, έχουν συμβάντα (events) και μεθόδους (methods).

Φόρμα (Form)

Είναι εκείνο το αντικείμενο πάνω στο οποίο «χτίζουμε» το project μας, το «χαρτί» πάνω στο οποίο θα σχεδιάσουμε τη μορφή του προγράμματός μας. Πρόκειται για την κεντρική μονάδα εργασίας την οποία διαμορφώνουμε οπτικά με τα αντικείμενα που προσθέτουμε επάνω της. Χωρίς τη φόρμα δεν εννοείται αντικειμενοστραφής προγραμματισμός. Πρόγραμμα (με τη στενή έννοια του όρου) χωρίς φόρμα δεν υφίσταται. Μόνο τα ειδικά (καθαρά υπολογιστικά) συστατικά (π.χ. DLL) δεν απαιτούν τη χρήση μιας φόρμας.

Η φόρμα αποτελεί το πιο σημαντικό στοιχείο ελέγχου. Πάνω της τοποθετούνται όλα τα εργαλεία τα οποία συνθέτουν το τελικό αποτέλεσμα (Project). Έχει πολλές ιδιότητες, συμβάντα και μεθόδους. Θα παρατηρήσατε ήδη ότι η φόρμα περιλαμβάνει εκ φύσεως ένα μικρό εικονίδιο (image), έναν τίτλο (Label), 3 CommandButtons καθώς και κάποια «αόρατα» στοιχεία χειρισμού.

3.1.2 Στοιχεία ελέγχου (Components) του toolbox

Στο Toolbox υπάρχει μια πληθώρα στοιχείων ελέγχου που κάθε ένα εξυπηρετεί διαφορετικές ενέργειες στην ανάπτυξη προγραμμάτων. Αυτά τοποθετούνται στη φόρμα με τη μέθοδο drag and drop. Παρακάτω περιγράψουμε συνοπτικά τα στοιχεία που χρησιμοποιήσαμε.

- **Label (Ετικέτα).** Πρόκειται για ένα απλό εργαλείο το οποίο χρησιμοποιούμε για να γράψουμε μια λεζάντα. Κατά την εκτέλεση του προγράμματος δεν μπορεί ο χρήστης να επέμβει στο κείμενό της.
- **Frame (πλαίσιο).** Πρόκειται για ένα εργαλείο ομαδοποίησης στοιχείων ελέγχου επάνω στη φόρμα. Είναι εξαιρετικά χρήσιμο γιατί παρέχει ευελιξία και διευκολύνει την οργάνωση των εργαλείων επάνω στη φόρμα. Οι ιδιότητές του, τα συμβάντα και οι μέθοδοι δεν διαφέρουν από αυτές της Label .
- **CheckBox (κουτί ελέγχου).** Ένα πολύ σημαντικό εργαλείο το οποίο χρησιμοποιείται για να δώσει στον χρήστη την ευχέρεια να εκτελέσει μια διαδικασία με πολλαπλές επιλογές. Στις ήδη γνωστές ιδιότητες που έχουμε δει παραπάνω, θα προσθέσουμε κάποιες ακόμα οι οποίες αποτελούν το κλειδί της διαχείρισης ενός CheckBox.
- **ComboBox (Πλαίσιο αναδιπλούμενης λίστας).** Το ComboBox είναι ένα πλαίσιο το οποίο χρησιμοποιούμε για οικονομία χώρου, όταν θέλουμε να διαθέσουμε ένα πλήθος επιλογών

στον χρήστη. Μας δίνει την δυνατότητα της απλής ή πολλαπλής επιλογής (ανάλογα με την τιμή μιας συγκεκριμένης ιδιότητας). Οι ιδιότητες που έχει είναι αρκετές. ωστόσο, θα εξετάσουμε μόνο ορισμένες από αυτές. Οι ιδιότητες που έχουν αναλυθεί σε προηγούμενα στοιχεία ελέγχου θα παραληφθούν για οικονομία χώρου.

- **PictureBox (Πλαίσιο Απεικόνισης).** Το PictureBox, σε αντίθεση με το Image, μπορεί να χρησιμοποιηθεί, πέρα από την απλή εμφάνιση μιας εικόνας και ως χώρος σχεδίασης. Μπορούμε να προβάλλουμε και να δημιουργήσουμε οτιδήποτε επιθυμούμε πάνω σε ένα PictureBox, από ελεύθερο ή γραμμικό σχέδιο, μέχρι και γραφικές παραστάσεις, να προσθέσουμε εργαλεία και να το χρησιμοποιήσουμε ως ομαδοποιητή κλπ. Είναι λοιπόν ένα πανίσχυρο εργαλείο, αλλά και δύσκολο για πλήρη αξιοποίηση.
- **TextBox (Πλαίσιο κειμένου).** Το πλαίσιο κειμένου είναι ένα πολύ χρήσιμο εργαλείο το οποίο μας επιτρέπει να εισάγουμε δεδομένα στο πρόγραμμα πριν και κατά τη φάση της εκτέλεσης.
- **CommandButton (Κουμπί ή πλήκτρο).** Το «κουμπί» είναι το πιο άμεσο εργαλείο που χρησιμοποιείται για να «πυροδοτήσει» την έναρξη κάποιας διαδικασίας. Χρησιμοποιείται σε κάθε εφαρμογή λόγω της απλότητας χειρισμού του.
- **ListBox (Πλαίσιο λίστας).** Το πλαίσιο λίστας μοιάζει πάρα πολύ με το πλαίσιο αναδιπλούμενης λίστας (ComboBox), τόσο στον τρόπο της λειτουργίας και της διαχείρισης, όσο και στην εμφάνιση.
- **MessageBox.** Στα Windows Forms, ο ευκολότερος τρόπος για να δείξουμε ένα παράθυρο διαλόγου ή παράθυρο προειδοποίησης είναι καλώντας το MessageBox.Show. Αυτό οδηγεί σε ένα προσαρμόσιμο παράθυρο διαλόγου που είναι πιο εύκολο από την χειροκίνητη εφαρμογή ενός διαλόγου
- **Trackbar.** Παρέχει ένα στοιχείο ελέγχου slider. Επιτρέπει στο χρήστη να επιλέξει μια τιμή από το slider σύροντας σε αυτό. Στο Visual Studio μπορούμε να αλλάξουμε πολλές πτυχές της trackbar, συμπεριλαμβανομένου του προσανατολισμού και της εμφάνισής της. Προσθέτουμε ένα πρόγραμμα χειρισμού συμβάντων για την εκτέλεση κώδικα όταν η trackbar χρησιμοποιείται. Πρόκειται για ένα ευέλικτο και παραμετροποιήσιμο στοιχείο ελέγχου.
- **Panel .** Δημιουργεί μια ομαδοποίηση των στοιχείων ελέγχου. Το control αυτό παρέχει ένα απλό πλαίσιο για την τοποθέτηση επιμέρους controls στο εσωτερικό του. Τα επιμέρους controls μπορεί να περιλαμβάνουν κουμπιά και πλαίσια κειμένου. Επιπλέον παρέχει επιλογές στο χρήστη να αλλάξει τα όρια και τους τρόπους προβολής του. Το panel control είναι ένας μηχανισμός ομαδοποίησης στις Windows Forms, και μπορεί να χρησιμοποιηθεί τόσο για την οπτική όσο και για τη λογική οργάνωση. Με τις επιλογές ορισμών των ορίων του, μπορεί να οργανωθεί οπτικά το πρόγραμμα σε επιμέρους controls. Με την ικανότητά του να επηρεάζει την κατάσταση σε όλα τα controls που εμπεριέχει, μπορεί να μειωθεί η λογική πολυπλοκότητα του προγράμματός.
- **GroupBox.** Σε ένα GroupBox μπορούν να τοποθετηθούν άλλα στοιχεία ελέγχου. Κατά το σχεδιασμό των διεπαφών χρήστη σε Windows Forms, αυτό το στοιχείο ελέγχου χρησιμοποιείται για τη δημιουργία ενός τετράγωνου σχήματος όπου μπορούν να τοποθετηθούν άλλα στοιχεία ελέγχου. Αυτό δημιουργεί ένα σημαντικό επίπεδο οπτικής συνέχειας. Το GroupBox βοηθάει στην οργάνωση ενός καθαρού και εύχρηστου interface για το πρόγραμμα και είναι χρήσιμο για τη δημιουργία παραθύρων προτιμήσεων και διαλόγων.
- **DateTimePicker.** Παρέχει μια διεπαφή για επιλογή ημερομηνίας. Επιτρέπει στο χρήστη να επιλέξει μια έγκυρη ημερομηνία και ώρα μέσω μιας διεπαφής ημερολογίου χωρίς σημαντική προσπάθεια.
- **Image (Εικόνα).** Το εργαλείο αυτό μας παρέχει τη δυνατότητα να προσθέτουμε εικόνες σε μια φόρμα.
- **TabControl.** Παρέχει μια διάταξη καρτελών στο περιβάλλον εργασίας του χρήστη. Το .NET Framework παρέχει αυτό το ευέλικτο και εύκολο στη χρήση control διάταξης . Με την προσθήκη αυτού του control γίνεται εφικτό να κρατήσει ένα σωρό από στοιχεία ελέγχου στο τελικό παράθυρο στο ελάχιστο, ενώ δεν περιορίζει το εύρος των επιλογών που προσφέρει αλλάζοντας τις σελίδες του, χειραγωγώντας το με κώδικα, και αλλάζοντας τις οπτικές ρυθμίσεις του. Κάθε tab του του tabcontrol λέγεται TabPage και είναι σημαντικό, καθώς ο

μηχανισμός αυτός επιτρέπει να προστεθεί ένας μεταβλητός αριθμός σελίδων για τον έλεγχο της καρτέλας.

- **Menu.** Μενού επιλογών. Το εικονίδιο του, όπως φαίνεται στην μπάρα εργαλείων της .NET, κάτω ακριβώς από το κεντρικό menu επιλογών. Αριστερά βλέπουμε το παράθυρο δημιουργίας του Menu επάνω στην τρέχουσα φόρμα. Ο Editor του Menu ανοίγει όταν εμείς πατήσουμε το παραπάνω εικονίδιο.
- **DataGridView.** Παρέχει μια οπτική διεπαφή με τα δεδομένα. Είναι ένας εξαιρετικός τρόπος για την εμφάνιση και να επεξεργαστεί των δεδομένων. Είναι προσβάσιμο με κωδικό VB.NET. Τα δεδομένα που έχουν επεξεργαστεί στο DataGridView μπορούν έπειτα να καταχωρηθούν κατευθείαν στη βάση δεδομένων. Με τα DataGridViews, μπορούμε να εμφανίσουμε δεδομένα σε μορφή πίνακα με πολλούς τρόπους. Για το σκοπό της χρήσης τους είναι πολύ πιο γρήγορα και αποτελεσματικότερα κατά το χρόνο εκτέλεσης τους από ό, τι οι περισσότερες άλλες λύσεις στα Windows Forms.

3.1.3 Επιμέρους στοιχεία ελέγχου

Πέρα από τα στοιχεία ελέγχου που συναντήσαμε στην μπάρα ToolBox υπάρχουν εκατοντάδες ακόμα που δημιουργήθηκαν είτε από την Microsoft είτε από ανεξάρτητους (τρίτους) κατασκευαστές, από τα οποία άλλα πωλούνται και άλλα διατίθενται δωρεάν. Τέτοια στοιχεία ελέγχου είναι τα ms chart και crystal viewer.

3.1.3.1 Microsoft Charts

Το chart control επιτρέπει τη δημιουργία διαγραμμάτων για Windows Forms εφαρμογές παρέχοντας απλά, διαισθητικά και οπτικά συναρπαστικά διαγράμματα για περίπλοκες στατιστικές και οικονομικές αναλύσεις. Τα chart controls έχουν τα ακόλουθα βασικά χαρακτηριστικά:

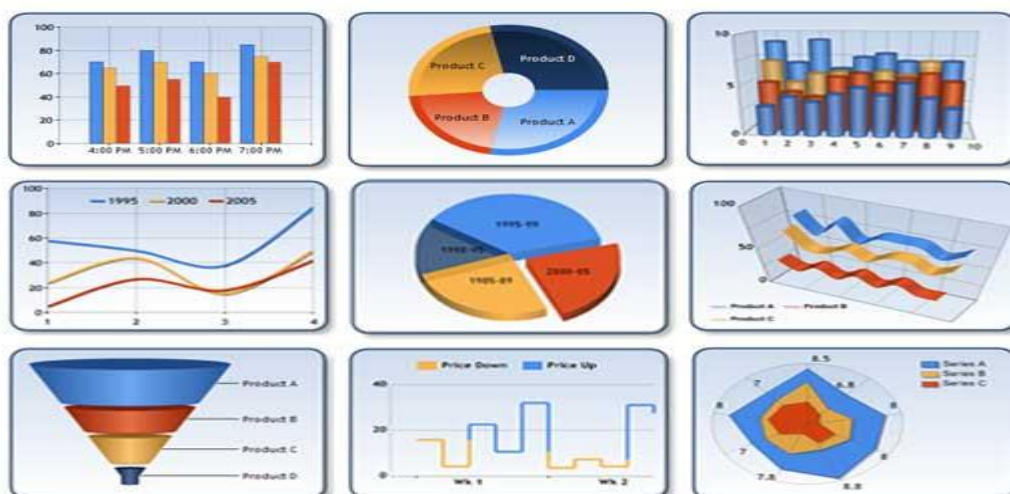
Κατηγορία	Χαρακτηριστικά
Περιβάλλον ανάπτυξης	<ul style="list-style-type: none"> • Υποστήριξη σχεδιασμού σε Visual Studio.
Επεκτασιμότητα	<ul style="list-style-type: none"> • Υποστήριξη για απεριόριστο αριθμό περιοχών γραφήματος, τίτλων και σχολιασμών. • Υποστήριξη για απεριόριστο αριθμό σειρών δεδομένων και σημείων.
Τύποι γραφημάτων	<ul style="list-style-type: none"> • 35 διαφορετικοί τύποι γραφήματος. • Data binding. • αντιγραφή, συγχώνευση, διάσπαση, διαλογή, αναζήτηση, ομαδοποίηση και φιλτράρισμα δεδομένων.
Δεδομένα	<ul style="list-style-type: none"> • εξαγωγή δεδομένων. • Η δυαδική και XML serialization. • Χειρισμός καθαρισμού σημείο δεδομένων. • Υποστήριξη για ημερομηνίες, ώρες, νόμισμα, και πολλά άλλα. • Περισσότερα από 50 τύπων οικονομικών και στατιστικών για την ανάλυση και μεταμόρφωση δεδομένων.
Εμφάνιση	<ul style="list-style-type: none"> • 3D υποστήριξη για τους περισσότερους τύπους γραφημάτων. • Διαχείριση για αυτόματη και χειροκίνητη διάταξη και ευθυγράμμιση. • 3D προσαρμογή. • Αυτόματη και χειροκίνητη αυξομείωση ορίων. • λογαριθμική κλίμακα για κάθε βάση. • Έξυπνη τοποθέτηση ετικετών δεδομένων.
Προσαρμογή	<ul style="list-style-type: none"> • Χειραγώγισμο γράφημα σε πραγματικό χρόνο. • Διαγράμματα Drill-down και tooltips.
Ειδίκευση σε Windows Forms	<ul style="list-style-type: none"> • Ζουμ και κύλιση. • Εκτύπωση Chart.

Πίνακας 3: Βασικά χαρακτηριστικά Ms charts

Μια εικόνα chart αποτελείται από διάφορα στοιχεία, όπως άξονες, σειρές και άλλα στοιχεία γραφήματος. Κάθε στοιχείο γραφήματος αντιστοιχεί σε ένα αντικείμενο.

- **Chart Picture** (εικόνα γραφήματος). Η εικόνα γραφήματος είναι ολόκληρη η εικόνα που παράγεται, και αντιστοιχεί στις ρίζες της κλάσης chart.
- **Title**. Ένας τίτλος για την εικόνα του γραφήματος. Μπορεί να προστεθεί οποιοσδήποτε αριθμός τίτλων σε μια εικόνα γραφήματος.
- **Series**. Μία σχετική ομάδα σημείων δεδομένων. Κάθε σειρά έχει ένα σχετικό τύπο γραφήματος. Ο αριθμός των σειρών και ο τρόπος που ένα διάγραμμα μπορεί να εμφανίσει εξαρτάται από τον τύπο γραφήματος που έχετε ορίσει και αντιστοιχεί στην κλάση Series.
- **Chart Area**. Μια ορθογώνια περιοχή που χρησιμοποιείται για τον σχεδιασμό των series, ετικετών, αξόνων, των γραμμών πλέγματος, tick marks, και ούτω καθεξής. Ανάλογα με τον τύπο γραφήματος, πολλαπλές σειρές μπορούν να παρουσιαστούν σε μία περιοχή γραφήματος. Το στοιχείο αυτό αντιστοιχεί στην κλάση ChartArea.
- **Plot Area**. Η ορθογώνια περιοχή σε μια περιοχή γραφήματος που χρησιμοποιείται για την οργάνωση chart series και γραμμών πλέγματος. Οι ετικέτες, τα σημάδια υποδιαίρεσης και οι τίτλοι αξόνων σχεδιάζονται έξω από την plot area, αλλά μέσα στην περιοχή γραφήματος.
- **Axis Label**. Μια ετικέτα επί ενός άξονα. Δημιουργείται αυτόματα, εάν δεν παρέχονται προσαρμοσμένες ετικέτες. Το στοιχείο αυτό αντιστοιχεί στην κλάση Label.
- **Axis Title**. Ο τίτλος ενός άξονα που περιγράφει το τι αντιπροσωπεύει ο άξονας.
- **Grid Lines**. Οι οριζόντιες και κατακόρυφες γραμμές πλέγματος, οι οποίες συνήθως συμβαίνουν σε συνδυασμό με τα tick marks. Το στοιχείο αυτό αντιστοιχεί στην κλάση Grid.
- **Tick Marks**. Τα σημάδια υποδιαίρεσης στους άξονες, που συνήθως δημιουργούνται σε συνδυασμό με τις γραμμές πλέγματος. Το στοιχείο αυτό αντιστοιχεί στην κλάση TickMark.
- **Label**. Μια ετικέτα που περιγράφει ένα σημείο δεδομένων.

Στα ms charts είναι δυνατή η προσαρμογή της εμφάνισης όλων των στοιχείων του γραφήματος με τη χρήση των ιδιοτήτων εμφάνισης τους. Κάθε στοιχείο γραφήματος περιέχει ένα πλούσιο σύνολο ιδιοτήτων εμφάνισης, το οποίο επιτρέπει τον έλεγχο της εμφάνισης του γραφήματος για κάποιες ή όλες τις ιδιότητες όπως χρώμα, στυλ, πλάτος, σκιά και ούτω καθεξής. Για στοιχεία με επιφάνεια, συμπεριλαμβανομένων των δεικτών και των σχολίων, είναι δυνατός ο έλεγχος των ιδιοτήτων για background όπως φόντο, χρώμα, δευτερεύον χρώμα, διαφάνεια, κλίση, εικόνα, κατάσταση αναδίπλωσης της εικόνας και άλλα πολλά. Για στοιχεία κειμένου, μπορεί να ελέγξει το μέγεθος, τη γραμματοσειρά, το χρώμα, τον προσανατολισμό, το στυλ, τη σκιά, και ούτω καθεξής.



Εικόνα 6: Διάφορες προβολές των Ms Charts

Μπορεί να δεσμεύσει δεδομένα ώστε να τα προβάλει σε διαγράμματα από διάφορες πηγές δεδομένων όπως data viewers, data readers (SQL, OleDb), dataSets, πίνακες, λίστες και απ' όλα τα IEnumerable αντικείμενα.

Το chart control έχει τη δυνατότητα να χρησιμοποιεί anti-aliasing ώστε να μπορεί να εξομαλύνει τα αιχμηρά χρώματα. Αυτό κάνει την εικόνα του γραφήματος πολύ πιο ευχάριστη για τα μάτια του θεατή. Χρησιμοποιεί ένα σύστημα συντεταγμένων για να τοποθετήσει τίτλους, σχόλια και τις περιοχές διαγραμμάτων στην εικόνα. Για να γίνει αυτό, χρησιμοποιεί την ιδιότητα θέσης κάθε στοιχείου. Επίσης μπορεί να δώσει στο θεατή του γραφήματος την ευελιξία να μεγεθύνει και να κατευθύνει την προβολή δεδομένων, ενεργοποιώντας κέρσορες και επιλεγμένες περιοχές με δυνατότητα ζουμ και κύλισης για τους άξονες X ή Y . Επιπλέον υποστηρίζει εκτύπωση του γραφήματος επιλέγοντας εκτύπωση από τις ιδιότητες του. Αυτό το αντικείμενο εκτυπώνει όλα τα στοιχεία του chart control κατά την τρέχουσα προβολή δεδομένων, εκτός από τις γραμμές κύλισης.

3.1.3.2 Crystal Reports

Τα Crystal Reports υπήρξαν στο πακέτο Visual Studio από το 1993 και είναι πλέον το καθιερωμένο πρότυπο για την υποβολή εκθέσεων στο Visual Studio.NET. Έχοντας εκδώσει περισσότερες από 4 εκατομμύρια άδειες τα Crystal Reports προηγούνται στις προτιμήσεις μεταξύ των συγγραφέων εκθέσεων για Windows και ενδυναμώνουν το μέλλον των εκθέσεων για Web. Με τα Crystal Reports γίνεται εφικτή η δημιουργία διαδραστικών περιεχόμενων σε ποιότητα παρουσίασης για πλατφόρμα .NET. Μπορούν να φιλοξενήσουν εκθέσεις σε Web και Windows πλατφόρμες και να δημοσιεύσουν Crystal Reports σαν Report Web Services σε ένα Web διακομιστή. Για παράδειγμα, μπορεί να δημιουργηθεί μια εφαρμογή Web που να επιτρέπει στους χρήστες να κάνουν επιλογές σε ένα γράφημα και να φιλτράρουν τις πληροφορίες του σύμφωνα με τις ανάγκες τους καθώς το διάγραμμα είναι στην πραγματικότητα ένα Crystal Report που αλληλεπιδρά με άλλα controls της εφαρμογής. Μέσω κώδικα μπορεί να γίνει ανάγνωση εγγραφών μιας βάσης δεδομένων και εκτύπωση τους μέσα στην Windows ή Web εφαρμογή.

Αντί για χρήση κώδικα, η δημιουργία και μορφοποίηση του report που χρειαζόμαστε μπορεί να γίνει μέσω του περιβάλλοντος εργασίας Designer Crystal Report. Η ισχυρή μηχανή των Crystal Reports επεξεργάζεται τη μορφοποίηση, την ομαδοποίηση, και την εκτίμηση των κριτηρίων που έχουμε ορίσει. Με τη χρήση των Crystal Report Experts, μπορούμε να δημιουργήσουμε γρήγορα reports με βάση τις αναπτυξιακές ανάγκες μας :

- Επιλογή από την διάταξη των reports που κυμαίνονται από τυποποιημένες αναφορές για να σχηματίσουν γράμματα, ή και δημιουργία δικής μας έκθεση από το μηδέν.
- Εμφάνιση διαγραμμάτων που οι χρήστες μπορούν να επιλέξουν ώστε να δούνε αναλυτικά τα στοιχεία της αναφοράς.
- Υπολογισμός περιλήψεων, υποσύνολων και ποσοστών επί των ομαδοποιημένων δεδομένων..
- Υπό όρους μορφοποίηση κειμένου και περιστροφή των αντικειμένων ενός κειμένου.

Όταν δημιουργούμε μια αναφορά, μπορούμε να καθορίσουμε την πηγή των δεδομένων της, το σχεδιασμό της διάταξης της αναφοράς, και να αποφασίσουμε τον τρόπο που οι χρήστες θα έχουν πρόσβαση στα δεδομένα της έκθεσης. Τα Crystal Reports συνδέονται με βάσεις δεδομένων μέσω drivers βάσεων δεδομένων. Κάθε driver είναι γραμμένος για να χειρίζεται ένα συγκεκριμένο τύπο βάσης δεδομένων ή τεχνολογίας πρόσβασης σε βάση δεδομένων. Για να δουλύνουν τα Crystal Reports απαιτούν framework για Crystal Reports 8.0 και παρέχουν στους χρήστες του Visual Studio .NET απεικόνιση δεδομένων και δυνατότητες ανάλυσης με απρόσκοπτη ενσωμάτωση με όλες τις γλώσσες που υποστηρίζονται στο Visual Studio. Διαθέτουν μια ανοικτή και ευέλικτη αρχιτεκτονική με πρότυπα όπως η XML για την από κοινού χρήση πληροφοριών και reports στο Web. Η παρουσίαση αναφορών σε windows εφαρμογές αλλά και σε HTML 3.2 και 4.0, είναι αρκετά διαδραστική παρέχοντας χαρακτηριστικά όπως πλοήγηση για αναφορές και αναζήτηση κειμένου.

Pull and Push Models

Προκειμένου να παρέχει την πιο ευέλικτη πρόσβαση σε δεδομένα για προγραμματιστές, οι drivers βάσης δεδομένων των Crystal Reports έχουν σχεδιαστεί να παρέχουν ένα μοντέλο ανάκτησης (pull) και προώθησης (push) για προσβάσεις σε δεδομένα.



Εικόνα 7: Μοντέλο Pull

Σε ένα μοντέλο pull, ο driver θα συνδεθεί στην βάση και θα τραβήξει δεδομένα καθώς του ζητείται. Με αυτό το μοντέλο, η σύνδεση στη βάση και οι εντολές SQL που εκτελούνται για την ανάκτηση δεδομένων, διαχειρίζονται από τα Crystal Reports καθαυτά και δεν απαιτούν χρήση κώδικα από τον προγραμματιστή.



Εικόνα 8: Μοντέλο Push

Σε αντίθεση με το μοντέλο pull, το μοντέλο push απαιτεί από τον προγραμματιστή να γράψει κώδικα προκειμένου να συνδεθεί στη βάση εκτελώντας μια εντολή SQL για την δημιουργία ενός recordset η dataset που αντιστοιχεί στα πεδία της αναφοράς και προωθώντας αυτό το αντικείμενο στις αναφορές. Με αυτή τη μέθοδο γίνεται εφικτή η από κοινού χρήση σύνδεσης μιας εφαρμογής και το φιλτράρισμα δεδομένων πριν σταλούν στα Crystal Reports.

Designer

Η δημιουργία των Crystal Reports γίνεται με την βοήθεια του Designer. Ο Crystal Report Designer ανοίγει αυτομάτως όταν το αντικείμενο Crystal Report προστίθεται στο project, ή όταν γίνεται διπλό κλικ σε ένα υπάρχων αντικείμενο Crystal Report του project. Στον Designer, η πρώτη επιλογή μας είναι αυτή της πηγής των δεδομένων (data source) στα οποία θα ανατρέχει το report. Οι πηγές μπορεί να είναι περισσότερες από δύο. Μετά επιλέγουμε τους πίνακες δεδομένων που θα χρησιμοποιηθούν στην αναφορά. Τα Crystal Reports μπορούν αυτομάτως να συνδέσουν τους πίνακες. Οι πίνακες της βάσης συνδέονται με τέτοιο τρόπο ώστε οι εγγραφές να φιλτράρονται με βάση τις σχέσεις των πινάκων της βάσης. Ο Crystal Report Designer χρησιμοποιείται με την λογική σύρε-ρίξε (drag-and-drop) όμοια με αυτή του Visual Studio .NET — μπορούμε δηλαδή να σύρουμε ένα πεδίο της βάσης, από έναν πίνακα της αναφοράς να το ρίξουμε στον Designer και μέσω του παραθύρου ιδιοτήτων του να το μορφοποιήσουμε. Τα αντικείμενα που μπορεί να μας παρέχει μία Αναφορά είναι:

- Πεδίο Βάσης (Database fields).
- Πεδίο Φόρμουλας (Formula fields).
- Πεδίο Παραμέτρου (Parameter fields).
- Πεδίο Αθροίσματος (Summary fields)
- Πίνακες (Charts).
- Υποαναφορά (Subreports).

Report Sections

Ο Crystal Report Designer είναι χωρισμένος σε τομείς αναφοράς (report sections), όπως επικεφαλίδα, υποσέλιδο και λεπτομέρειες. Τα αντικείμενα τα ρίχνουμε στον τομέα της επιλογής μας. Έτσι τα δεδομένα θα εμφανίζονται σύμφωνα με τις επιλογές μας. Συγκεκριμένα η εμφάνιση των δεδομένων της αναφοράς διαφοροποιούνται ανάλογα με το Section στο οποίο τοποθετούνται. Για παράδειγμα η είσοδος ενός πίνακα στο Report Header Section έχει ως αποτέλεσμα την εμφάνισή του

ReportDocument Properties

Το ReportDocument περιέχει πολλές ιδιότητες που μας επιτρέπει να καθορίσουμε την μορφή και την συμπεριφορά της Αναφοράς. Οι κυριότερες από αυτές είναι οι παρακάτω.

- **Database.** Το αντικείμενο Database παρέχει πρόσβαση στις πληροφορίες που περιέχονται στην βάση δεδομένων που περιλαμβάνονται στην Αναφορά. Το αντικείμενο περιέχει μία συλλογή πινάκων. Αυτή η συλλογή περιέχει το άθροισμα των πινάκων της βάσης που χρησιμοποιούνται στην αναφορά και τις πληροφορίες σύνδεσης μέσω του αντικειμένου ConnectionInfo. Επίσης παρέχει πληροφορίες για όλα τα πεδία των πινάκων μέσω του DatabaseFieldDefinitions.
- **DataDefinition.** Το DataDefinition παρέχει πρόσβαση στα πεδία πινάκων, πεδία παραμέτρων, πεδία ταξινόμησης, πεδία ομαδοποίησης, πεδία αθροισμάτων και πεδία SQL εντολών.
- **ExportOptions .** Το ExportOptions παρέχει πρόσβαση στις επιλογές εξαγωγής της Αναφοράς που αφορούν τον τύπο προορισμού και την μορφή του εξαχθέντος αρχείου (Excel, PDF, Word and HTML).
- **PrintOptions.** Το PrintOptions παρέχει τις επιλογές για την μορφοποίηση της εκτύπωσης μιας αναφοράς όπως το όνομα του εκτυπωτή, το μέγεθος του χαρτιού κτλ.
- **ReportDefinition.** Το ReportDefinition επιτρέπει την λήψη των Areas, ReportObjects και Sections σε μια αναφορά. Μπορούμε δηλαδή να πάρουμε και να προσδιορίσουμε τις επιλογές μορφοποίησης.
- **ReportOptions.** Το ReportOptions μας επιτρέπει να πάρουμε και να προσδιορίσουμε τα δεδομένα που σχετίζονται με τις επιλογές της Αναφοράς.
- **SummaryInfo.** Το SummaryInfo μας επιτρέπει να πάρουμε και να προσδιορίσουμε γενικές πληροφορίες που περιέχονται στην αναφορά. Αυτά περιλαμβάνουν Λέξεις κλειδιά (KeywordsInReport), συγγραφέας Αναφοράς (ReportAuthor), σχόλια Αναφοράς (ReportComments), θέμα Αναφοράς (ReportSubject), και τίτλο Αναφοράς (ReportTitle).

3.1.4 Αντικειμενοστραφής Προγραμματισμός

Ο αντικειμενοστραφής προγραμματισμός (OOP) είναι μια μεθοδολογία σχεδίασης και κατασκευής λογισμικού. Είναι η αμέσως επόμενη λογική αναβάθμιση μετά το δομημένο προγραμματισμό για τη βελτιστοποίηση της επαναχρησιμοποίησης και συντήρησης κώδικα. Με άλλα λόγια ο αντικειμενοστραφής προγραμματισμός είναι μια μεθοδολογία ανάπτυξης στοιχείων λογισμικού (κλάσεων) με συσχετιζόμενες συμπεριφορές (μεθόδους) και περιορισμούς δεδομένων (ιδιότητες) που βοηθάει στην συναρμολόγηση όλων των στοιχείων για τη δημιουργία μιας ολοκληρωμένης εφαρμογής. Επιτρέπει την ομαδοποίηση των δεδομένων με μεταβλητές που εμπεριέχονται μέσα σε κλάση. Αυτά τα δεδομένα είναι ξεχωριστά και διακριτά από οποιαδήποτε άλλη κλάση στο πρόγραμμα και καμία κλάση δεν μπορεί να παρέμβει σε αυτά χωρίς να περάσει από τη συγκεκριμένη διεπαφή της κλάσης.

Κάθε αντικείμενο καθορίζει την λειτουργικότητα του σαν ένα σετ ιδιοτήτων και μεθόδων στις οποίες θα ανταποκρίνεται. Διαφορετικός κώδικας μπορεί να καλεί τις μεθόδους στο αντικείμενο κάνοντας τις να εκτελούν κάποια συμπεριφορά και επίσης να χρησιμοποιεί τις ιδιότητες για την ανάκτηση ή αλλαγή πληροφοριών. Με αυτόν τον τρόπο ο κώδικας δεν μπορεί να επηρεάσει άμεσα τις πληροφορίες ή διεργασίες άλλων αντικειμένων.

Κλάσεις

Στην .NET όλα τα controls που τοποθετούμε στη φόρμα μας αποτελούν κλάσεις. Πατώντας διπλό κλικ σε κάποιο control (π.χ. κουμπί) ανοίγει ο editor δημιουργώντας μια κλάση προκειμένου να εισάγουμε κώδικα. Οι κλάσεις αυτές μπορούμε να καθορίσουμε εάν είναι public ή private ρυθμίζοντας την προσβασιμότητα σε αυτές από τα διάφορα μέρη του προγράμματος.

Μία κλάση έχει γενικά την δομή Public Class Name και αποτελείται από ένα σώμα αρχικοποίησης (initialization block), τις μεταβλητές, μία ή περισσότερες μεθόδους και τη μέθοδο constructor. Το σώμα αρχικοποίησης εκτελείται μία και μοναδική φορά πριν εκτελεστεί οτιδήποτε άλλο και αφορά κάποιες αρχικές συνθήκες. Αυτό συμβαίνει είτε όταν η κλάση φορτώνεται για πρώτη φορά (στατικό), είτε κάθε φορά που δημιουργείται ένα αντικείμενο της Κλάσης αυτής (μη-στατικό).

Οι μεταβλητές που δηλώνουμε μπορεί να είναι είτε μεταβλητές αντικείμενων (instance variables), μεταβλητές δηλαδή που αφορούν κάθε αντικείμενο ξεχωριστά και οι τιμές τους είναι μοναδικές σε σχέση με αυτό, είτε μεταβλητές κλάσης ή στατικές (class ή static variables) οι οποίες λαμβάνουν τιμές κοινές για όλα τα αντικείμενα και μπορούν να χρησιμοποιηθούν ακόμη και εάν δεν δημιουργηθεί κανένα αντικείμενο της κλάσης αυτής. Η μέθοδος constructor αφορά στη μέθοδο που θα εκτελεστεί όταν δημιουργείται ένα αντικείμενο της κλάσης αυτής και έχει πάντα την ίδια ονομασία με το όνομα της κλάσης. Μπορεί να υπάρχουν περισσότερες από μία τέτοιες μέθοδοι, κάθε μία από τις οποίες λαμβάνει διαφορετικά ορίσματα εισόδου. Οι υπόλοιποι μέθοδοι που μπορεί να περιέχονται σε μία κλάση αφορούν είτε μεθόδους αντικείμενων (instance methods) οι οποίοι αναφέρονται σε αντικείμενα της κλάσης, είτε μεθόδους κλάσης ή στατικές μεθόδους (class ή static methods) και αφορούν μεθόδους οι οποίοι μπορούν να εκτελεστούν ακόμη και εάν δεν έχουν δημιουργηθεί αντικείμενα της κλάσης αυτής.

Οι δηλώσεις που αποτελούν και ταυτόχρονα προσδιορίζουν μία κλάση, γενικά μπορεί να είναι δύο ειδών όπως σώμα δεδομένων (data members) και μέθοδοι (methods). Σε κάθε κλάση μπορούμε να προσδιορίσουμε δύο είδη μεταβλητών τις μεταβλητές αντικειμένων (instance variables) και τις μεταβλητές κλάσης (class ή static variables). Με μεταβλητές αντικειμένων κάθε αντικείμενο που δημιουργείται από μία κλάση, κρατάει και ένα διαφορετικό αντίγραφο των μεταβλητών αυτών. Επομένως και οι τιμές που λαμβάνουν οι μεταβλητές αυτές είναι μοναδικές για κάθε διαφορετικό αντικείμενο. Με μεταβλητές κλάσης μία συγκεκριμένη κλάση μπορεί να έχει μόνο ένα αντίγραφο των μεταβλητών αυτών, το οποίο μοιράζεται σε όλα τα αντικείμενα που θα δημιουργηθούν στη συνέχεια. Το σημαντικό με τις μεταβλητές αυτές είναι πως μπορούν να χρησιμοποιηθούν ακόμα και εάν δεν δημιουργηθεί κανένα αντικείμενο της κλάσης αυτής. Οι μεταβλητές αυτές καλούνται και στατικές (static fields), ενώ πριν από αυτές πρέπει πάντα να προηγείται το λεκτικό static.

Class	Μία ομαδοποίηση δεδομένων και κώδικα. Τα δεδομένα είναι προσβάσιμα με τις ιδιότητες. Ο κώδικας τους αναφέρεται σαν μέθοδος.
Object	Μια εκδοχή της κλάσης στην μνήμη. Μια εκδοχή δημιουργείται χρησιμοποιώντας την δήλωση <<Dim>> και την λέξη-κλειδί <<New>>.
Properties	Μια ρουτίνα που εκτελείται από ένα αντικείμενο για να προβάλλει δεδομένα και να επιτρέψει σε εξωτερικό κώδικα να τα επηρεάσει.
Method	Μια ενέργεια που εκτελείται από ένα αντικείμενο. Στην VB.NET οι μέθοδοι καθορίζονται σαν Subs και Functions.

Πίνακας 4: Στοιχεία OOP

Event-Driven Programming

Κατά την αλληλεπίδραση με μια εφαρμογή διαδραματίζονται συμβάντα (events). Συμβάντα ονομάζονται όλες εκείνες οι ενέργειες οι οποίες γίνονται από τον χρήστη κατά τη χρήση του προγράμματος. Για παράδειγμα, συμβάν για ένα κουμπί (button) το οποίο προσθέσαμε σε μια φόρμα, είναι το «κλικ» του ποντικιού, το «δεξί κλικ», το πέρασμα του ποντικιού από πάνω κτλ. Τα events είναι κάτι σαν σήματα από ένα μέρος της εφαρμογής σε κάποιο άλλο ή ενέργειες του χρήστη στις οποίες εμείς προσθέτουμε κάποιες λειτουργίες. Ορισμένοι τομείς κώδικα που αντιδρούν σε αυτά τα events ονομάζονται event handlers. Οι event handlers δεν μπορούν να είναι functions καθώς τα events δεν επιστρέφουν κάποια τιμή. Συνήθως ένας event handler ονομάζεται από το όνομα του αντικειμένου συν το όνομα του event. Μια πηγή συμβάντων (event source) είναι οποιοδήποτε αντικείμενο το οποίο μπορεί να στείλει επισήμανση της εμφάνισης ενός event. Η μετάδοση του σήματος λέγεται raising event και πολλές από τις κλάσεις της βιβλιοθήκης κλάσεων της .NET framework έχουν αυτή τη δυνατότητα.

Μέθοδοι

Μέθοδοι (sub) ονομάζονται κάποιες λειτουργίες οι οποίες είναι κατά το πλείστον προαποφασισμένου αποτελέσματος, ενώ εμείς προσθέτουμε απλά μια παράμετρο. μία μέθοδος αφορά ένα σώμα εντολών (με τη μορφή προγραμματιστικής συνάρτησης) οι οποίες επενεργούν σε κάποιες μεταβλητές. Μία μέθοδος μπορεί να δέχεται κάποια ορίσματα τιμών ενώ ταυτόχρονα μπορεί να επιστρέφει κάποιο αποτέλεσμα. Για παράδειγμα, μια ενδεικτική μέθοδος για

ένα κουμπί είναι η «move». Το τι θα κάνει είναι προαποφασισμένο. Θα μετακινήσει το κουμπί. Το πού όμως θα μετακινηθεί, το δίνει ο συντάκτης ή ο χρήστης του προγράμματος.

Οι μέθοδοι μιας κλάσης χωρίζονται σε δύο κατηγορίες σε μεθόδους αντικειμένων (Instance Methods) και μεθόδους κλάσης (Class Methods). Η πρώτη κατηγορία αφορά μεθόδους οι οποίες όταν καλούνται αναφέρονται σε ένα συγκεκριμένο αντικείμενο μιας κλάσης. Αντίθετα από ότι συμβαίνει με τις μεταβλητές αντικειμένων, στη μνήμη κρατείται ένα μόνο αντίγραφο για κάθε τέτοια μέθοδο. Κάθε φορά που ένα αντικείμενο εκτελεί μία τέτοια μέθοδο, ένας ειδικός μηχανισμός εξασφαλίζει την μοναδικότητα της εκτέλεσής της από το συγκεκριμένο αντικείμενο. Αντίθετα οι μέθοδοι κλάσης μπορούν να εκτελεστούν ακόμα και εάν δεν υπάρχουν αντικείμενα της συγκεκριμένης κλάσης. Για το λόγο αυτό καλούνται και στατικές (static methods) και ξεχωρίζουν προτάσσοντας το λεκτικό static.

Συναρτήσεις

Συναρτήσεις είναι διεργασίες, δηλαδή ένα σύνολο δηλώσεων που εκτελούν μια λειτουργία όταν καλούνται. Η βασική διαφορά τους με τις μεθόδους είναι ότι επιστρέφουν ένα αποτέλεσμα ενώ οι μέθοδοι δεν επιστρέφουν καμία τιμή. Οι συναρτήσεις μπορούν να επιστρέψουν τιμές είτε χρησιμοποιώντας την δήλωση return, είτε εκχωρώντας την τιμή στο όνομα της συνάρτησης. Οι συναρτήσεις μπορούν επίσης να καλέσουν τον εαυτό τους. Τέτοιες συναρτήσεις λέγονται αναδρομικές συναρτήσεις.

Μεταβλητές

Μεταβλητές ονομάζονται εκείνες οι παράμετροι τις οποίες χρησιμοποιούμε για να αποθηκεύουμε κάποια τιμή. Υπάρχουν εξ ορισμού διάφοροι τύποι μεταβλητών οι οποίοι χρησιμοποιούνται ανάλογα με τον τύπο των δεδομένων των οποίων επιθυμούμε να «αποθηκεύσουμε» την τιμή. Η χρήση μεταβλητών γίνεται ώστε να μπορούμε εύκολα να κατηγοριοποιήσουμε τα δεδομένα μας και δεύτερον για οικονομία μνήμης, επομένως και για την αύξηση της ταχύτητας επεξεργασίας. Οι μεταβλητές μπορούν να καταχωρίσουν οποιαδήποτε τιμή (όχι μόνο αριθμητική). Μπορεί να δέχονται ορίσματα (τιμές) ανάλογα με τον τύπο.

Πίνακες Μεταβλητών

Σε κάποιες διαδικασίες, χρειαζόμαστε ένα πλήθος μεταβλητών οι οποίες θα δέχονται ως όρισμα ιδίου τύπου δεδομένα ενώ σε κάποιες άλλες περιπτώσεις, μας είναι άγνωστο το πλήθος των μεταβλητών που θα απαιτηθούν όπως σε περιπτώσεις που έχουμε κάποιο αρχείο το οποίο περιέχει συντεταγμένες, και θέλουμε να το διαβάσουμε και να καταχωρίσουμε κάθε τιμή από αυτό το αρχείο σε μια ξεχωριστή μεταβλητή. Αρχικά δεν γνωρίζουμε πόσο μεγάλο είναι το αρχείο που θα ανοιχθεί (πόσες γραμμές / records περιέχει), αλλά μας ενδιαφέρει να διαβάσουμε όλα τα δεδομένα. Προφανώς δεν θα μπορούσαμε να γράψουμε δηλώσεις για κάθε μεταβλητή (μπορεί να χρειαζόταν εκατομμύρια τέτοιες δηλώσεις!!!). Σε αυτές τις περιπτώσεις η λύση δίνεται με τη χρήση πινάκων μεταβλητών. Οι τύποι των πινάκων μεταβλητών, καθώς και ο τρόπος δήλωσής τους, είναι ο ίδιος με τις απλές μεταβλητές. Αυτό που αλλάζει είναι η σύνταξη και η χρήση της ίδιας της μεταβλητής – πίνακα.

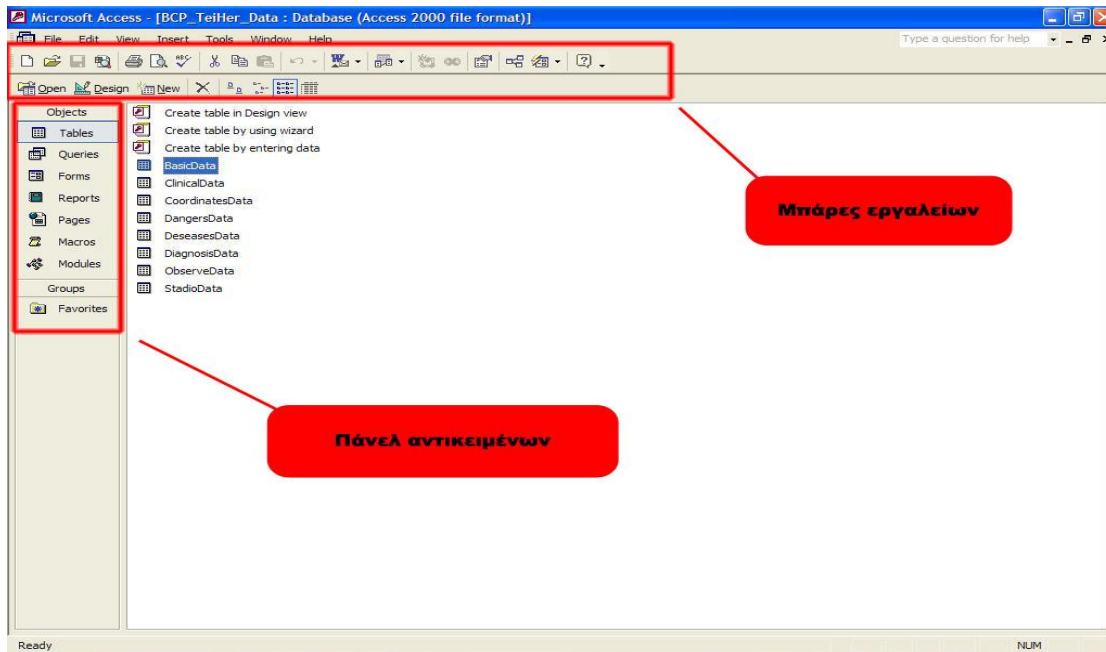
3.2 Microsoft Access

Η Microsoft Access έχει όλα τα χαρακτηριστικά ενός κλασικού συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) και αρκετά παραπάνω. Είναι ένα ευέλικτο και εύκολο στη χρήση RDBMS, καθώς και ένα πλήρες εργαλείο ανάπτυξης εφαρμογών για βάσεις δεδομένων. Μπορούμε να χρησιμοποιήσουμε την Access για να κατασκευάσουμε και να εκτελέσουμε μια εφαρμογή φτιαγμένη στα δικά μας μέτρα. Μπορούμε να περιορίζουμε, να επιλέγουμε και να προσθέτουμε τα δεδομένα με τη χρήση ερωτημάτων (Queries) ή ακόμα να δημιουργούμε φόρμες (Forms) για την εμφάνιση και την τροποποίηση των δεδομένων μας. Μπορούμε επίσης να δημιουργήσουμε απλές ή πολύπλοκες αναφορές (Reports). Οι φόρμες και οι αναφορές αναφέρονται σε κάποιον πίνακα ή σε κάποιο ερώτημα και έτσι «κληρονομούν» τις ιδιότητες του πίνακα ή του ερωτήματος αντίστοιχα. Αυτό σημαίνει ότι οι μορφές (Formats) και οι κανόνες εγκυρότητας (Validation Rules) χρειάζεται να οριστούν μόνο μία φορά κατά τον σχεδιασμό ενός πίνακα.

Ανάμεσα στα ισχυρότερα χαρακτηριστικά της Access είναι και οι Οδηγοί (Wizards), τους οποίους μπορούμε να χρησιμοποιούμε για την κατασκευή πινάκων και ερωτημάτων και για τον

ορισμό μιας μεγάλης ποικιλίας φορμών και αναφορών. Οι Οδηγοί αυτοί είναι έτοιμες σχεδιάσεις που έχει ενσωματωμένες η Access για να μας βοηθήσει στη σχεδίαση της βάσης δεδομένων μέσω κατανοητών βημάτων. Η Access περιλαμβάνει ακόμη την περιεκτική γλώσσα προγραμματισμού Microsoft Visual Basic for Applications (VBA), που μπορούμε να χρησιμοποιήσουμε για να κατασκευάσουμε πολύ ισχυρές εφαρμογές.

3.2.1 Περιβάλλον



Εικόνα 10: Περιβάλλον της ACCESS

Στο πάνελ των αντικειμένων εμφανίζονται τα αντικείμενα που περιέχονται στην βάση δεδομένων που μπορεί να είναι πίνακες και ερωτήματα. Εκτός από τους πίνακες, μια βάση δεδομένων μπορεί να περιέχει και άλλα αντικείμενα, όπως για παράδειγμα Φόρμες και Εκθέσεις. Μια βάση δεδομένων δεν μπορεί να περιέχει κάποιο άλλο αντικείμενο αν προηγουμένως δεν περιέχει τουλάχιστον έναν πίνακα.

3.2.2 Πίνακες

Πρόκειται για τον δομικό λίθο μιας ΒΔ γι' αυτό και το βασικό παράθυρο της ΒΔ ξεκινάει από αυτόν. Είναι ένα αντικείμενο που ορίζουμε και το χρησιμοποιούμε για την αποθήκευση των δεδομένων μας. Κάθε πίνακας περιέχει πληροφορίες για ένα συγκεκριμένο θέμα, όπως είναι οι πελάτες, οι παραγγελίες τους, οι μαθητές κ.ά. Οι πίνακες περιέχουν πεδία (fields) ή στήλες (columns), όπου αποθηκεύονται τα διαφορετικά είδη πληροφοριών, όπως είναι το όνομα ενός πελάτη ή ο βαθμός ενός μαθητή και εγγραφές (records) ή γραμμές (rows) που περιέχουν όλες τις πληροφορίες για μια συγκεκριμένη περίπτωση του πίνακα, όπως π.χ. όλες οι πληροφορίες για έναν μαθητή που ονομάζεται Αντωνιάδης. Σε κάθε πίνακα μπορούμε να ορίσουμε ένα βασικό ή πρωτεύον κλειδί (primary key), που είναι ένα ή περισσότερα πεδία που χαρακτηρίζουν μοναδικά την εγγραφή μέσα στον πίνακα και ένα ή περισσότερα ευρετήρια (indexes) για να μπορούμε να αυξήσουμε την ταχύτητα πρόσβασης στα δεδομένα μας. Το πρωτεύον κλειδί μπορεί να είναι ο κωδικός ενός πελάτη, το Α.Φ.Μ. ενός φορολογούμενου, ο αριθμός μητρώου ενός μαθητή, η πινακίδα ενός αυτοκινήτου κ.ά.. Π.χ. σε μια βάση δεδομένων όπου καταχωρούνται στοιχεία όπως όνομα, επώνυμο, διεύθυνση, τηλέφωνο, μια εγγραφή είναι μια καταχώρηση που αφορά ένα συγκεκριμένο άτομο με τα στοιχεία του, ενώ πεδία είναι το όνομα, το επώνυμο, η διεύθυνση και το τηλέφωνο.

ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΔΙΕΥΘΥΝΣΗ	ΤΗΛΕΦΩΝΟ
Ιωάννης	Παπαδόπουλος	Κωλοκοτρώνη 18	2101000000

Πίνακας 5: Παράδειγμα πίνακα βάσης δεδομένων

Η δημιουργία μίας βάσης δεδομένων ξεκινάει από τη δημιουργία πίνακα. Για μια βάση δεδομένων τηλεφωνικού καταλόγου ο αρχικός πίνακας μπορεί να έχει τα πεδία ΚΩΔΙΚΟΣ, ΟΝΟΜΑ, ΕΠΩΝΥΜΟ, ΟΔΟΣ, ΑΡΙΘΜΟΣ, ΤΚ, ΤΗΛΕΦΩΝΟ. Επιλέγοντας <<Προβολή σχεδίασης>>, εμφανίζεται ο πίνακας που δημιουργήθηκε σε προβολή σχεδίασης απ' που μπορούν να ρυθμιστούν με λεπτομέρειες όλες τις ιδιότητες του πίνακα. Σε προβολή φύλλου δεδομένων είναι δυνατή η εισαγωγή δεδομένων στον πίνακα κατά την οποία οι εγγραφές αποθηκεύονται στον πίνακα τη στιγμή που πληκτρολογούνται. Μια ΒΔ δεν μπορεί να περιέχει κάποιο άλλο αντικείμενο αν προηγουμένως δεν περιέχει τουλάχιστον έναν πίνακα. Φυσικά και οι ίδιοι οι πίνακες της Access υπακούουν σε κάποιους περιορισμούς:

Ιδιότητα	Μέγιστη τιμή
Αριθμός χαρακτήρων στο όνομα πίνακα	64
Αριθμός χαρακτήρων στο όνομα πεδίου	64
Αριθμός πεδίων πίνακα	255
Αριθμός ανοιχτών πινάκων	2048
Μέγεθος πίνακα	2 GB
Αριθμός αριθμητών (indexes) σε πίνακα.	32
Αριθμός πεδίων αριθμητή (index)	10
Αριθμός χαρακτήρων σε μήνυμα επιβεβαίωσης.	255
Αριθμός χαρακτήρων σε κανόνα επιβεβαίωσης.	2.048
Αριθμός χαρακτήρων σε πίνακα ή πεδίο περιγραφής.	255
Αριθμός χαρακτήρων για κάθε εγγραφή (εκτός πεδίων Memo και OLE Object) όταν η ιδιότητα Unicode Compression των πεδίων έχει τιμή Yes.	4.000
Αριθμός χαρακτήρων σε πεδίο ιδιοτήτων.	255

Πίνακας 6: Ιδιότητες πίνακα της Access

3.2.3 Φόρμες

Είναι ένα αντικείμενο που χρησιμεύει κατά κύριο λόγο για την εισαγωγή και την εμφάνιση των δεδομένων ή για τον έλεγχο της εκτέλεσης της εφαρμογής. Μπορούμε να χρησιμοποιούμε φόρμες για να έχουμε μια ωραία παρουσίαση των δεδομένων που προέρχονται από ερωτήματα ή ακόμα και πίνακες. Οι φόρμες μπορούν ακόμα να εκτελούν μακροεντολές ή διαδικασίες της γλώσσας VBA σαν απόκριση σε κάποια συμβάντα όπως για παράδειγμα να υπολογίζεται η αξία του ΦΠΑ σε μια παραγγελία. Σε μια βάση δεδομένων οι φόρμες χρησιμεύουν για την παρουσίαση των εγγραφών της βάσης ή για την τροποποίηση τους ή ακόμη και για προσθήκη νέων εγγραφών στη βάση. Η παρουσίαση, η τροποποίηση και η προσθήκη γίνεται ανά εγγραφή.

Το αν μια φόρμα θα έχει τη δυνατότητα μόνο για προβολή των εγγραφών της βάσης δεδομένων ή θα έχει και τη δυνατότητα για τροποποίησή τους ή ακόμη και για προσθήκη νέων εγγραφών στη βάση, ρυθμίζεται από τις αντίστοιχες επιλογές στο Φύλλο ιδιοτήτων της φόρμας. Το κείμενο που εμφανίζεται στην γραμμή τίτλου του παραθύρου της φόρμας ονομάζεται λεζάντα φόρμας και η ρύθμισή του γίνεται από την ομώνυμη επιλογή στο Φύλλο ιδιοτήτων της φόρμας. Η εμφάνιση ή μη των κουμπιών μετακίνησης της φόρμας ρυθμίζεται από την ομώνυμη επιλογή στο Φύλλο ιδιοτήτων της φόρμας.

Σε μια φόρμα μπορούμε να βάλουμε διάφορα κουμπιά εντολών. Π.χ. για πλοήγηση στις εγγραφές, για προσθήκη νέας εγγραφής, διαγραφή εγγραφής, εκτύπωση της εγγραφής, έξοδο από τη βάση δεδομένων κλπ. Μέσω μιας εργαλειοθήκης της καρτέλας << Σχεδίαση των Εργαλείων

σχεδίασης φόρμας>> μπορούμε να βάλουμε κουμπιά εντολών σε μια φόρμα, πρέπει προηγουμένως να την ανοίξουμε σε προβολή σχεδίασης. Μέσω των επιλογών της Access μπορούμε αν θέλουμε να ρυθμίσουμε τη βάση έτσι ώστε όταν γίνεται διπλό κλικ στο αρχείο της να εμφανίζεται αυτόματα μια φόρμα.

3.2.4 Ερωτήματα

Είναι ένα αντικείμενο που «απομονώνει» ότι στοιχεία θέλουμε και μας δίνει μια συγκεκριμένη άποψη των δεδομένων μας, η οποία άποψη μπορεί να προέρχεται από έναν ή περισσότερους πίνακες. Μπορούμε να ορίσουμε ερωτήματα για να δημιουργήσουμε νέους πίνακες από τα δεδομένα ενός ή περισσότερων ήδη υπάρχοντων πινάκων. Για παράδειγμα, μπορούμε να ορίσουμε ένα ερώτημα που θα παίρνει δεδομένα από τους πίνακες πελατών και παραγγελιών και θα δημιουργεί έναν νέο πίνακα, που θα αναφέρεται όμως σαν ερώτημα, όπου θα περιέχονται τα στοιχεία επώνυμο, όνομα και ποσότητα παραγγελίας από τους πελάτες που έκαναν παραγγελίες ενός συγκεκριμένου προϊόντος τον περασμένο μήνα. Τα στοιχεία που δημιουργεί αυτό το ερώτημα μπορούμε μετά να τα επεξεργαστούμε σαν έναν νέο πίνακα.

Τα ερωτήματα χρησιμεύουν στο να εμφανίζουν εκείνες τις εγγραφές από μια βάση δεδομένων, οι οποίες πληρούν κάποια κριτήρια (αυτά που καθορίζει ο χρήστης). Π.χ. σε μια βάση δεδομένων τηλεφωνικού καταλόγου, ένα ερώτημα μπορεί να ταξινομεί τις εγγραφές κατά αλφαβητική σειρά, ένα άλλο ερώτημα να εμφανίζει συγκεκριμένα ονόματα (π.χ. τους Νίκους ή τις Μαρίες κλπ.), ένα άλλο ερώτημα να εμφανίζει όλους όσοι μένουν σε μια συγκεκριμένη διεύθυνση κλπ. Η δημιουργία ενός ερωτήματος μπορεί να γίνει εύκολα με χρήση drag and drop των πεδίων που θέλουμε να χρησιμοποιήσουμε και τσεκάροντας ποια πεδία θα εμφανίζονται κατά την εκτέλεση του καθώς και τον τρόπο ταξινόμησης τους. Διαφορετικά για πιο σύνθετα ερωτήματα μπορεί να γραφτεί απευθείας κώδικας με γλώσσα SQL.

```
SELECT Count (DangersData.OikogeniakoIstorikoKarkinouMastouYN)
FROM DangersData
WHERE DangersData.OikogeniakoIstorikoKarkinouMastouYN="Να1";
UNION ALL SELECT Count (DangersData.OikogeniakoIstorikoAllonKarkinonYN)
FROM DangersData
WHERE DangersData.OikogeniakoIstorikoAllonKarkinonYN="Να1";
```

3.2.5 Αντικείμενα της Access

Αναφορά (Report)

Είναι ένα αντικείμενο σχεδιασμένο για τη μορφοποίηση, την εκτέλεση υπολογισμών, την εκτύπωση και τη σύνοψη κάποιων επιλεγμένων δεδομένων. Πριν τυπώσουμε μια αναφορά, μπορούμε να τη δούμε στην οθόνη μας (preview ή προεπισκόπηση).

Μακροεντολή (Macro)

Είναι ένα αντικείμενο που αποτελεί το δομημένο ορισμό μιας ή περισσότερων ενεργειών που θέλουμε να εκτελέσει η Access σαν απόκριση σ' ένα ορισμένο συμβάν. Για παράδειγμα, μπορούμε να σχεδιάσουμε μια μακροεντολή που θα ανοίγει μια δεύτερη φόρμα, σαν απόκριση στην επιλογή ενός στοιχείου της κύριας φόρμας. Μπορούμε επίσης να έχουμε μια μακροεντολή που θα ελέγχει την εγκυρότητα των στοιχείων ενός πεδίου όταν θα κάνουμε αλλαγές σ' αυτά. Μπορούμε ακόμα να περιλάβουμε συνθήκες στις μακροεντολές για να ορίζουμε τότε πρέπει να εκτελεστούν κάποιες ενέργειες των μακροεντολών και τότε όχι.

Μπορούμε να χρησιμοποιήσουμε μακροεντολές για το άνοιγμα και την εκτέλεση ερωτημάτων, για το άνοιγμα πινάκων ή για την εκτύπωση ή την εμφάνιση αναφορών. Ακόμα, μπορούμε μέσα από μια μακροεντολή, να εκτελούμε άλλες μακροεντολές ή διαδικασίες της VBA.

Υπομονάδα (Module)

Είναι ένα αντικείμενο που περιέχει διαδικασίες (procedures ή functions) που τις ορίσαμε εμείς με τη χρήση της VBA. Οι υπομονάδες παρέχουν μια πιο διακριτική ροή των ενεργειών και μας επιτρέπουν να παγιεύουμε τα λάθη, κάτι που δεν μπορούμε να κάνουμε με τις μακροεντολές. Μπορεί να είναι αυτόνομα αντικείμενα, με συναρτήσεις που μπορούν να κληθούν από οποιοδήποτε σημείο

της εφαρμογής μας, ή μπορεί να συσχετίζονται απευθείας με τις φόρμες ή τις αναφορές για να αποκρίνονται μόνο στα συμβάντα των φορμών και των αναφορών.

Συμβάντα (events)

Συμβάν είναι κάθε αλλαγή στην κατάσταση ενός αντικειμένου της Access. Για παράδειγμα, μπορούμε να γράψουμε μακροεντολές ή διαδικασίες της VBA για φόρμες που θα ανταποκρίνονται στο άνοιγμα της φόρμας, στο κλείσιμο της φόρμας, στην εισαγωγή μιας νέας γραμμής στη φόρμα ή στην αλλαγή των δεδομένων της τρέχουσας εγγραφής κάθε αντικειμένου μιας φόρμας ή αναφοράς που περιέχει δεδομένα. Μπορούμε να σχεδιάσουμε μια μακροεντολή ή διαδικασία VBA που να αποκρίνεται ακόμα και στο πάτημα κάποιων συγκεκριμένων πλήκτρων κατά την εισαγωγή δεδομένων.

3.2.6 Πρωτεύοντα και Ξένα Κλειδιά

Σ' έναν σωστό σχεδιασμό μιας σχεσιακής βάσης δεδομένων, κάθε εγγραφή οποιουδήποτε πίνακα πρέπει να είναι μοναδική, δηλ. κάποιο πεδίο (ή συνδυασμός πεδίων) πρέπει να περιέχει διαφορετική τιμή σε κάθε εγγραφή του πίνακα για να μπορούμε έτσι να ξεχωρίζουμε την εγγραφή. Αυτό το μοναδικό αναγνωριστικό πεδίο ονομάζεται πρωτεύον κλειδί (primary key). Όταν οργανώνουμε τα δεδομένα σε μια βάση δεδομένων, το αποτέλεσμα είναι πολλοί ξεχωριστοί πίνακες. Για να συνδέσουμε δύο πίνακες, για παράδειγμα, τον πίνακα πελατών και τον πίνακα παραγγελιών, θα χρησιμοποιήσουμε το πεδίο κωδικός πελάτη και στους δύο πίνακες. Στον πίνακα πελατών θα είναι βέβαιο το πρωτεύον κλειδί, καθώς δεν μπορεί να υπάρχουν δύο ή περισσότεροι πελάτες με τον ίδιο κωδικό πελάτη, αλλά στον πίνακα παραγγελιών θα μπορεί να υπάρχουν πολλές εγγραφές με τον ίδιο κωδικό πελάτη, καθώς ο κάθε πελάτης κάνει πολλές παραγγελίες. Αυτά τα πεδία «σύνδεσης» ονομάζονται ξένα κλειδιά (foreign keys). Όταν ορίζουμε τους πίνακές μας στην Access, συνδέουμε τα πρωτεύοντα κλειδιά με τα ξένα κλειδιά, για να δηλώσουμε στην Access πώς πρέπει να συνδυάσει τα δεδομένα όταν θέλουμε πληροφορίες από περισσότερους από έναν πίνακες.

Μονοσήμαντες και Αμφιμονοσήμαντες Σχέσεις

Συνήθως, η αποδοτικότερη σχέση ανάμεσα σε δύο πίνακες είναι η μονοσήμαντη («ένα προς πολλά», one-to-many). Αυτό σημαίνει ότι για κάθε εγγραφή του πρώτου πίνακα, υπάρχουν πολλές συσχετισμένες εγγραφές στον δεύτερο, αλλά σε κάθε εγγραφή του δεύτερου πίνακα αντιστοιχεί μόνο μία εγγραφή του πρώτου. Για παράδειγμα, κάθε μαθητής μπορεί να έχει πολλές βαθμολογίες στον πίνακα Μαθητής-Τεστ, αλλά κάθε εγγραφή του πίνακα Μαθητής-Τεστ αντιστοιχεί σ' έναν μόνο μαθητή. Σε μια βάση δεδομένων με πελάτες και παραγγελίες, ένας πελάτης μπορεί να κάνει πολλές παραγγελίες, αλλά μία παραγγελία ανήκει μόνο σ' έναν πελάτη. Η αμφιμονοσήμαντη σχέση είναι η «ένα προς ένα», one-to-one, δηλ., σε κάθε εγγραφή του ενός πίνακα αντιστοιχεί μία και μόνο μία εγγραφή του άλλου πίνακα και το αντίστροφο. Για παράδειγμα, σε μια βάση δεδομένων με πίνακα Κρατών και πίνακα Πρωτευουσών, υπάρχει μια αμφιμονοσήμαντη σχέση. Σ' αυτή την περίπτωση, μπορούμε να έχουμε έναν μόνον πίνακα Κρατών και Πρωτευουσών.

Δημιουργία Συνδέσμων Πινάκων

Η Access δεν μας επιτρέπει να δημιουργήσουμε μια πολυσήμαντη σχέση άμεσα μεταξύ δύο πινάκων. Αν μια σχέση διαπιστώσουμε ότι είναι «πολλά προς πολλά» (many-to-many), θα πρέπει να δημιουργήσουμε έναν ξεχωριστό πίνακα διασταύρωσης (intersection table) για να μπορέσουμε να χειριστούμε τη σχέση αποτελεσματικά. Για παράδειγμα, ένας μαθητής μπορεί να προτιμά πολλά κολέγια και ένα κολέγιο μπορεί να προτιμάται από πολλούς μαθητές. Η λύση είναι η δημιουργία του πίνακα Μαθητές-κολέγια, που είναι ένας πίνακας διασταύρωσης, έχει μονοσήμαντη σχέση και με τους δύο πίνακες, Μαθητές και κολέγια, και μας λύνει τα χέρια. Σ' ένα άλλο παράδειγμα, ένας ιδιοκτήτης μπορεί να έχει πολλά οικόπεδα στην κατοχή του, αλλά ένα ποσοστό από κάθε οικόπεδο, και ένα οικόπεδο μπορεί να είναι ιδιοκτησία πολλών. Η λύση είναι η δημιουργία ενός πίνακα διασταύρωσης που θα περιέχει σαν πεδία τον Κωδικό Ιδιοκτήτη, τον Κωδικό Οικοπέδου και το Ποσοστό Ιδιοκτησίας του κάθε ιδιοκτήτη. Για να ολοκληρώσουμε τους συνδέσμους μεταξύ των πινάκων, τοποθετούμε ένα αντίγραφο του πρωτεύοντος κλειδιού από τον πίνακα με σχέση «ένα» στον πίνακα με σχέση «πολλά».

Τα Ερωτήματα Ενέργειας

Με τα ερωτήματα ενεργείας (action queries) μπορούμε γρήγορα, εύκολα και με ασφάλεια να τροποποιήσουμε, εισάγουμε, δημιουργήσουμε ή και να διαγράψουμε κάποια δεδομένα από τη βάση δεδομένων μας. Αν θέλουμε να κάνουμε μια αλλαγή σε μια εγγραφή σ' έναν πίνακα ή σ' ένα ερώτημα, αυτό είναι αρκετά εύκολο να γίνει. Εντοπίζουμε την εγγραφή με τους γνωστούς τρόπους και κάνουμε τη διόρθωση. Το πρόβλημα προκύπτει όταν χρειαστεί να κάνουμε την ίδια αλλαγή σε πολλές εγγραφές. Το να κάνουμε τις τροποποιήσεις σε κάθε μία εγγραφή ξεχωριστά και κουραστικό είναι και επικίνδυνο για λάθη. Είναι καλή πρακτική, πριν δημιουργήσουμε και εκτελέσουμε ένα ερώτημα ενέργειας για να ενημερώσουμε ή και να διαγράψουμε πολλές εγγραφές της βάσης δεδομένων μας, να δημιουργούμε πρώτα ένα επιλεκτικό ερώτημα, όπως μάθαμε στο προηγούμενο κεφάλαιο, με κριτήρια που θα επιλέγουν τις εγγραφές που θέλουμε να ενημερώσουμε. Αφού λοιπόν, βεβαιωθούμε ότι η Access επέλεξε σωστά τις εγγραφές που θέλουμε, τότε είναι εύκολο να μετατρέψουμε αυτό το επιλεκτικό ερώτημα σε ερώτημα ενέργειας για να κάνουμε τη δουλειά μας.

3.2.6 Δημιουργίας μιας Βάσης Δεδομένων στην Access

Το .mdb είναι η επέκταση που έχουν τα αρχεία βάσεων δεδομένων της Access. Μόλις εκκινήσουμε την Access, διαλέγουμε το πλήκτρο επιλογής Άνοιγμα υπάρχουσας βάσης δεδομένων, μετά διαλέγουμε τη βάση δεδομένων που θέλουμε να επεξεργαστούμε είτε διπλοπατώντας πάνω στο όνομά της με ποντίκι ή το πατάμε μια φορά με το ποντίκι και μετά πατάμε στο πλήκτρο OK. Ξεκινάει η εφαρμογή και βλέπουμε το παράθυρο της βάσης δεδομένων της. Για μια υπάρχουσα βάση δεδομένων, το παράθυρο Βάσης Δεδομένων που φαίνεται στην προηγούμενη εικόνα παραμένει πάντα στο ίδιο σημείο της οθόνης που το τοποθετήσαμε την τελευταία φορά. Η γραμμή τίτλου του παραθύρου μάς δείχνει το όνομα της βάσης δεδομένων που ανοίξαμε. Παρόλο που μπορούμε να έχουμε ανοιχτή μόνο μία βάση δεδομένων κάθε φορά, μπορούμε να τη συνδέουμε με πίνακες άλλων βάσεων δεδομένων της Access. Για τα πλήκτρα που εμφανίζονται στη γραμμή εργαλείων, μπορούμε να τοποθετήσουμε τον δείκτη του ποντικιού πάνω σ' ένα πλήκτρο, χωρίς να το πατήσουμε, και η Access μάς εμφανίζει μια σύντομη περιγραφή (ToolTip) για το τι κάνει αυτό το πλήκτρο.

Η Access έχει περισσότερες από δώδεκα έτοιμες γραμμές εργαλείων και μάς δείχνει κάθε φορά τη γραμμή εργαλείων που διαθέτει τα καταλληλότερα πλήκτρα για την εργασία που κάνουμε τη συγκεκριμένη στιγμή. Παρόλα αυτά είναι δυνατή η επιλογή των γραμμών εργαλείων που θα είναι ενεργές καθώς και ο ορισμός των πλήκτρων που θα εμφανίζονται σε αυτές. Μπορούμε, ακόμα, να ορίσουμε δικές μας γραμμές εργαλείων που θα είναι συνέχεια ενεργές ή θα ανοίγουν και θα κλείνουν από μακροεντολές ή από υπομονάδες. Στο πάνω μέρος του παραθύρου Βάσης Δεδομένων υπάρχουν οι δείκτες καρτελών (tabs) που μας επιτρέπουν να διαλέξουμε έναν από τους έξι βασικούς τύπους αντικειμένων της Access : πίνακες, ερωτήματα, φόρμες, αναφορές, μακροεντολές ή υπομονάδες.

3.2.7 Αναφορά σε τύπους δεδομένων μιας βάσης Access.

Μία εγγραφή σε έναν πίνακα της Access αποτελείται από πεδία τιμών. Κάθε πίνακας μπορεί να περιέχει πολλά πεδία διαφορετικού τύπων δεδομένων. Ο τύπος δεδομένων κάθε πεδίου υποδεικνύει το είδος των δεδομένων που αποθηκεύονται στο πεδίο.

Τύπος δεδομένου v	Περιγραφή	Όρια	Αποθήκευση σε Bytes
Text	Μικρές σε μέγεθος αλφαριθμητικές τιμές.	255 χαρακτήρες	
Memo	Μεγάλο κείμενο.	65.536 χαρακτήρες	
Byte	Ακέραιοι αριθμοί	Αριθμοί από 0 έως 255	1 byte
Integer	Ακέραιοι αριθμοί.	Αριθμοί από -32,768 έως 32,767	2 bytes

Long	Ακέραιοι αριθμοί.	Αριθμοί από -2,147,483,648 έως 2,147,483,647	4 bytes
Single	Ακέραιοι, δεκαδικοί σταθερής υποδιαστολής και δεκαδικοί κινητής υποδιαστολής.	Για τους περισσότερους δεκαδικούς.	4 bytes
Double	Ακέραιοι, δεκαδικοί σταθερής υποδιαστολής και δεκαδικοί κινητής υποδιαστολής.	Για τους περισσότερους δεκαδικούς.	8 bytes
Currency	Νομισματικές μονάδες.	15 ψηφία συν 4 δεκαδικά.	8 bytes
AutoNumber	Αριθμοί οι οποίοι αυτομάτως παράγονται κατά αύξουσα σειρά για κάθε εγγραφή.		4 bytes
Date/Time	Ημερομηνία ή χρόνος.		8 bytes
Yes/No	Δίτιμες τιμές (Boolean).		1 bit
Ole Object	Αντικείμενο διασύνδεσης και ενσωμάτωσης. Μπορεί να είναι οποιοσδήποτε τύπος αρχείου.		Εώς 1GB
Hyperlink	Υπερσύνδεσμος.		
Lookup Wizard	Λίστα επιλογών από drop-down λίστα.		4 bytes

Πίνακας 7: Τύποι δεδομένων της ACCESS

Από αυτούς τους τύπους δεδομένων ορισμένοι χρίζουν περισσότερης ανάλυσης λόγω της συχνότητας χρήσης τους.

Property Use

Caption	Η ετικέτα που εμφανίζεται για αυτό το πεδίο στις φόρμες, τις αναφορές και τα ερωτήματα. Αν η ιδιότητα είναι κενή χρησιμοποιείται το όνομα του πεδίου.
Field Size	Καθορίζει τον χώρο που θα καταλαμβάνει η κάθε τιμή. Η προκαθορισμένη τιμή είναι Long Integer.
Format	Αν το πεδίο δεν χρησιμοποιείται ως πρωτεύων κλειδί, η ιδιότητα αυτή μπορεί να έχει τιμή που ταιριάζει στις ανάγκες μας.
Indexed	Καθορίζει αν το πεδίο αριθμείται.
New values	Καθορίζει αν η αυτόματη αρίθμηση θα γίνεται με αύξουσα ή τυχαία σειρά.

Πίνακας 8: Ιδιότητες του Autonumber

Property Use

Allow Zero Length	Καθορίζει αν το πεδίο θα μπορεί να είναι κενό.
Caption	Η ετικέτα που εμφανίζεται για αυτό το πεδίο στις φόρμες, τις αναφορές και τα ερωτήματα.
Default Value	Καθορίζει αυτομάτως μια προεπιλεγμένη τιμή για το πεδίο.
Field Size	Η ιδιότητα αυτή μπορεί να έχει τιμή από 1 έως 255. Text fields can range from 1 to 255 ανάλογα με το μέγεθος που θέλουμε να έχει η γραμματοσειρά του πεδίου.
Format	Καθορίζει τον τρόπο με τον οποίο εμφανίζεται ή εκτυπώνεται το πεδίο.

IME Mode	Ελέγχει τον τρόπο εμφάνισης χαρακτήρων σε Ασιατικές εκδόσεις Windows.
IME Sentence Mode	Ελέγχει τον τρόπο εμφάνισης φράσεων σε Ασιατικές εκδόσεις Windows.
Indexed	Καθορίζει αν το πεδίο θα είναι αριθμημένο.
Required	Ελέγχει αν το πεδίο θα έχει υποχρεωτικά τιμή.
Unicode Compression	Ελέγχει αν το κείμενο του πεδίου θα συμπιέζεται.
Validation Rule	Παρέχει κανόνα επιβεβαίωσης προκειμένου να καταχωρηθεί τιμή.
Validation Text	Παρέχει το κείμενο που εμφανιστεί όταν δεν θα επιβεβαιώνεται ο κανόνας επιβεβαίωσης.

Πίνακας 9: Ιδιότητες κειμένου

Property	Use
Caption	Η ετικέτα που εμφανίζεται για αυτό το πεδίο στις φόρμες, τις αναφορές και τα ερωτήματα.
Decimal Places	Καθορίζει τον αριθμό των δεκαδικών ψηφίων που θα εμφανίζονται.
Default Value	Καθορίζει αυτομάτως μια προεπιλεγμένη τιμή για το πεδίο.
Field Size	Έχουμε τις επιλογές Byte, Integer, Long Integer
Format	Καθορίζει τον τρόπο με τον οποίο εμφανίζεται ή εκτυπώνεται το πεδίο.
Indexed	Καθορίζει αν το πεδίο θα είναι αριθμημένο.
Input Mask	Καθορίζει αν θα εμφανίζονται επιλεγμένοι χαρακτήρες στις τιμές του πεδίου.
Required	Ελέγχει αν το πεδίο θα έχει υποχρεωτικά τιμή.
Validation Rule	Παρέχει κανόνα επιβεβαίωσης προκειμένου να καταχωρηθεί τιμή.
Validation Text	Παρέχει το κείμενο που εμφανιστεί όταν δεν θα επιβεβαιώνεται ο κανόνας επιβεβαίωσης.

Πίνακας 10: Ιδιότητες number

3.3 OLE DB τεχνολογία διασύνδεσης Βάσης Δεδομένων με εφαρμογή

Το OLE DB (Object Linking and Embedding Database) είναι ένα API (application programming interface) σχεδιασμένο από την Microsoft που επιτρέπει την πρόσβαση δεδομένων από ποικίλες πηγές με έναν ενοποιημένο τρόπο. Το API παρέχει ένα σύνολο μεθόδων που βασίζονται στο Component Object Model (COM). Αρχικά σχεδιάστηκε ως εναλλακτική επιλογή και διάδοχος του ODBC, επεκτείνοντας την λειτουργικότητά του ώστε να υποστηρίζονται και μη σχεσιακές βάσεις δεδομένων όπως βάσεις αντικειμένων και λογιστικά φύλλα που δεν χρησιμοποιούν απαραίτητα SQL. Το OLE DB διαχωρίζει την αποθήκευση των υπό προσπέλαση δεδομένων από την εφαρμογή, μέσω εννοιών όπως είναι η πηγή των δεδομένων, οι εντολές και οι εγγραφές. Ο λόγος της χρήσης αυτών των εννοιών είναι ότι διαφορετικές εφαρμογές χρειάζονται πρόσβαση σε διαφορετικές πηγές και τύπους δεδομένων χωρίς να είναι αναγκαία η γνωστή η δομή της βάσης. Το OLE DB μπορεί να παίζει τον ρόλο του παρόχου δεδομένων στις εφαρμογές που ζητούν δεδομένα για την λειτουργία τους ή εκείνου που ζητάει δεδομένα από εφαρμογές με τις οποίες συνδέεται καθώς επίσης υποστηρίζει απευθείας σύνδεση με SQL βάσεις δεδομένων.

Το αντικείμενο Σύνδεσης

Το αντικείμενο σύνδεσης είναι αυτό που χρειαζόμαστε ώστε να συνδεθούμε σε μια βάση δεδομένων. Υπάρχουν διάφορα αντικείμενα σύνδεσης για να χρησιμοποιήσουμε ανάλογα με το είδος της βάσης στην οποία συνδεόμαστε. Για να συνδεθούμε σε μια βάση τύπου Access χρησιμοποιούμε

ένα αντικείμενο το οποίο ονομάζεται "OLEDBconnection". Για παράδειγμα για να αρχικοποιήσουμε το αντικείμενο αυτό στην VB χρησιμοποιούμε την παρακάτω εντολή:

```
Dim con As New OleDb.OleDbConnection
```

Το αντικείμενο con που αρχικοποιήσαμε παραπάνω παρέχει πολλές ιδιότητες και μεθόδους. Η πιο βασική ιδιότητα ονομάζεται ConnectionString στην οποία θα πρέπει να περάσουμε δύο πράγματα: Την τεχνολογία που θα χρησιμοποιήσουμε για να συνδεθούμε με την βάση και το μονοπάτι της βάσης. Ένα παράδειγμα υλοποίησης είναι το εξής:

```
Dim dbProvider As String
Dim dbSource As String
dbProvider = "PROVIDER=Microsoft.Jet.OLEDB.4.0;"
dbSource = "Data Source = C:/AddressBook.mdb"
con.ConnectionString = dbProvider & dbSource
```

Η τεχνολογία είναι το dbProvider όπου στην συγκεκριμένη περίπτωση η τεχνολογία σύνδεσης είναι η JET και το dbSource η διεύθυνση της βάσης.

Αφού έχουμε καθορίσει την παράμετρο ConnectionString, μπορούμε να ανοίξουμε την σύνδεση. Αυτό γίνεται με την χρήση της μεθόδου Open του αντικειμένου Connection. Στο δικό μας παράδειγμα θα πρέπει να γράψουμε την εντολή con.Open() κάθε φορά και αφού τελειώσουμε με την χρήση του αντικειμένου Connection θα πρέπει να ακολουθεί το κλείσιμο της σύνδεσης με την εντολή con.Close() .

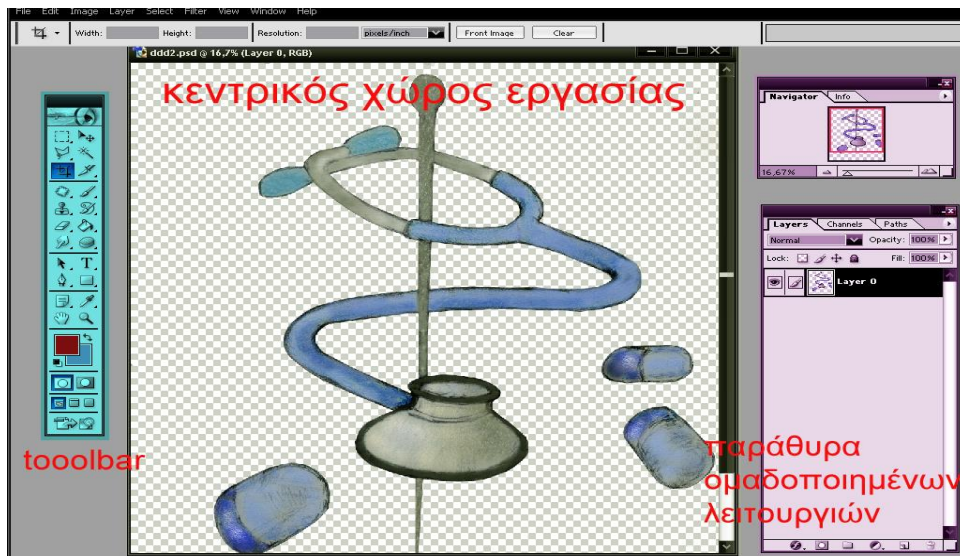
Data Adapter	Χρησιμοποιείται για την επικοινωνία dataset και βάσης δεδομένων. Δεν επηρεάζει την βάση από μόνο του.
Command builder	Χρησιμοποιείται για την ανανέωση της βάσης.
SQL Adapter	Δημιουργεί τα SQL statements INSERT, DELETE και UPDATE.
Command Object	Έχει property το command Text που είναι string.Χρησιμοποιεί τις μεθόδους Execute.nonQuery(), Execute.Reader(), Execute.Scalar(). Η ExecutenonQuery δεν επιστρέφει αποτελέσματα ενώ η ExecuteScalar επιστρέφει μια μοναδική τιμή της πρώτης γραμμής και πρώτης στήλης.
DataSet	Αποθηκεύονται σε αυτά σε μορφή άορατου πίνακα, καταχωρήσεις της βάσεις που επιλέξαμε να χειριστούμε.
DataTable	Παρόμοιο με DataSet με τη διαφορά ότι μπορούμε με αυτό μόνο να διαβάσουμε τις καταχωρήσεις.

Πίνακας 11: Αντικείμενα του OLE DB

3.4 Photoshop

Το Photoshop της Adobe είναι ίσως το πιο γνωστό πρόγραμμα του πακέτου που έχει να μας προσφέρει. Χρησιμοποιείται από επαγγελματίες γραφίστες μέχρι και ερασιτέχνες φωτογράφους. Είναι ότι καλύτερο κυκλοφορεί στην αγορά για όποιον θέλει να δημιουργήσει εικόνες, γραφικά και να επεξεργαστεί φωτογραφικές εικόνες. Περιέχει βασικά δύο ομάδες εργαλείων, μια για ζωγραφική και μια για επεξεργασία εικόνας. Το Photoshop είναι ένα πρόγραμμα κατασκευής bitmap γραφικών και επεξεργασίας εικόνας. Με το Photoshop μπορούμε να επέμβουμε ώστε να διορθώσουμε μία φωτογραφία, να της αλλάξουμε το χρώμα, διαστάσεις και γενικά να την επεξεργαστούμε. Όταν επεξεργαζόμαστε μια εικόνα στο PhotoShor, μπορούμε να την οξύνουμε (sharpen) για να διορθώσουμε την εστίασή της, να θολώσουμε το φόντο της, να αλλάξουμε τη φωτεινότητα και την αντίθεσή της ή και να αντικαταστήσουμε ένα χρώμα μ' ένα άλλο. Μπορούμε ακόμη να αποσπάσουμε ένα κομμάτι από μια εικόνα και να το αντιγράψουμε, να του αλλάξουμε μέγεθος και γενικά να κάνουμε πάνω του ότι επεξεργασία θέλουμε.

3.4.1 Το περιβάλλον



Εικόνα 11: Περιβάλλον εργασίας Adobe Photoshop

Το Photoshop είναι ένα άνογα δομημένο πρόγραμμα. Περιέχει μπάρα με συντομεύσεις και ένα κεντρικό μενού περιήγησης και χρήσης για ταχύτερη πρόσβαση στις βασικές λειτουργίες του. Μεταξύ άλλων το περιβάλλον του περιέχει τον κεντρικό χώρο εργασίας, επιλογές χρωμάτων, επιλογές χρωμάτων για κείμενο και γράμματα, επιλογές για τα layers, και προεσοπίσεις. Ο χώρος εργασίας του είναι προσαρμόσιμος ανάλογα με τις ανάγκες του χρήστη καθώς οι λειτουργίες του είναι ομαδοποιημένες σε παλέτες και μπορούν να ρυθμιστούν εάν θα είναι εμφανείς ή όχι και σε ποιο σημείο θα τοποθετηθούν.

Εργαλειοθήκη

Το πρώτο εργαλείο είναι το move tool το οποίο μετακινεί επιλεγμένα κομμάτια και το marquee tool που επιτρέπει να επιλέγουμε συγκεκριμένα σημεία της φωτογραφίας. Χωρίζεται σε 4 υποκατηγορίες ανάλογα με το σχήμα που θα επιλέξουμε. Το επόμενο εργαλείο είναι το lasso tool είναι το κύριο εργαλείο που χρησιμοποιούμε στο photoshop και μας επιτρέπει να επιλέξουμε κομμάτια της φωτογραφίας με ακρίβεια. Όπως και το marquee tool έτσι και αυτό έχει 3 υποκατηγορίες. Το polygonal lasso tool επιλέγει μόνο ευθείες. Το magnetic lasso tool είναι το πιο χρησιμοποιούμενο από τα υπόλοιπα καθώς καταφέρνει να επιλέξει ακριβώς ένα συγκεκριμένο αντικείμενο στην φωτογραφία. Το quick selection tool επιτρέπει την επιλογή ενός αντικείμενου απλά και γρήγορα με ενδεχόμενο λάθους καθώς η επιλογή στην εικόνα γίνεται με απλό κλικ. Το crop tool είναι το εργαλείο με το οποίο μπορούμε να επιλέξουμε μέρος της φωτογραφίας και να αφαιρέσουμε το υπόλοιπο. Το brush tool το οποίο μας επιτρέπει να βάνουμε μια περιοχή με χρώμα. Το type tool μας επιτρέπει να βάλουμε κείμενο στην φωτογραφία μας. Το shape tool δημιουργεί σχήματα όπως ευθείες τετράγωνα κύκλους.

Οι Παλέτες

Το PhotoShop έχει κάποιες κινητές παλέτες που μπορούμε να τις κρύβουμε ή να τις εμφανίζουμε στην οθόνη, οι οποίες περιέχουν δικές τους παλέτες ή καρτέλες και επιτρέπουν την πρόσβαση σε επιλογές που έχουν σχέση με την απόδοση των εργαλείων, αλλάζουν την όψη των εικόνων. Πατώντας το πλήκτρο Tab μπορούμε να κρύβουμε και να επανεμφανίζουμε τις παλέτες, την εργαλειοθήκη και τη γραμμή κατάστασης. Μπορούμε να σύρουμε την καρτέλα μιας παλέτας από τη γραμμή τίτλου της και να την μετακινήσουμε όπου θέλουμε ή να την τοποθετήσουμε μαζί με τις καρτέλες μιας άλλης παλέτας. Για να ελευθερώσουμε χώρο στην οθόνη και να κρύψουμε τα περιεχόμενα των καρτελών αλλά όχι και τη γραμμή τίτλου τους, μπορούμε να κάνουμε κλικ στο πλήκτρο σύμπτυξης, που βρίσκεται αριστερά από το πλήκτρο κλεισίματος, ή να κάνουμε διπλό κλικ σε μια από τις καρτέλες της παλέτας. Οι παλέτες δεν κλείνουν εντελώς, αλλά ελευθερώνουν χώρο. Αν

κάνουμε κλικ στο πλήκτρο που βρίσκεται κάτω από τη γραμμή τίτλου μιας παλέτας και έχει σχήμα δεξί βελάκι, θα εμφανισθεί ένα μενού με χρήσιμες επιλογές για τη συγκεκριμένη παλέτα (καρτέλα).

Η παλέτα Color εμφανίζει κάποιους ροοστάτες (sliders), για να ορίσουμε χρώματα. Αν κάνουμε κλικ στο πλήκτρο με το δεξί βελάκι, από το πτυσσόμενο μενού μπορούμε να επιλέξουμε έναν από τους ροοστάτες Grayscale, RGB, HSB, CMYK ή Lab. Αν επιλέξουμε Color Bar θα εμφανισθεί το πλαίσιο διαλόγου Color Bar, όπου από τη λίστα Style μπορούμε να επιλέξουμε το είδος της γραμμής χρωμάτων, από RGB, CMYK, Grayscale και Current Colors. Η παλέτα διαθέτει τα δικά της εικονίδια χρώματος προσκηνίου και παρασκηνίου, κάνουμε κλικ στο εικονίδιο που θέλουμε να αλλάξουμε το χρώμα του και επιλέγουμε ένα χρώμα σύροντας τους ροοστάτες ή καταχωρώντας τιμές στα αντίστοιχα πλαίσια κειμένου ή κάνοντας κλικ στη γραμμή χρωμάτων.

3.4.2 Εικόνα

Κάθε εικόνα στο Photoshop μπορεί να δέχεται επεξεργασία για την αλλαγή των χαρακτηριστικών της όπως της ανάλυσης της, των διαστάσεων της ή των χρωμάτων της. Το πρόγραμμα μπορεί επίσης να συμπίεσει την εικόνα χρησιμοποιώντας μια ευρεία λίστα από διαδεδομένους αλγόριθμους.

Ανάλυση

Όπως σε κάθε πρόγραμμα επεξεργασίας εικόνας, έτσι και στο PhotoShop, οι εικόνες που επεξεργαζόμαστε αποτελούνται από πολύ μικρά τετραγωνάκια που ονομάζονται pixels ή εικονοστοιχεία. Το PhotoShop είναι στην ουσία ένα πολύ καλό εργαλείο καταμέτρησης και χρωματισμού των pixels. Το κάθε pixel είναι χρωματισμένο μ' ένα και μόνο ένα χρώμα και όταν τοποθετούμε όλα τα pixels μαζί σε μια διάταξη πλέγματος, τα αντιλαμβανόμαστε σαν μια κανονική φωτογραφία. Η εικόνα στην οθόνη του υπολογιστή αποτελείται και αυτή από pixels, που είναι κι αυτά τετράγωνα και τοποθετημένα σε πλέγμα. Μια τυπική οθόνη 13 ιντσών έχει κατά πλάτος 640 pixels και κατά ύψος 480 pixels. Το πλάτος του κάθε pixel είναι ίσο με 1/72 της ίντσας ή περίπου 0,35 χιλιοστά.

Στην ανάλυση καθορίζεται η ποιότητα της εικόνας κατά την εμφάνιση ή την εκτύπωση. Με τον όρο ανάλυση, εννοούμε την απόσταση των pixels μέσα στην εικόνα και την οποία την μετράμε σε ppi (pixel per inch). Πχ αν μία εικόνα έχει ανάλυση 72ppi σημαίνει ότι περιέχει 72 X 72=5184 pixels ανά τετραγωνική ίντσα. Συνεπώς όσο μεγαλύτερη ανάλυση έχει η εικόνα, τόσο περισσότερα pixels θα την απαρτίζουν με αποτέλεσμα την μεγαλύτερη ευκρίνεια και ποιότητά της. Το μειονέκτημα είναι ότι όσο μεγαλύτερη ανάλυση έχει μία εικόνα καταλαμβάνει περισσότερο χώρο στον δίσκο σε megabytes.

Οι Διορθώσεις Χρωμάτων

Οι εντολές για διόρθωση χρωμάτων βρίσκονται στο υπομενού Adjust του μενού Image και οι βασικότερες είναι οι Levels, Auto Levels και Variations. Με την εντολή Auto Levels μπορούμε να διορθώσουμε αυτόματα την αντίθεση μιας εικόνας. Το PhotoShop μετατρέπει αυτόματα το πιο ανοικτό γκριζο σε λευκό και το πιο σκούρο γκριζο σε μαύρο και “απλώνει” τα ενδιάμεσα χρώματα. Με την επιλογή Levels μπορούμε να ρυθμίσουμε τα σκούρα και τα ανοικτά χρώματα καθώς και τα μεσαία γκριζα χρώματα. Η εντολή Variations προορίζεται ειδικά για τη διόρθωση έγχρωμων εικόνων. Μπορούμε να αυξομειώσουμε την ένταση των χρωμάτων ή να εφαρμόσουμε μια χρωματική απόχρωση στην εικόνα για να αφαιρέσουμε κάποια υπερβολική δόση χρώματος.

Αλλαγή Διαστάσεων Εικόνας

Εκτός από τη γνωστή εντολή Image Size. του μενού Image, με την οποία μπορούμε να αλλάξουμε τον αριθμό των pixels που αποτελούν την εικόνα και στην ουσία το μέγεθός της στον δίσκο, υπάρχει και η εντολή Scale του υπομενού Transform του μενού Edit, με την οποία μπορούμε να αλλάξουμε το μέγεθος μιας εικόνας που εισάγαμε σε μια επιλογή. Γύρω από την εικόνα εμφανίζονται κάποιες λαβές. Αν κρατάμε πατημένο το πλήκτρο Shift ενώ σύρουμε μια από τις γωνιακές λαβές, μπορούμε να αλλάξουμε την κλίμακα της εικόνας αναλογικά.

Αποθήκευση Εικόνας

Στο PhotoShop μπορούμε να αποθηκεύσουμε μια εικόνα σε μια από πολλές μορφές αρχείων (file formats), όπου καθεμία έχει διαφορετικό σκοπό. Το PhotoShop μπορεί να επεξεργάζεται (να

ανοίγει και να αποθηκεύει) πολύ πιο γρήγορα τις εικόνες στην εγγενή τους μορφή απ' ότι σε οποιαδήποτε άλλη μορφή, αλλά είναι πολύ λίγα τα προγράμματα που υποστηρίζουν την εγγενή μορφή αρχείων του PhotoShop.

3.5 Microsoft HTML help workshop

Το Microsoft HTML help workshop είναι το πρότυπο σύστημα βοήθειας για την πλατφόρμα των Windows. Διαθέτει δημιουργία online βοήθειας για μια εφαρμογή λογισμικού ή δημιουργία περιεχόμενου για έναν τίτλο πολυμέσων ή τοποθεσία Web. Οι προγραμματιστές μπορούν να το χρησιμοποιήσουν για να τη δημιουργία θεματικής βοήθειας σε μια εφαρμογή. Μιας και δημιουργεί ένα σύστημα παροχής πληροφοριών μπορεί να χρησιμοποιηθεί για μια ευρεία γκάμα εφαρμογών συμπεριλαμβανομένων οδηγών εκμάθησης, διαδραστικών βιβλίων και κυρίως για δημιουργία βοήθειας χρήσης εφαρμογών. Παρέχει ένα εύκολο στη χρήση περιβάλλον και προσφέρει σημαντικά πλεονεκτήματα όπως τη δυνατότητα προσθήκης ενός πίνακα περιεχομένων καθώς και χρήση λέξεων κλειδιών για δυνατότητα hyperlink. Επίσης καθιστά δυνατή τη συμπίεση HTML, γραφικών και άλλων αρχείων σε ένα σχετικά μικρό σε όγκο αρχείο βοήθειας του οποίου η κατάληξη είναι .chm. Υποστηρίζει HTML, ActiveX, Java γλώσσες προγραμματισμού scripting (Jscript , VB Scripting edition) καθώς και όλα τα html format εικόνας (.jpeg, gif και png). Επίσης υποστηρίζει Unicode ώστε να μπορούν χρήστες διαφορετικών γλωσσικών εκδόσεων των windows να βλέπουν τα αρχεία βοήθειας.



Εικόνα 12: Περιβάλλον εργασίας HTML Workshop

Το Help Viewer είναι το πρόγραμμα το οποίο προβάλει τα αρχεία βοήθειας και είναι χωρισμένο σε τρία τμήματα. Οι χρήστες βλέπουν τα θέματα βοήθειας στο τμήμα με το όνομα Topic. Αριστερά του βρίσκεται το Navigation τμήμα που περιέχει ένα πίνακα περιεχομένων, ένα ευρετήριο, μια λίστα από αγαπημένα θέματα και ένα tab αναζήτησης κειμένου. Το τρίτο τμήμα περιέχει την εργαλειοθήκη που επιτρέπει στους χρήστες να εμφανίσουν ή να αποκρύψουν το τμήμα Navigation η ακόμα να προχωρήσουν στο επόμενο ή προηγούμενο θέμα. Επίσης είναι διαθέσιμα τα κουμπιά Stop, Refresh, Locate και Home. Μέσω του προγράμματος Οι συγγραφείς των αρχείων help μπορούν να προσαρμόζουν τα προεπιλεγμένα χαρακτηριστικά τους καθώς και τον τρόπο προβολής των θεμάτων τους.

3.6 Inno Setup

Για την υλοποίηση του αρχείου εγκατάστασης της εφαρμογής, χρησιμοποιήσαμε το πρόγραμμα Inno Setup. Με το Inno Setup μπορούμε να δημιουργήσουμε προγράμματα εγκατάστασης που αποτελούνται από ένα και μόνο εκτελέσιμο αρχείο. Υποστηρίζει όλες τις εκδόσεις των Windows, από τα NT 4.0 μέχρι και τα Seven. Επιπλέον προσφέρει υποστήριξη για 64-bit εφαρμογές σε

αντίστοιχα συστήματα. Προσφέρεται η δυνατότητα για disk spanning, κατά την οποία δημιουργούνται προγράμματα εγκατάστασης από περισσότερα αρχεία συγκεκριμένου μεγέθους για την τοποθέτηση σε αντίστοιχα μέσα.

Τα προγράμματα εγκατάστασης που δημιουργούνται, διαθέτουν το κλασικό περιβάλλον των Windows, το οποίο μπορούμε να αλλάξουμε όπως επιθυμούμε. Υποστηρίζεται η χρήση τυποποιημένων μεθόδων εγκατάστασης, από τις οποίες μπορεί να διαλέξει ο χρήστης αυτήν που θέλει να εκτελέσει (π.χ. Πλήρης, Ελάχιστη ή Προσαρμοσμένη). Σε κάθε μία από αυτές μπορούμε να καθορίσουμε πλήρως τις ενέργειες που θα εκτελεστούν. Όλες οι ενέργειες που εκτελούνται, μπορούν να αναιρεθούν πλήρως κατά την διάρκεια της απεγκατάστασης, μην αφήνοντας κανενός είδους υπολείμματα στον υπολογιστή που είχε γίνει η εγκατάσταση.

Τα προγράμματα εγκατάστασης συμπιέζονται ώστε να καταλαμβάνουν όσο το δυνατόν λιγότερο χώρο. Στα χαρακτηριστικά τους περιλαμβάνεται η σύγκριση έκδοσης αρχείων, η αντικατάσταση αρχείων σε χρήση, η καταχώρηση αρχείων dll/ocx, και η εγκατάσταση γραμματοσειρών. Επίσης είναι δυνατή η δημιουργία συντομεύσεων στο Start Menu και στο Desktop, καθώς και η καταχώρηση τιμών στην registry ή σε αρχεία ini. Επιπλέον μπορούν να εκτελεστούν προγράμματα πριν, κατά την διάρκεια ή μετά το τέλος της εγκατάστασης. Ακόμα προσφέρεται υποστήριξη για πολυγλωσσικά προγράμματα.

4. Αρχές υλοποίησης εφαρμογής

Σε αυτό το κεφάλαιο γίνεται αναφορά στις βασικές αρχές που διέπουν στην δημιουργία μιας ολοκληρωμένης εφαρμογής. Είναι χωρισμένο σε 2 υπό ενότητες όπου στην πρώτη γίνεται αναφορά στις αρχές σχεδίασης της διεπαφής και στην δεύτερη στις αρχές σχεδίασης βάσης δεδομένων.

4.1 Αρχές σχεδίασης διεπαφής

Τα γραφικά όλων των ειδών, είναι το κυρίαρχο στοιχείο σε μια εφαρμογή πολλαπλών μέσων. Δίνουν άλλωστε μια άμεση όψη των αντικειμένων και μεταφέρουν, στο σύνολό τους, τις εξωτερικές πληροφορίες χωρίς την ανάγκη ενδιάμεσων βοηθημάτων και επεξηγήσεων. Είναι αυτό που βλέπεις. Η χρήση των γραφικών είναι ιδανική για την ενδυνάμωση της γνώσης και παρέχει αποτελεσματική βοήθεια στην εποπτική ανάλυση ενός φαινομένου. Είναι το καλύτερο μέσο για πρότυπη παρουσίαση εννοιών. Μια εικόνα χίλιες λέξεις. Η εικόνα σε κάθε μορφή τής, ακίνητη ή κινητή, σαν φωτογραφία ή slides, σαν ζωγραφιά ή σαν σχέδιο, είναι ολοκληρωμένη αφ'εαυτής. Οι πληροφορίες που μεταφέρει είναι καθαρές και χρησιμοποιούνται άμεσα. Η σπουδαιότητα της κάθε μορφής είναι σημαντική και διαφέρει ανάλογα με το είδος και τη χρήση. Συμπληρωμένη με επεξηγηματικό κείμενο και ήχο αποκτά μια τελείως φυσική έννοια της γνώσης.

Η διεπαφή του συστήματος πρέπει να είναι σχεδιασμένη έτσι ώστε να ανταποκρίνεται στις ικανότητες, την πείρα και τις προσδοκίες των μελλοντικών χρηστών του. Οι χρήστες συχνά κρίνουν ένα σύστημα από τη διασύνδεση που παρέχει και όχι από τις λειτουργικές δυνατότητες. Ο τρόπος επικοινωνίας με το χρήστη πρέπει να είναι εύκολος και intuitive. Η πρώτη εργασία πριν το σχεδιασμό είναι η ανάλυση εργασιών του χρήστη (user task analysis) που να δείχνει ποιες εργασίες καλείται να διεκπεραιώσει. Μια κακοσχεδιασμένη διεπαφή μπορεί να κάνει τους χρήστες να υποπέσουν σε καταστροφικά λάθη. Ο κακός σχεδιασμός της διεπαφής χρήστη είναι ο λόγος για τον οποίο πολλά συστήματα λογισμικού δεν χρησιμοποιούνται ποτέ.

Στυλ αλληλεπίδρασης	Κύρια πλεονεκτήματα	Κύρια μειονεκτήματα	Παραδείγματα εφαρμογών
Άμεσος χειρισμός	Γρήγορη και διαισθητική αλληλεπίδραση. Εύκολος στην εκμάθηση	Μπορεί να είναι δύσκολος στην υλοποίηση. Κατάλληλος μόνο όπου υπάρχει οπτική μεταφορά για εργασίες και αντικείμενα.	Βιντεοπαιχνίδια, συστήματα CAD.
Επιλογή μενού	Αποφεύγει λάθη του χρήστη. Χρειάζεται λίγη πληκτρολόγηση.	Αργή για έμπειρους χρήστες. Μπορεί να γίνει περίπλοκη αν υπάρχουν πολλές επιλογές μενού.	Τα περισσότερα συστήματα γενικής χρήσης.
Συμπλήρωση φόρμας	Απλή εισαγωγή δεδομένων. Εύκολη στην εκμάθηση. Ελεγχόμενη.	Καταλαμβάνει πολύ χώρο στην οθόνη. Προκαλεί προβλήματα όπου οι επιλογές χρήστη δεν ταιριάζουν με τα πεδία της φόρμας.	Έλεγχος αποθήκης. Επεξεργασία προσωπικών στοιχείων.
Γλώσσα διαταγών	Ισχυρή και ευέλικτη	Δύσκολη στην εκμάθηση. Κακή διαχείριση λαθών.	Λειτουργικά συστήματα. Συστήματα διαταγών και ελέγχου.

Φυσική γλώσσα	Προσιτή στους περιστασιακούς χρήστες. Εύκολα επεκτάσιμη.	Απαιτεί περισσότερη πληκτρολόγηση. Τα συστήματα που αντιλαμβάνονται φυσικές γλώσσες είναι αναξιόπιστα.	Συστήματα ανάκτησης πληροφοριών.
---------------	--	--	----------------------------------

Πίνακας 12: Στυλ αλληλεπιδράσεων

4.1.1 Ανθρώπινοι παράγοντες στο σχεδιασμό διεπαφής

Προκειμένου μια εφαρμογή να είναι όσο το δυνατόν περισσότερο φιλική προς τον χρήστη, στο σχεδιασμό της πρέπει να ληφθούν υπόψη ορισμένοι ανθρώπινοι παράγοντες όπως οι :

- Περιορισμένη βραχυπρόθεσμη μνήμη .Ο άνθρωπος μπορεί να θυμάται ταυτόχρονα περίπου 7 στοιχεία πληροφοριών. Αν παρουσιάζετε στους χρήστες περισσότερες πληροφορίες, είναι πιο επιρρεπείς σε λάθη.
- Οι άνθρωποι κάνουν λάθη. Όταν ο χρήστης κάνει ένα λάθος και τα συστήματα δυσλειτουργούν, τα περιττά προειδοποιητικά μηνύματα και οι συναγερμοί αυξάνουν το άγχος του και επομένως την πιθανότητα περισσότερων λαθών.
- Κάθε άνθρωπος διαφέρει. Οι άνθρωποι έχουν ένα μεγάλο εύρος φυσικών δυνατοτήτων. Ο σχεδιαστής δεν πρέπει να σχεδιάζει με βάση μόνο τις δικές του δυνατότητες.
- Καθένας έχει διαφορετικές προτιμήσεις αλληλεπίδρασης. Σε ορισμένους αρέσει να δουλεύουν με εικόνες, σε άλλους με κείμενο.

4.1.2 Αρχές σχεδιασμού διεπαφής

Στο σχεδιασμό διεπαφής πρέπει να λαμβάνονται υπόψη οι ανάγκες, η πείρα και οι δυνατότητες των χρηστών του συστήματος. Οι αρχές του σχεδιασμού διεπαφής αποτελούν τη βάση για τη σχεδίαση κάθε διεπαφής παρότι δεν εφαρμόζονται όλες τους σε κάθε σχεδίαση.

- Εξοικείωση χρηστών. Η διασύνδεση πρέπει να χρησιμοποιεί όρους και έννοιες που προέρχονται από τις εμπειρίες των ανθρώπων που θα χρησιμοποιούν περισσότερο το σύστημα και όχι ορολογία υπολογιστών. Για παράδειγμα, ένα σύστημα που αφορά το περιβάλλον του γραφείου πρέπει να χρησιμοποιεί έννοιες όπως επιστολές, έγγραφα, φακέλους κ.λπ. αντί εννοιών όπως κατάλογοι αρχείων, αναγνωριστικά αρχείων κ.λπ.
- Συνέπεια. Η διασύνδεση πρέπει να είναι συνεπής, δηλαδή, όπου είναι δυνατό, παρόμοιες λειτουργίες θα πρέπει να ενεργοποιούνται με τον ίδιο τρόπο. Το σύστημα πρέπει να παρουσιάζει το επίπεδο συνέπειας που πρέπει. Οι εντολές/διαταγές και τα μενού του συστήματος πρέπει να έχουν την ίδια μορφή, η στίξη των εντολών πρέπει να είναι όμοια, κ.λπ.
- Ελάχιστες εκπλήξεις. Οι χρήστες δεν πρέπει να εκπλήσσονται ποτέ από τη συμπεριφορά του συστήματος. Αν μια διαταγή λειτουργεί με ένα συγκεκριμένο τρόπο, ο χρήστης πρέπει να είναι σε θέση να προβλέπει τη λειτουργία παρόμοιων διαταγών.
- Ανακαμψιμότητα. Η διασύνδεση πρέπει να περιλαμβάνει μηχανισμούς που θα επιτρέπουν στους χρήστες να ανακάμπτουν από σφάλματα. Το σύστημα πρέπει να είναι ανθεκτικό σε σφάλματα των χρηστών και να τους επιτρέπει να ανακάμπτουν από τα λάθη τους. Αυτό μπορεί να συνεπάγεται την ύπαρξη μιας λειτουργίας αναίρεσης, την επιβεβαίωση καταστροφικών ενεργειών, διαγραφή με δυνατότητα επαναφοράς των αρχείων, κ.λπ.
- Καθοδήγηση χρηστών. Η διασύνδεση πρέπει να παρέχει κατανοητές αποκρίσεις όταν συμβαίνουν λάθη, και να παρέχει στο χρήστη συναφή βοήθεια. Στους χρήστες πρέπει να παρέχεται καθοδήγηση, όπως συστήματα βοήθειας, ηλεκτρονικά εγχειρίδια κ.λπ.
- Διαφορετικότητα χρηστών. Η διασύνδεση πρέπει να παρέχει κατάλληλες λειτουργίες αλληλεπίδρασης για του διάφορους τύπους χρηστών του συστήματος. Πρέπει να υποστηρίζονται λειτουργίες αλληλεπίδρασης για διάφορους τύπους χρηστών. Για παράδειγμα, κάποιιοι χρήστες αντιμετωπίζουν προβλήματα όρασης, επομένως πρέπει να διατίθενται και μεγαλύτερα μεγέθη γραμματοσειρών κειμένου

4.1.3 Οπτικοποίηση δεδομένων

Η οπτικοποίηση δεδομένων αναφέρεται σε τεχνικές παρουσίασης μεγάλων ποσοτήτων πληροφοριών. Μπορεί να αποκαλύψει σχέσεις μεταξύ των οντοτήτων που περιλαμβάνονται στα δεδομένα, καθώς και τάσεις. Παραδείγματα οπτικοποίησης δεδομένων αποτελούν μετεωρολογικές πληροφορίες οι οποίες συγκεντρώνονται από πολλές πηγές, η κατάσταση ενός τηλεφωνικού δικτύου που απεικονίζεται γραφικά με τη μορφή ενός συνόλου συνδεδεμένων κόμβων, η κατάσταση ενός χημικού εργοστασίου που αναπαρίσταται οπτικά με ενδείξεις πιέσεων και θερμοκρασιών σε ένα σύνολο συνδεδεμένων δεξαμενών και αγωγών, το μοντέλο ενός μορίου σε τρεις διαστάσεις ακόμα και ένα σύνολο ιστοσελίδων που εμφανίζεται ως υπερβολικό δένδρο.

4.1.4 Προβολές με χρώματα

Η χρησιμοποίηση χρωμάτων έχει τη σημασία τους. Οφείλουμε να κάνουμε δοκιμές πριν καταλήξουμε στον κατάλληλο συνδυασμό. Ένα ενδιαφέρον σημείο στην παρουσίαση είναι το χρώμα γεμίματος και ο καμβάς του βάθους, δηλαδή της επιφάνειας πάνω στην οποία θα γράφει το κείμενο. Η υφή και το χρώμα του πρέπει να ταιριάζουν με την αισθητική του χρήστη, γεγονός που θα βοηθήσει στην αποδοχή του μηνύματος απ' αυτόν. Το χρώμα χρησιμοποιείται στην παρουσίαση για την επίτευξη τριών βασικών στόχων όπως προσδιορισμός, αντίθεση και επισήμανση. Προσθέτει μια επιπλέον διάσταση στη διασύνδεση και μπορεί να βοηθήσει το χρήστη να αντιλαμβάνεται πολύπλοκες δομές πληροφοριών. Μπορούν να χρησιμοποιηθούν χρώματα για την επισήμανση εξαιρετικών συμβάντων. Συνηθισμένα λάθη στη χρήση των χρωμάτων κατά το σχεδιασμό διασυνδέσεων χρήστη αποτελούν η χρήση χρωμάτων για την ερμηνεία εννοιών και η υπερβολική χρήση χρωμάτων στην οθόνη. Για να γίνει όμως αποτελεσματική η χρήση των χρωμάτων σε μια εφαρμογή πολυμέσων είναι απαραίτητο να τηρηθούν οι κανόνες χρήσης τους όπως:

- Περιορισμός του αριθμού των χρωμάτων που χρησιμοποιούνται καθώς και των συμπληρωματικών χρωμάτων όπως κόκκινο με πράσινο που δημιουργούν επιμέρους τρεμοπαίγματα στην οθόνη. Περιορίζεται το πλήθος των χρωμάτων σε δύο τρία το πολύ.
- Χρησιμοποίηση αλλαγής χρώματος ως ένδειξη αλλαγής της κατάστασης του συστήματος.
- Χρησιμοποίηση χρωματικής κωδικοποίησης για την υποστήριξη της εργασίας που προσπαθούν να διεκπεραιώσουν οι χρήστες με προσεκτικό και συνεπή τρόπο.
- Δίνουμε ιδιαίτερη προσοχή στους συνδυασμούς των χρωμάτων. Να επιλέγεται χρώμα φόντου το οποίο να είναι απαλό και φωτεινό και διαδοχικά να επιλέγεται χρώμα για τις μεγαλύτερες περιοχές και ακόλουθα για τις μικρότερες.
- Να δίνονται ζωνρά χρώματα για το παρασκήνιο. Τα πιο οικία χρώματα είναι τα χρώματα της φύσης.

4.1.5 Μηνύματα σφαλμάτων

Ο σχεδιασμός των μηνυμάτων σφαλμάτων έχει καίρια σημασία. Οι χρήστες μπορεί να απορρίψουν ένα σύστημα εξαιτίας των ποιοτικά κακών μηνυμάτων σφαλμάτων. Τα μηνύματα πρέπει να χαρακτηρίζονται από ευγένεια, ακρίβεια, συνέπεια και δημιουργική κριτική. Το γνωστικό υπόβαθρο και η πείρα του χρήστη πρέπει να αποτελούν καθοριστικό παράγοντα στο σχεδιασμό των μηνυμάτων.

Παράγοντας	Περιγραφή
Θεματικό πλαίσιο	Όπου είναι δυνατό, τα μηνύματα που παράγονται από το σύστημα θα πρέπει να αντικατοπτρίζουν το τρέχον θεματικό πλαίσιο. Το σύστημα θα πρέπει να είναι ενήμερο για το τι κάνει ο χρήστης, ώστε να μπορεί να παράγει μηνύματα σχετικά με την τρέχουσα δραστηριότητα.
Πείρα	Καθώς οι χρήστες εξοικειώνονται με ένα σύστημα, αρχίζουν να εκνευρίζονται από τα μεγάλα και υπερβολικά επεξηγηματικά μηνύματα. Ωστόσο οι αρχάριοι είναι δύσκολο να καταλάβουν τις σύντομες, λακωνικές περιγραφές ενός προβλήματος. Θα πρέπει να παρέχετε και τους δύο τύπους μηνυμάτων και να επιτρέπετε στους χρήστες να ρυθμίζουν την συνοπτικότητά τους.

Βαθμός δεξιοτήτας	Τα μηνύματα πρέπει να είναι προσαρμοσμένα στις ικανότητες των χρηστών, όπως και στην πείρα τους. Μηνύματα για διαφορετικές κατηγορίες χρηστών μπορούν να εκφράζονται με διαφορετικούς τρόπους, ανάλογα με την ορολογία που είναι οικεία στον αναγνώστη τους.
Ύφος	Τα μηνύματα πρέπει να είναι θετικά και όχι αρνητικά, δηλαδή πρέπει να χρησιμοποιούν ενεργητική και όχι παθητική φωνή στη διατύπωση. Σε καμία περίπτωση δεν πρέπει να είναι προσβλητικά ή να περιέχουν αστεϊσμούς.
Κουλτούρα	Όπου είναι δυνατό, ο σχεδιαστής των μηνυμάτων πρέπει να είναι εξοικειωμένος με την κουλτούρα της χώρας στην οποία θα διατεθεί το σύστημα. Υπάρχουν σαφείς πολιτισμικές διαφορές μεταξύ Ευρώπης, Ασίας και Αμερικής. Ένα μήνυμα κατάλληλο για μια κουλτούρα μπορεί να μην είναι αποδεκτό από μία άλλη.

Πίνακας 13: Παράγοντες για καθορισμό μηνυμάτων.

4.1.6 Γνωρίσματα χρηστικότητας

Για τον καθορισμό της χρηστικότητας μιας εφαρμογής υπάρχουν κάποια βασικά γνωρίσματα όπως :

- Ευκολία εκμάθησης. Ο χρόνος που χρειάζεται ένας νέος χρήστης για να χρησιμοποιεί το σύστημα με παραγωγικό τρόπο;
- Ταχύτητα χρήσης. Κατά πόσο ανταποκρίνεται ο χρόνος απόκρισης του συστήματος στην εργασιακή πρακτική του χρήστη.
- Ανθεκτικότητα. Αφορά το πόσο ανεκτικό είναι το σύστημα σε σφάλματα του χρήστη;
- Ανακαμψιμότητα. Αφορά το πόσο καλό είναι το σύστημα στην ανάκαμψη από λάθη του χρήστη.
- Προσαρμοστικότητα. Αφορά το πόσο στενά συνδέεται το σύστημα με ένα μόνο μοντέλο εργασίας.

4.1.7 Γραμματοσειρά

Συνήθως, για να γράψουμε ένα κείμενο χρησιμοποιούμε μια γραμματοσειρά με σταθερό ως επί το πλείστον μέγεθος και επίσης μερικές παραλλαγές της, πλάγια, έντονα υπογραμμισμένα κλπ. για να δώσουμε έμφαση. Χρησιμοποιούμε πάντα την ίδια γραμματοσειρά για την ίδια εργασία, αλλάζοντας χαρακτηριστικά της όπως χρώμα, πάχος, πλάγια για να προκαλέσουμε την προσοχή, χωρίς όμως να γίνετε υπερβολική χρήση των παραλλαγών αυτών. Τα κείμενα μεταφέρουν μηνύματα δεν είναι αινίγματα προς αποκρυπτογράφηση. Οι χαρακτήρες πρέπει να είναι αρκετά μεγάλοι και ευανάγνωστοι ανάλογα με το σκοπό που εξυπηρετούν. Αλλαγή μεγέθους γίνεται μόνο σε εξαιρετικές περιπτώσεις, επικεφαλίδες κλπ. και ανάλογα με την σπουδαιότητα του μηνύματος.

4.1.8 Πλοήγηση

Η χαρτογράφηση της δομής του project είναι ένα από τα αρχικά στάδια στη φάση του χρονοδιαγράμματος και μάλιστα από τα βασικότερα, αφού αφορά την απρόσκοπτη και χωρίς εκπλήξεις λειτουργία του project. Ο χάρτης πλοήγησης, συνήθως σε σχήμα δέντρου, περιγράφει τους κόμβους και τις συνδέσεις ανάμεσα στα διάφορα στάδια της παρουσίασης των περιεχομένων και βοηθάει στην καλύτερη οργάνωση των περιεχομένων και των μηνυμάτων. Μας προμηθεύει άλλωστε με έναν πίνακα περιεχομένων, ανάλογο με τις κάρτες λογικής ροής, του διαλογικού inter-faces. Αυτός περιγράφει τα αντικείμενα πολυμέσων και δείχνει τι συμβαίνει όταν ο χρήστης ενεργεί σε ένα από αυτά. Οι βασικές μορφές πλοήγησης είναι:

- Γραμμική. Διαδοχική μεταφορά από το πρώτο στάδιο στο επόμενο ή από την μια πληροφορία στη γειτονική.
- Ιεραρχημένη. Πλοήγηση διά μέσου των κλάδων μιας δομής σε σχήμα δέντρου που ακολουθεί τη λογική ανάπτυξη των περιεχομένων.

- Μη γραμμική. Σε αυτή είναι δυνατή η τυχαία προσπέλαση των περιεχομένων του project.
- Σύνθετη. Σε αυτήν είναι δυνατόν σε τμήματά της να χρησιμοποιηθούν διαφορετικά είδη πλοήγησης.

Η μέθοδος που προβλέπει τον τρόπο πλοήγησης αποτελεί μέρος του user interface. Η επιτυχία του εξαρτάται όχι μόνον από το γενικό σχεδιασμό και το γραφικό καλλωπισμό του μα και από πολλές μηχανικές λεπτομέρειες, όπως π.χ. το μέγεθος κ την θέση των κουμπιών, την ύπαρξη <<hot spots>>. σχετικών με τη δραστηριότητα του χρήστη κλπ σε συνδυασμό με τα συνηθισμένα κουμπιά του γραφικού περιβάλλοντος και των pull down menu. Τα περισσότερα είναι μη γραμμικά, γιατί πραγματικά είναι σπουδαίο να καλλιεργείται στο χρήστη η ιδέα της ελεύθερης και απρόσκοπτης επιλογής που δίνει την δυνατότητα να μετακινείται από ένα σημείο του project, ή μιας δραστηριότητας, σε ένα άλλο.

Πολλά συστήματα πολυμέσων επιτρέπουν τη δημιουργία hot spot and buttons, τα οποία ελέγχουν την εμφάνιση σε κάθε σημείο της οθόνης πληροφοριών διαφόρων μορφών, σχετικών με το αντικείμενο μελέτης. Οι δυνατότητές τους αυξάνουν όταν η χρήση τους είναι intuitive. Ο αριθμός τους δε πρέπει να διατηρείται σε χαμηλά επίπεδα, με αυτό τον τρόπο γίνεται ευκολότερη η χρήση τους. Η ύπαρξή τους σε κάθε οθόνη είναι απαραίτητη, τουλάχιστον τα δυο-τρία βασικά όπως: διαγραφή λειτουργίας, επιστροφή κλπ. Τρεις είναι οι βασικές κατηγορίες τους: κειμένου, υπογραμμισμένη έγχρωμη λέξη γραφικών, τμήμα εικόνας ή χάρτη και εικονιδίων που συμβολίζουν χαρακτηριστικές λειτουργίες. Το κουμπί πατιέται μία φορά και λειτουργεί αλλάζοντας χαρακτηριστικά, π.χ. στα win γίνεται πιο ανοικτό και η σκιά μεταφέρεται εσωτερικά δίνοντας την εντύπωση του πατημένου κουμπιού ή με διπλό κλικ.

4.2 Αρχές σχεδίασης Βάσεων δεδομένων

Μια σωστά σχεδιασμένη βάση δεδομένων παρέχει πρόσβαση σε ενημερωμένες, ακριβείς πληροφορίες. Επειδή η σωστή σχεδίαση είναι ουσιαστικής σημασίας για την επίτευξη των στόχων κατά την δημιουργία μιας βάση δεδομένων, η επένδυση του χρόνου που απαιτείται για την εκμάθηση των αρχών της καλής σχεδίασης είναι σημαντική καθώς το αποτέλεσμα μπορεί να είναι μια βάση δεδομένων που καλύπτει τις ανάγκες μας και μπορεί να εξυπηρετήσει με ευκολία τις αλλαγές. Για αυτό τον σκοπό παρέχονται κατευθυντήριες γραμμές για το σχεδιασμό μιας βάσης δεδομένων ώστε να γίνει διάκριση για το ποιες πληροφορίες απαιτούνται, για τον διαχωρισμό τους στους κατάλληλους πίνακες και στήλες και πώς αυτοί οι πίνακες συσχετίζονται μεταξύ τους.

Η Microsoft Office Access οργανώνει τις πληροφορίες σε πίνακες οι οποίοι είναι λίστες με γραμμές και στήλες, που θυμίζουν σημειωματάριο λογιστή ή φύλλο εργασίας του Microsoft Office Excel. Μια απλή βάση δεδομένων, ίσως περιέχει έναν μόνο πίνακα. Οι περισσότερες βάσεις δεδομένων περιέχουν περισσότερους από έναν. Για παράδειγμα μπορούμε να έχουμε έναν πίνακα που αποθηκεύει πληροφορίες για προϊόντα, έναν άλλον που αποθηκεύει πληροφορίες για παραγγελίες και έναν άλλον με πληροφορίες για πελάτες.

Υπάρχουν ορισμένες αρχές που καθοδηγούν τη διαδικασία της σχεδίασης βάσης δεδομένων. Η πρώτη αρχή είναι ότι οι διπλότυπες πληροφορίες (που επίσης ονομάζονται πλεονάζοντα δεδομένα) είναι κακές, διότι σπαταλούν χώρο και αυξάνουν την πιθανότητα σφαλμάτων και ασυνεπειών. Η δεύτερη αρχή είναι ότι η ορθότητα και η πληρότητα των πληροφοριών είναι σημαντικές. Εάν η βάση δεδομένων περιέχει λανθασμένες πληροφορίες, οι εκθέσεις που αντλούν πληροφορίες από τη βάση δεδομένων επίσης θα περιέχουν λανθασμένες πληροφορίες. Ως αποτέλεσμα, οι όποιες αποφάσεις βασίζονται σε αυτές τις εκθέσεις θα είναι βασισμένες σε παραπληροφόρηση. Επομένως μια καλή σχεδίαση βάσης δεδομένων είναι αυτή που:

- Χωρίζει τις πληροφορίες σε πίνακες που βασίζονται σε θέματα, για να ελαττωθούν τα πλεονάζοντα δεδομένα.
- Παρέχει στην Access τις πληροφορίες που απαιτούνται για να συνδεθούν οι πληροφορίες στους πίνακες, σύμφωνα με τις ανάγκες.
- Βοηθά την υποστήριξη και εξασφαλίζει την ακρίβεια και την ακεραιότητα των πληροφοριών.
- Εξυπηρετεί την επεξεργασία των δεδομένων και των αναγκών έκθεσης.

4.2.1 Η διαδικασία σχεδίασης

Η διαδικασία σχεδίασης αποτελείται από τα ακόλουθα βήματα:

- Καθορισμός του σκοπού που θα εξυπηρετεί η βάση δεδομένων. Αυτό το βήμα βοηθά στην προετοιμασία για τα υπόλοιπα βήματα.
- Εύρεση και οργάνωση των απαιτούμενων πληροφοριών. Συγκέντρωση όλων των τύπων πληροφοριών που χρειάζεται να εγγραφούν στη βάση δεδομένων, όπως όνομα ασθενούς και πλήθος εξετάσεων
- Χωρισμός των πληροφοριών σε πίνακες. Διαχωρισμός των στοιχείων πληροφοριών σε κύριες οντότητες ή θέματα, όπως "Βασικά στοιχεία ασθενούς" ή "Κλινική εξέταση" και έπειτα κάθε θέμα γίνεται πίνακας.
- Μετατροπή των στοιχείων πληροφοριών σε στήλες. Καθορισμός των πληροφοριών που είναι να αποθηκευτούν σε κάθε πίνακα. Κάθε στοιχείο μετατρέπεται σε πεδίο και εμφανίζεται στον πίνακα ως στήλη. Για παράδειγμα, ο πίνακας "Βασικά στοιχεία ασθενούς" μπορεί να περιλαμβάνει πεδία όπως "Όνοματεπώνυμο" και "Ηλικία".
- Καθορισμός πρωτεύοντων κλειδιών. Επιλογή του πρωτεύοντος κλειδιού κάθε πίνακα. Το πρωτεύον κλειδί είναι μια στήλη που χρησιμοποιείται για τον μοναδικό προσδιορισμό κάθε γραμμής. Για παράδειγμα, "ID ασθενούς" ή "ID εξέτασης".
- Δημιουργία σχέσεων πινάκων. Εξέταση κάθε πίνακα και καθορισμός του τρόπου με τον οποίο τα δεδομένα ενός πίνακα σχετίζονται με τα δεδομένα των άλλων πινάκων. Πρόσθεση πεδίων σε πίνακες ή δημιουργία νέων πινάκων για τη διευκρίνιση των απαιτούμενων σχέσεων.
- Βελτίωση της σχεδίασης. Ανάλυση της σχεδίασης για σφάλματα. Δημιουργία πινάκων και πρόσθεση μερικών εγγραφών ως δείγμα δεδομένων. Επαλήθευση των αποτελεσμάτων που ζητούνται από τους πίνακες ώστε να γίνουν οι απαραίτητες προσαρμογές στη σχεδίαση.
- Εφαρμογή των κανόνων κανονικοποίησης. Εφαρμογή των κανόνων κανονικοποίησης δεδομένων για την εξέταση της σωστής δομής των πινάκων ώστε να γίνουν σε αυτούς οι απαραίτητες προσαρμογές.
- Καθορισμός του σκοπού που θα εξυπηρετεί η βάση δεδομένων. Είναι καλή ιδέα η καταγραφή του σκοπού που θα εξυπηρετεί η βάση δεδομένων, του σκοπού που θα εξυπηρετεί, του τρόπου κατά του οποίου θα χρησιμοποιηθεί, καθώς και το ποιος θα τη χρησιμοποιεί. Για παράδειγμα για μια μικρή βάση δεδομένων μιας μικρής οικογενειακής επιχείρησης η καταγραφή θα ήταν "Η βάση δεδομένων πελατών διατηρεί μια λίστα με πληροφορίες πελατών, για να εξυπηρετεί ταχυδρομικές αποστολές και εκθέσεις". Εάν η βάση δεδομένων είναι πιο σύνθετη ή χρησιμοποιείται από πολλά άτομα, όπως συμβαίνει συχνά σε μια εταιρεία, η καταγραφή του σκοπού της θα μπορούσε κάλλιστα να πιάνει μια παράγραφο ή και περισσότερο, και θα πρέπει να περιλαμβάνει το πότε και το πώς θα χρησιμοποιεί ο καθένας τη βάση δεδομένων. Ο στόχος είναι να υπάρχει μια καλά ανεπτυγμένη δήλωση αποστολής, στην οποία να γίνεται αναφορά σε όλη τη διάρκεια της διαδικασίας της σχεδίασης. Μια τέτοια δήλωση σας βοηθά στην εστίαση των στόχων κατά τη λήψη αποφάσεων.

Εύρεση και οργάνωση των απαιτούμενων πληροφοριών

Για την εύρεση και οργάνωση των απαιτούμενων πληροφοριών, η αρχή γίνεται από τις ήδη υπάρχουσες. Για παράδειγμα, μπορεί να γίνεται η καταγραφή των παραγγελιών αγοράς σε ένα καθολικό ή να διατηρούνται οι πληροφορίες πελατών σε έγγραφες φόρμες, μέσα σε ένα ερμάριο αρχείων. Οφείλει να γίνει η συγκέντρωση όλων αυτών των εγγράφων και η καταγραφή κάθε τύπου πληροφοριών που εμφανίζεται (π.χ. κάθε πλαίσιο που συμπληρώνεται σε μια φόρμα). Έπειτα γίνεται προσδιορισμός των πληροφοριών και των πλαισίων προς συμπλήρωση που θα συμπεριληφθούν στη φόρμα. Για παράδειγμα για μια λίστα πελατών σε κάρτες ευρετηρίου η εξέταση αυτών των καρτών ίσως δείξει ότι κάθε κάρτα περιέχει το όνομα, τη διεύθυνση, την πόλη, την περιφέρεια, τον ταχυδρομικό κώδικα και το τηλέφωνο ενός πελάτη. Καθένα από αυτά τα στοιχεία αντιπροσωπεύει μια πιθανή στήλη σε έναν πίνακα. Επιπλέον η σύνταξη ενός πρωτότυπου για κάθε έκθεση ή κατάλογο εξόδου μπορεί να βοηθήσει στο ποια στοιχεία θα χρειαστούν για την παραγωγή της έκθεσης.

Κάθε πληροφορία πρέπει να διασπάται στα μικρότερα χρήσιμα μέρη της. Στην περίπτωση που για ένα ονοματεπώνυμο το επώνυμο είναι να γίνει άμεσα διαθέσιμο, θα γίνει διάσπαση του

ονοματεπωνύμου σε δύο μέρη, Όνομα και Επώνυμο. Για παράδειγμα για την ταξινόμηση μιας έκθεσης κατά επώνυμο, είναι χρήσιμο η αποθήκευση του επώνυμου του πελάτη να έχει γίνει χωριστά. Γενικά, για την ταξινόμηση, την αναζήτηση, τον υπολογισμό και την έκθεση βάσει ενός στοιχείου μιας πληροφορίας, η τοποθέτηση αυτού του στοιχείου πρέπει να γίνει σε ξεχωριστό πεδίο. Απαραίτητος είναι επίσης ο καθορισμός των ερωτήσεων στις οποίες θα απαντά η βάση δεδομένων. Για παράδειγμα ο αριθμός των πωλήσεων του κύριου προϊόντος για κάθε μήνα ή η τοποθεσία με τις υψηλότερες πωλήσεις; Η πρόβλεψη αυτών των ερωτήσεων βοηθά στην εστίαση των πρόσθετων στοιχείων που είναι να καταγραφούν.

Χωρισμός των πληροφοριών σε πίνακες

Για το διαχωρισμό των πληροφοριών σε πίνακες, επιλέγονται οι κύριες οντότητες ή θέματα. Οι κύριες οντότητες μπορεί να είναι τα προϊόντα, οι προμηθευτές, οι πελάτες και οι παραγγελίες, Επομένως η αρχή γίνεται με τη δημιουργία αυτών των τεσσάρων πινάκων όπου ένας θα περιλαμβάνει τα δεδομένα σχετικά με τα προϊόντα, ένας τα δεδομένα σχετικά με τους προμηθευτές, ένας τα δεδομένα σχετικά με τους πελάτες και ένας τα δεδομένα σχετικά με τις παραγγελίες. Η τοποθέτηση όλων των στοιχείων σε ένα πίνακα δεν είναι τόσο λειτουργική καθώς κάθε γραμμή περιέχει πληροφορίες και για το προϊόν και για τον προμηθευτή του. Επειδή υπάρχουν πολλά προϊόντα από τον ίδιο προμηθευτή, οι πληροφορίες για το όνομα και τη διεύθυνση του προμηθευτή θα πρέπει να επαναλαμβάνονται πολλές φορές και αυτό σπαταλά το χώρο του δίσκου. Η εγγραφή των πληροφοριών του προμηθευτή μόνο μία φορά σε έναν ξεχωριστό πίνακα "Προμηθευτές", και στη συνέχεια η σύνδεση αυτού του πίνακα στον πίνακα "Προϊόντα", αποτελεί μια πολύ καλύτερη λύση.

Ένα δεύτερο πρόβλημα με αυτή τη σχεδίαση παρουσιάζεται όταν χρειάζεται η τροποποίηση των πληροφοριών του προμηθευτή. Επειδή αυτές οι πληροφορίες εμφανίζονται πολλές φορές, μπορεί συμπτωματικά η αλλαγή της μιας θέσης να μην ανανεώσει μια άλλη που απαιτείται. Κατά τη σχεδίαση της βάσης, η εγγραφή κάθε δεδομένου είναι καλό να γίνεται μόνο μία φορά. Εάν επαναλαμβάνεται η ίδια πληροφορία σε περισσότερες από μία θέσεις, όπως η διεύθυνση ενός συγκεκριμένου προμηθευτή, η τοποθέτηση της πληροφορίας είναι καλό να γίνει σε ξεχωριστό πίνακα.

Πρόβλημα που προκύπτει με αυτήν την σχεδίαση υπάρχει επίσης κατά την διαγραφή ενός μόνο πεδίου από μία εγγραφή. Σε αυτή την περίπτωση είναι αδύνατη η διαγραφή χωρίς την απώλεια των πληροφοριών ολόκληρης της καταχώρησης καθώς κάθε εγγραφή περιέχει δεδομένα σχετικά με το πεδίο προς διαγραφή, μαζί με δεδομένα σχετικά με άλλα πεδία. Γι' αυτό το λόγο πρέπει να γίνει ο διαχωρισμός των δεδομένων χωρίζοντας τον πίνακα σε περισσότερους πίνακες. Στις στήλες κάθε πίνακα πρέπει να αποθηκεύονται δεδομένα που είναι σχετικά μόνο με το θέμα που αντιπροσωπεύει. Για παράδειγμα, σε έναν πίνακα με "Προϊόντα" πρέπει να αποθηκεύονται μόνο δεδομένα σχετικά με τα προϊόντα ενώ η διεύθυνση του προμηθευτή είναι ένα δεδομένο σχετικό με τον προμηθευτή και ανήκει στον πίνακα "Προμηθευτές".

Μετατροπή των στοιχείων πληροφοριών σε στήλες

Ο καθορισμός των στηλών ενός πίνακα εξαρτάται από ποιες πληροφορίες χρειάζονται για το θέμα που καταγράφεται στον πίνακα. Για παράδειγμα, για έναν πίνακα "Πελάτες", οι στήλες "Όνομα", "Διεύθυνση", "Πόλη" αποτελούν μια καλή αρχική λίστα. Κάθε εγγραφή του πίνακα περιέχει το ίδιο σύνολο στηλών και δεδομένα σχετικά με έναν πελάτη. Εάν απαιτείται φιλτράρισμα για την εκτέλεση αναζητήσεων μπορούν να βελτιωθούν οι υπάρχουσες στήλες χωρίζοντας τις σε επιπλέον στήλες. Για παράδειγμα, η αποθήκευση του ονόματος του πελάτη ως δύο ξεχωριστές στήλες για όνομα και επώνυμο, ώστε να γίνει η ταξινόμηση, η αναζήτηση και η καταχώρηση σε ευρετήριο μόνο σε αυτές τις στήλες.

Να μην περιλαμβάνονται υπολογισμένα δεδομένα

Στις περισσότερες περιπτώσεις, δεν συνίσταται η αποθήκευση στους πίνακες των αποτελεσμάτων υπολογισμών. Αντίθετα όλοι οι υπολογισμοί μπορούν να γίνουν στην Access. Για παράδειγμα, έστω ότι υπάρχει μια έκθεση "Προϊόντα κατά παραγγελία", στην οποία εμφανίζεται το μερικό άθροισμα των μονάδων κατά παραγγελία, για κάθε κατηγορία προϊόντων στη βάση δεδομένων. Ωστόσο, δεν υπάρχει στήλη για το μερικό άθροισμα "Μονάδες κατά παραγγελία" σε κανέναν πίνακα. Αντί για αυτό, ο πίνακας "Προϊόντα" περιλαμβάνει μια στήλη "Μονάδες κατά παραγγελία", που αποθηκεύει τις μονάδες κατά παραγγελία για κάθε προϊόν. Με τη χρήση αυτών των δεδομένων, η

Access υπολογίζει το μερικό άθροισμα κάθε φορά που εκτυπώνεται η έκθεση. Το ίδιο το μερικό άθροισμα δεν πρέπει να αποθηκεύεται σε πίνακα.

Καθορισμός πρωτευόντων κλειδιών

Κάθε πίνακας πρέπει να περιλαμβάνει μια στήλη ή σύνολο στηλών, που να προσδιορίζει μοναδικά κάθε γραμμή που είναι αποθηκευμένη στον πίνακα. Αυτός είναι συχνά ένας μοναδικός αριθμός αναγνώρισης, όπως ένας κωδικός υπαλλήλου ή ένας σειριακός αριθμός. Στην ορολογία βάσης δεδομένων, αυτή η πληροφορία ονομάζεται πρωτεύον κλειδί του πίνακα. Στην Access χρησιμοποιούνται πεδία πρωτευόντων κλειδιών για να συσχετίζουν γρήγορα δεδομένα από πολλαπλούς πίνακες.

Το πρωτεύον κλειδί ενός πίνακα συνήθως είναι ένα μοναδικό αναγνωριστικό όπως ένας αριθμός προϊόντος που προσδιορίζει μοναδικά κάθε προϊόν. Διπλότυπες τιμές σε ένα πρωτεύον κλειδί δεν γίνεται να υπάρχουν. Η χρήση ονομάτων για παράδειγμα, ως πρωτεύον κλειδί είναι λανθασμένη γιατί δεν είναι μοναδικά. Είναι πολύ πιθανό να υπάρχουν δύο άτομα με το ίδιο όνομα στον ίδιο πίνακα. Ένα πρωτεύον κλειδί πρέπει πάντα να έχει μια τιμή. Εάν η τιμή μιας στήλης μπορεί κάποια στιγμή να μετατραπεί σε μη ανατεθειμένη ή άγνωστη (μια τιμή που λείπει), δεν είναι δυνατόν να χρησιμοποιηθεί ως στοιχείο σε ένα πρωτεύον κλειδί. Η τιμή ενός πρωτεύοντος κλειδιού δεν θα πρέπει να μεταβάλλεται. Σε μια βάση δεδομένων που χρησιμοποιεί περισσότερους από έναν πίνακες, το πρωτεύον κλειδί ενός πίνακα μπορεί να χρησιμοποιηθεί ως αναφορά σε άλλους πίνακες. Εάν το πρωτεύον κλειδί αλλάξει, η αλλαγή πρέπει να εφαρμοστεί επίσης οπουδήποτε αναφέρεται αυτό το κλειδί. Η χρήση ενός πρωτεύοντος κλειδιού που δεν αλλάζει, μειώνει την πιθανότητα να βρεθεί το πρωτεύον κλειδί εκτός συγχρονισμού με τους άλλους πίνακες που το αναφέρουν.

Συχνά, ως πρωτεύον κλειδί χρησιμοποιείται ένας αυθαίρετος μοναδικός αριθμός. Εάν δεν υπάρχει μια στήλη ή ένα σύνολο στηλών που θα μπορούσαν να αποτελέσουν ένα καλό πρωτεύον κλειδί, τότε μπορεί να χρησιμοποιηθεί μια στήλη που έχει τύπο δεδομένων "Αυτόματη αρίθμηση". Κατά την "Αυτόματη αρίθμηση", από την Access αντιστοιχίζεται αυτόματα μια τιμή. Ένα τέτοιο αναγνωριστικό δεν περιέχει δεδομένα. Δεν περιέχει πληροφορίες δεδομένων που περιγράφουν τη γραμμή την οποία αντιπροσωπεύει. Τα αναγνωριστικά που είναι κενά δεδομένων είναι ιδανικά για να χρησιμοποιηθούν ως πρωτεύοντα κλειδιά, επειδή δεν αλλάζουν. Ένα πρωτεύον κλειδί που περιέχει δεδομένα σχετικά με μια γραμμή είναι πιο πιθανό να αλλάξει, επειδή η ίδια η πληροφορία δεδομένων ενδέχεται να αλλάξει.

Δημιουργία σχέσεων πινάκων

Αποτελεί ένα μέσο συγκέντρωσης των πληροφοριών με απεικονίσεις που έχουν σημασία. Η Access είναι ένα συσχετιστικό σύστημα διαχείρισης βάσης δεδομένων. Σε μια συσχετιστική βάση δεδομένων, οι πληροφορίες χωρίζονται σε διαφορετικούς πίνακες που βασίζονται σε θέματα. Κατόπιν, χρησιμοποιούνται οι σχέσεις πινάκων για τη συγκέντρωση των πληροφοριών. Οι σχέσεις πινάκων μπορούν να είναι ένα προς πολλά, ένα προς ένα ή πολλά προς πολλά. Ο προσδιορισμός της σχέσης μεταξύ πινάκων βοηθά στο να εξασφαλίζεται η σωστή χρήση πινάκων και στηλών. Όταν υπάρχει μια σχέση ένα-προς-ένα ή ένα-προς-πολλά, οι πίνακες που εμπλέκονται χρειάζεται να έχουν μία ή περισσότερες κοινές στήλες. Όταν υπάρχει μια σχέση πολλά-προς-πολλά, χρειάζεται ένας τρίτος πίνακας για να απεικονίσει τη σχέση.

Βελτίωση της σχεδίασης

Κατά την βελτίωση της σχεδίασης γίνεται επανεξέταση των πινάκων, των πεδίων και των σχέσεων που δημιουργήθηκαν για τον εντοπισμό πιθανών προβλημάτων. Για παράδειγμα, ίσως χρειαστεί να προστεθεί μια επιπλέον στήλη που παραλείφθηκε κατά τη φάση της σχεδίασης, ή ο διαχωρισμός ενός πίνακα σε δύο πίνακες, για να κατάργηση τυχόν διπλοτυπίας. Επιπλέον γίνεται επαλήθευση των δεδομένων που είναι να εμφανιστούν. Έπειτα γίνεται εξάλειψη των περιττών διπλότυπων δεδομένων. Συνοψίζοντας οι ενέργειες που πρέπει να γίνουν για βελτίωση της σχεδίασης είναι οι εξής:

- Επιπλέον στήλες που είναι να προστεθούν.
- Περιττές στήλες για αφαίρεση.
- Εξάλειψη διπλότυπων πληροφοριών.

- Επανασχεδίαση του πίνακα με το μικρότερο δυνατό αριθμό πεδίων.
- Διάσπαση απαραίτητων στοιχείων πληροφορίας σε περισσότερα μέρη.
- Επανατοποθέτηση λανθασμένα τοποθετημένων στηλών στους κατάλληλους θεματικούς πίνακες.
- Επανεξέταση των σχέσεων των πινάκων ώστε όλοι οι πίνακες να συνδέονται κατάλληλα.
- Εντοπισμός των επαναλαμβανόμενων ομάδων.

Εφαρμογή των κανόνων κανονικοποίησης

Η εφαρμογή των κανόνων κανονικοποίησης δεδομένων είναι το επόμενο βήμα στη σχεδίαση. Η χρήση αυτών των κανόνων βοηθά στον έλεγχο της σωστής δομής των πινάκων. Η διαδικασία εφαρμογής αυτών των κανόνων στη σχεδίαση της βάσης δεδομένων ονομάζεται κανονικοποίηση της βάσης δεδομένων ή απλώς κανονικοποίηση. Η κανονικοποίηση είναι ιδιαίτερα χρήσιμη μετά την απεικόνιση όλων των στοιχείων πληροφορίας και την κατάληξη σε μια προκαταρκτική σχεδίαση. Ο στόχος της είναι να βοηθήσει στην επιβεβαίωση ότι τα στοιχεία πληροφορίας είναι διαχωρισμένα στους κατάλληλους πίνακες. Η κανονικοποίηση δεν εξασφαλίζει ότι υπάρχουν εξαρχής τα σωστά δεδομένα. Η εφαρμογή των κανόνων γίνεται διαδοχικά, επιβεβαιώνοντας σε κάθε βήμα ότι η σχεδίαση προσεγγίζει σε μία που περιέχει αυτό που ονομάζεται "κανονικές φόρμες". Πέντε κανονικές φόρμες είναι ευρέως αποδεκτές αλλά οι τρεις πρώτες είναι αυτές που απαιτούνται για την πλειοψηφία της σχεδίασης βάσεων δεδομένων.

- Η **πρώτη κανονική φόρμα** δηλώνει ότι σε κάθε διασταύρωση γραμμών και στηλών στον πίνακα, υπάρχει μία τιμή και ποτέ μια λίστα από τιμές
- Η **δεύτερη κανονική φόρμα** απαιτεί από κάθε στήλη χωρίς κλειδί να είναι πλήρως εξαρτημένη από ολόκληρο το πρωτεύον κλειδί και όχι απλώς από ένα μέρος του. Δηλαδή ένα πεδίο να μην εξαρτάται από πρωτεύον κλειδί το οποίο δεν σχετίζεται μαζί του θεματικά.
- Η **τρίτη κανονική φόρμα** απαιτεί από κάθε στήλη χωρίς κλειδί όχι μόνον να εξαρτάται από ολόκληρο το πρωτεύον κλειδί, αλλά να είναι ανεξάρτητη από όλες τις άλλες. Με άλλα λόγια, κάθε στήλη χωρίς κλειδί πρέπει να εξαρτάται αποκλειστικά και μόνο από το πρωτεύον κλειδί.

5. Ανάπτυξη εφαρμογής

Τη διαδικασία της ανάπτυξης της εφαρμογής τη διασπάσαμε σε τέσσερις ενότητες, στην προετοιμασία, στην ανάπτυξη interface, στην ανάπτυξη της βάσης δεδομένων και στην υλοποίηση κώδικα. Στην ενότητα προετοιμασία περιγράφουμε όλες τις ενέργειες που έγιναν ώστε να ξεκινήσει η υλοποίηση της εφαρμογής. Στην ενότητα δημιουργία interface περιγράφουμε τη διεπαφή που υλοποιήσαμε ώστε το πρόγραμμα να είναι όσο το δυνατόν πιο λειτουργικό και χρηστικό. Στην ενότητα ανάπτυξη της βάσης δεδομένων περιγράφουμε την βάση δεδομένων ασθενών και εξετάσεων που υλοποιήσαμε. Στην ενότητα υλοποίηση κώδικα γίνεται περιγραφή των μεθόδων, συναρτήσεων που υλοποιήσαμε για την ορθή εκτέλεση του προγράμματος και των λειτουργιών που παρέχουμε.

5.1 Προετοιμασία

Συγκεντρώθηκαν όλες οι απαιτήσεις του χρήστη όσον αφορά τα δεδομένα που θα καταχωρηθούν καθώς και οι προτιμήσεις του για την γραφική αναπαράστασή τους. Ο ρόλος του χρήστη σε αυτήν την περίπτωση είναι ο υπεύθυνος ιατρός του ογκολογικού τμήματος καρκίνου του μαστού καθώς και οι νοσηλεύτριες που θα έχουν πρόσβαση στο σύστημα. Στο πλαίσιο αυτό μας δόθηκαν έξι καρτέλες στοιχείων ασθενούς και εξετάσεων με τίτλους **παράγοντες κινδύνου, κλινική εξέταση, παρούσα νόσος/ατομικό αναμνηστικό, παρακολούθηση ασθενών με καρκίνο του μαστού, διαγνωστικός έλεγχος και σταδιοποίηση καρκίνου**. Σύμφωνα με τον χρήστη κάθε ασθενής κάνει κάθε εξέταση τουλάχιστον από μία φορά πλην των εξετάσεων διαγνωστικός έλεγχος, κλινική εξέταση και παρακολούθηση ασθενών με καρκίνο του μαστού που μπορεί να κάνει πολλαπλές.

Εικόνα 13: Αρχική μορφή στοιχείων

Έγινε καταγραφή όλου του υπολογιστικού εξοπλισμού του τμήματος και του λογισμικού και των προγραμμάτων που είναι εγκατεστημένα. Κατά την καταγραφή του λογισμικού βρέθηκαν εγκατεστημένα Windows Xp με service pack 3, το Microsoft Office 2003 και το NET Framework 3.5. Ο υπολογιστικός εξοπλισμός του τμήματος αποτελείται από επεξεργαστές διπύρηνους συχνότητας 2.3 gigahertz, σκληρούς δίσκους χωρητικότητας 500 gigabyte, μνήμες RAM 4 gigabyte, σύγχρονες κάρτες γραφικών και σύγχρονες οθόνες 20 ιντσών. Το συμπέρασμα που βγήκε από την ανάλυση του εξοπλισμού είναι ότι το τμήμα διαθέτει τον κατάλληλο εξοπλισμό για να υποστηρίξει την λειτουργία απαιτητικών προγραμμάτων και υψηλών αναλύσεων απεικόνισης.

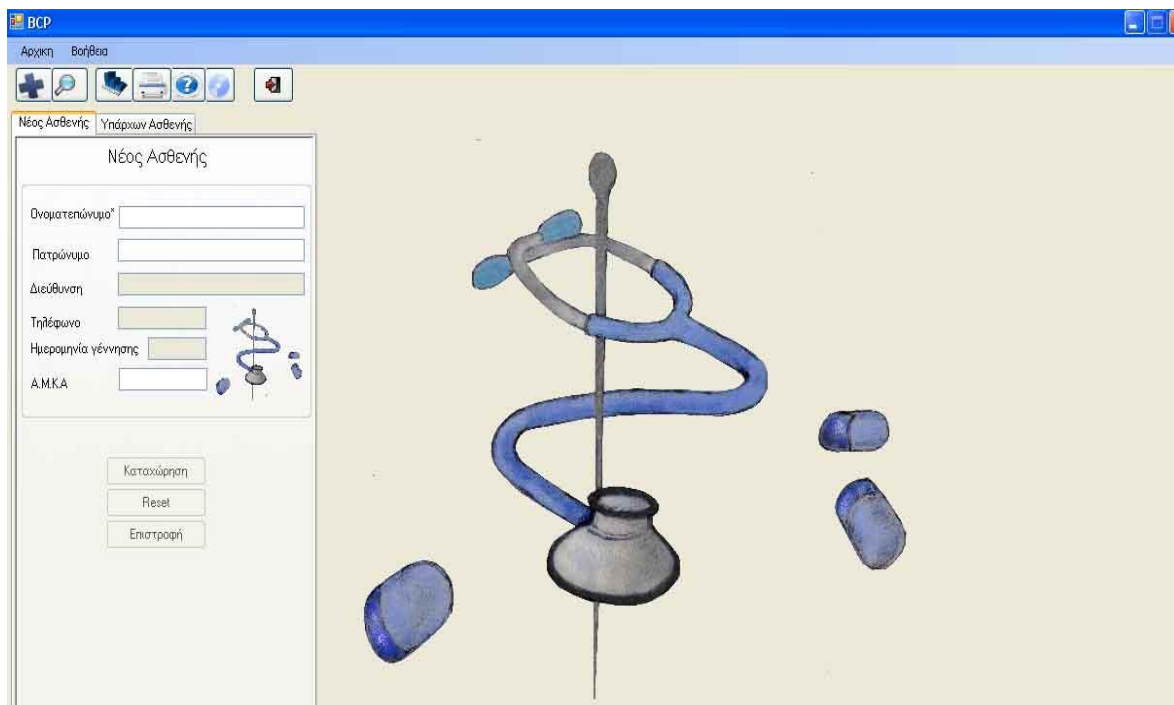
Εγκαταστάθηκαν στο σύστημα που αναπτύξαμε την εφαρμογή τα προγράμματα Microsoft visual Studio 2008 με επιπλέον plug-ins τα Crystal Reports και ms charts, τα προγράμματα Microsoft help center, Adobe Photoshop και τα απαιτούμενα συστατικά όπως Net Framework 4. Επίσης για την δημιουργία της βάσης εγκαταστάθηκε το πακέτο Microsoft office που περιλαμβάνει την Microsoft Access. Η εφαρμογή υλοποιήθηκε εξολοκλήρου σε λειτουργικό Windows Xp έκδοσης service pack 3.

5.2 Δημιουργία interface

Το Interface προσαρμόστηκε για αναλύσεις 1024 x 768 και άνω οπότε απαιτεί οθόνη και κάρτα γραφικών που να υποστηρίζει τέτοιες αναλύσεις για την καλύτερη δυνατή προβολή του. Ο Σχεδιασμός του έγινε με γνώμονα τις αρχές σχεδίασης διεπαφής και τις απαιτήσεις του τελικού χρήστη. Για τον λόγο αυτό επιλέχθηκε το πρότυπο περιβάλλον λειτουργίας windows εφαρμογών καθώς είναι γνώριμο στους περισσότερους χρήστες και καθώς ήταν ήδη εγκατεστημένο στο υπολογιστικό σύστημα του χρήστη. Τα βασικά controls που το απαρτίζουν είναι ένα panel (panelSearch) ,τρεις forms (form1, formPrint, formStats), δύο tabControls (tabcontrol1 , tabcontrolKartel) και ένα menuStrip. Η υλοποίηση του έγινε εξολοκλήρου με το ολοκληρωμένο περιβάλλον ανάπτυξης Visual Studio 2008.

5.2.1 Το κεντρικό παράθυρο της εφαρμογής

Στην εικόνα φαίνεται το κεντρικό παράθυρο της εφαρμογής που εμφανίζεται όταν γίνει εκκίνηση του προγράμματος. Είναι η κύρια φόρμα της εφαρμογής μας (Form1). Οι επιλογές για τις λειτουργίες της εφαρμογής βρίσκονται στο μενού επιλογών. Οι βασικότερες και περισσότερο χρησιμοποιούμενες από αυτές βρίσκονται και στην επιφάνεια εργασίας της εφαρμογής με την μορφή κουμπιών συντόμευσης. Τα κουμπιά αυτά είναι έξι, ένα για κάθε παράθυρο και είναι ορατά καθ' όλη τη διάρκεια χρήσης του προγράμματος προκειμένου ο χρήστης να είναι σε θέση να προβάλει με ευκολία τις διάφορες λειτουργίες της εφαρμογής. Οι λειτουργίες των κουμπιών και μενού είναι οι εισαγωγή νέου ασθενούς, υπάρχων ασθενής, εγχειρίδιο χρήσης, προβολή στατιστικών, εκτύπωση, back up και η έξοδος του προγράμματος. Η αρχική οθόνη περιέχει ένα tabControl (tabcontrol1) με δύο tabPages μία για εισαγωγή νέου ασθενούς και μία για τους υπάρχοντες ασθενείς.

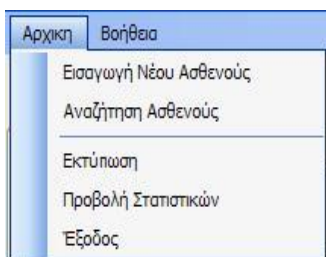


Εικόνα 14: Το περιβάλλον εργασίας της εφαρμογής.

Μενού επιλογών

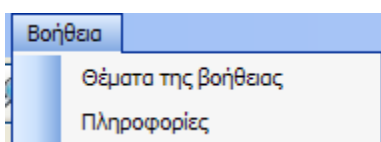
Όπως ήδη αναφέρθηκε, όλες οι λειτουργίες του προγράμματος, είναι προσβάσιμες και από το μενού επιλογών. Υπάρχουν δύο μενού :

1. Το «αρχική» το οποίο περιλαμβάνει διαδικασίες για τη διαχείριση των ασθενών. Το μενού αυτό περιλαμβάνει πέντε υπό μενού.



- **Εισαγωγή νέου ασθενούς.** Με αυτή την επιλογή ο χρήστης μεταφέρεται στο Tabpage “Νέος ασθενής” το οποίο και καθίσταται ενεργό.
- **Αναζήτηση ασθενούς.** Με αυτή την επιλογή ο χρήστης μεταφέρεται στο Tabpage “Υπάρχων ασθενής” το οποίο και καθίσταται ενεργό.
- **Εκτύπωση.** Με αυτή την επιλογή ο χρήστης μεταφέρεται στο παράθυρο της φόρμας PrintForm, όπου και παρέχονται οι επιλογές εκτύπωσης. Για αυτήν την φόρμα θα αναφερθούμε αναλυτικότερα παρακάτω.
- **Προβολή στατιστικών.** Με αυτή την επιλογή ο χρήστης μεταφέρεται στο παράθυρο της φόρμας FormStats, όπου και παρέχονται οι επιλογές εμφάνισης των στατιστικών. Για αυτήν την φόρμα θα αναφερθούμε αναλυτικότερα παρακάτω.
- **Έξοδος.** Με αυτή την επιλογή ο χρήστης κλείνει την εφαρμογή.








2. Το «**Βοήθεια**» το οποίο περιλαμβάνει παραπομπές σε αρχεία βοήθειας. Το μενού αυτό περιλαμβάνει δύο υπομενού.



- **Θέματα βοήθειας.** Με αυτή την επιλογή ο χρήστης εμφανίζει ένα αρχείο βοήθειας των windows που περιέχει βοήθεια χρήσης της εφαρμογής σε μορφή HTML το οποίο και θα αναλύσουμε παρακάτω.
- **Πληροφορίες.** Με αυτή την επιλογή εμφανίζεται παράθυρο με τις πληροφορίες σχετικές του λογισμικού μας.

Κουμπιά συντόμευσης

Όπως αναφέραμε και παραπάνω τα κουμπιά συντομεύσεων είναι επτά. Οι ενέργειες που εκτελούνται κατά το πάτημα των κουμπιών είναι ισοδύναμες με αυτές των μενού που αναφερθήκαμε παραπάνω.

1. Κουμπί Νέου ασθενούς: 
2. Κουμπί Υπάρχοντος ασθενούς: 
3. Κουμπί Στατιστικών: 
4. Κουμπί Εκτύπωσης: 
5. Κουμπί Βοήθειας: 
6. Κουμπί Εξόδου: 
7. Κουμπί back up της βάσης δεδομένων: 

TabControl1

Το TabControl1 είναι το κύριο στοιχείο ελέγχου του αρχικού περιβάλλοντος και περιλαμβάνει 2 tabpages την **tabpage1** με κείμενο Εισαγωγή Νέου Ασθενούς και την **tabpage2** με κείμενο Υπάρχων Ασθενής.

TabPage1 : Εισαγωγή Νέου Ασθενούς

Εικόνα 15: Tab page Νέος Ασθενής

Περιλαμβάνει έξι textbox εισαγωγής δεδομένων για ονοματεπώνυμο, πατρώνυμο, διεύθυνση, τηλέφωνο, ημερομηνία γέννησης και Α.Μ.Κ.Α καθώς και τρία buttons καταχώρηση reset και επιστροφή. Οι λειτουργίες των κουμπιών είναι οι εξής:

- **Καταχώρηση.** Καταχωρεί τα βασικά στοιχεία από το tab page Νέου ασθενή στην βάση δεδομένων. Αμέσως μετά την εκτέλεση εμφανίζεται μήνυμα επιβεβαίωσης και κατόπιν οι καρτέλες του νέου ασθενούς. Μετά την καταχώρηση ο τίτλος του αλλάζει σε «Αποθήκευση» και η λειτουργία του αλλάζει σε αποθήκευση των δεδομένων των καρτελών. Το κουμπί «Επιστροφή» επίσης γίνεται ενεργό.
- **Reset.** Καθαρίζει όλες τις εγγραφές από τα controls του tabcontrol1 και tabcontrol καρτελών.
- **Επιστροφή.** ο κουμπί αυτό είναι ανενεργό όσο δεν έχει καταχωρηθεί ένας νέος ασθενής και γίνεται ενεργό όταν καταχωρηθεί νέος ασθενής, ώστε να υπάρχει η επιλογή καταχώρησης και νέου ασθενούς.

TabPage2: Υπάρχων Ασθενής

id	Όνοματεπώνυμο	Πατρώνυμο	Τηλέφωνο	Διεύθυνση	Ημερομηνία γέννησης	ΑΜΚΑ
1	Στυλιανού Σταύρος		00000	Σοφίτα	26/6/1952	26065200888
2	Στυλιανού Σταύρος		00000	Σοφίτα	26/6/1952	
3	Στυλιανού Σταύρος1		00000	Σοφίτα	26/6/1952	
4	Ντάλλας Νέστωρ		00000	Σοφίτα	26/6/1952	
5	Φρονιμάκης Νικόλαος		00000	Ιεράπετρα	1/2/1900	
6	Καπριδάκη Αργυρώ		00000	Χανιά	30/1/1900	
7	Καπριδάκη Μαρία		00000	Χανιά	1/2/1900	
8	Νταλάρας Γεώργιος		00000	Κιάτωνα	30/1/1900	
9	Νικολάου Νικόλαος		00000	Πετράλωνα	4/1/1900	
10	Δεμερτζή Χριστίνα		00000	Καϊθήα	1/2/1900	

Εικόνα 16 Tab page Υπάρχων Ασθενής

Περιλαμβάνει ένα combobox που δέχεται ονοματεπώνυμο και φορτώνει όλα τα ονοματεπώνυμα από την βάση στοιχισμένα αλφαβητικά και συμπληρώνει τα υπόλοιπα πεδία του tab page από την βάση δεδομένων ανάλογα με την καταχώρηση που επιλέχθηκε. Επίσης περιλαμβάνει δύο Label ένα με το ID ασθενούς, ένα για προβολή της ηλικίας, πέντε textbox για προβολή του πατρώνυμου, διεύθυνση, τηλέφωνο, ημερομηνία γέννησης και Α.Μ.Κ.Α. Τα textbox αυτά λειτουργούν και ως πεδία αναζήτησης . Πληκτρολογώντας μια τιμή σε κάποιο πεδίο και πατώντας

enter, επιλέγει την πρώτη καταχώρηση που έχει αυτήν την τιμή και εστιάζοντας της στο datagridview. Επίσης φορτώνει το PanelSearch που περιέχει το Datagridview DT με όλες τις καταχωρήσεις ασθενών από την βάση καθώς και τρία κουμπιά διαχείρισης.

1. Διαγραφή. Διαγράφει την καταχώρηση που είναι επιλεγμένη στο Datagridview.
2. Προβολή .Εμφανίζει τις καρτέλες της καταχώρησης του ασθενή που είναι επιλεγμένη
3. Εκτύπωση Εμφανίζει την φόρμα εκτύπωσης.

TabControl Καρτελών

Εικόνα 17: Tab Control Καρτελών

Περιλαμβάνει 6 tabpages με τις καρτέλες και τα ερωτηματολόγια των εξετάσεων και κουμπιά για εμφάνιση προηγούμενης και επόμενης καρτέλας. Αυτό το control είναι εμφανές κατά την καταχώρηση νέου Ασθενούς και κατά την προβολή και επεξεργασία υπάρχοντος ασθενούς. Η λειτουργία των πεδίων που περιέχει είναι για την καταχώρηση των στοιχείων των εξετάσεων όπως και για την προβολή τους. Κατά την εμφάνιση των καρτελών «Κλινική εξέταση», «Διαγνωστικός έλεγχος» και «Παρακολούθηση ασθενών με καρκίνο του μαστού», μπορούν να έχουν πολλαπλές εγγραφές για τον ίδιο ασθενή. Για την απεικόνιση αυτών των εγγραφών δημιουργήσαμε τα groupBoxes gbxClinical, gbxDiagnosis, gbxObserve.

Εικόνα 18: Data grid view Κλινικής Εξέτασης

Τα groupBoxes αυτά είναι ενεργά και εμφανή μόνο όταν οι αντίστοιχες καρτέλες είναι επιλεγμένες. Περιέχουν ένα DateTimePicker με το οποίο μπορούμε να επιλέξουμε την ημερομηνία που έγινε η εξέταση, ένα Datagridview στο οποίο εμφανίζονται όλες οι καταχωρήσεις καρτελών για τον εκάστοτε ασθενή. Οι διάφορες καταχωρήσεις εμφανίζονται με βάση την ημερομηνία της εξέτασης και με διπλό κλικ σε μια γραμμή του πίνακα, προβάλλονται τα δεδομένα αυτής στην εκάστοτε καρτέλα. Εμπεριέχονται επίσης δύο κουμπιά ένα για καταχώρηση όπου καταχωρούνται τα υπάρχοντα στοιχεία ως νέα εγγραφή και ένα για διαγραφή όπου διαγράφεται η επιλεγμένη καταχώρηση από τον πίνακα.

Tabpage Παρούσα νόσος / Οικογενειακό ιστορικό

Εικόνα 19: Tab page Παρούσα νόσος/Οικογενειακό ιστορικό

Περιέχει 31 textbox , 11 combobox εισαγωγής στοιχείων. Ορισμένα textbox και combobox είναι απενεργοποιημένα αρχικά και ενεργοποιούνται εφόσον η τιμή του combobox που τους αντιστοιχεί είναι «Ναι».

Tabpage Παράγοντες κινδύνου

Εικόνα 20: Tab page Παράγοντες κινδύνου

Περιλαμβάνει 6 textbox , 6 combobox εισαγωγής στοιχείων και 3 groupbox . Ορισμένα textbox είναι απενεργοποιημένα αρχικά και ενεργοποιούνται εφόσον η τιμή του combobox που τους αντιστοιχεί είναι «Ναι» .Τα groupbox περιλαμβάνουν 21 textbox και labels και ενεργοποιούνται εφόσον η τιμή του combobox που τους αντιστοιχεί είναι «Ναι». Διαφορετικά απενεργοποιούνται και καθαρίζουν ότι τιμές περιλαμβάνουν τα controls τους.

Tabpage Κλινική εξέταση

Εικόνα 21: Tab page Κλινική εξέταση

Περιέχει 34 labels, 8 textbox, 3 combobox όπου το ένα περιέχει εικονίδια, 44 checkbox εισαγωγής στοιχείων. Επίσης περιλαμβάνει ένα groupBox που έχει 12 checkbox και 6 labels και ενεργοποιείται εφόσον η τιμή του combobox που του αντιστοιχεί είναι «Ναι».

Tabpage Διαγνωστικός έλεγχος

Εικόνα 22: Tab page Διαγνωστικός Έλεγχος

Περιέχει 56 labels , 36 checkbox , 1 combobox, 24 textbox εισαγωγής στοιχείων. Περιέχει ένα σύστημα εικόνας όπου δίνεται η δυνατότητα στο χρήστη να σημειώσει μέχρι 10 κουκίδες επάνω της

που αποθηκεύονται σε μορφή συντεταγμένων στη βάση και να επιλέξει το πάχος της κουκίδας μέσω του trackbar.

Tabpage Σταδιοποίηση καρκίνου

The screenshot shows a software interface for breast cancer staging. It features several input fields and checkboxes organized into sections: TNM, Clinical Examination, Histopathological Type, Metastatic Status, and Treatment. At the bottom, there are navigation buttons for previous and next screens.

Εικόνα 23: Tab Page Σταδιοποίηση καρκίνου

Περιέχει 31 textbox, 11 combobox εισαγωγής στοιχείων. Ορισμένα textbox και combobox είναι απενεργοποιημένα αρχικά και ενεργοποιούνται εφόσον η τιμή του combobox που τους αντιστοιχεί είναι «Ναι» .

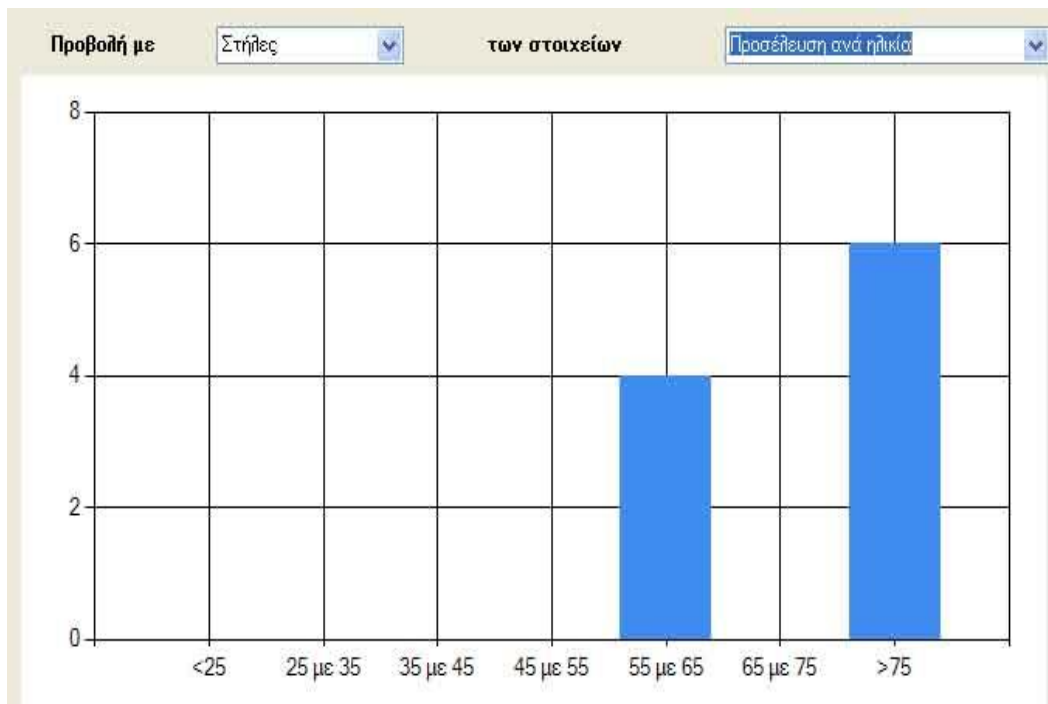
Tabpage Παρακολούθηση Ασθενών με καρκίνο του μαστού

The screenshot shows a software interface for monitoring breast cancer patients. It includes sections for 'Mhnes meta tin epeμβasi' (Time since surgery), 'Trauma', 'Plastiki apokatactasi' (Plastic reconstruction), 'Parakliniki eleyxos' (Systemic exams), 'Aktinografía' (Actinography), and 'Yperhografhima' (Superography). It contains numerous textboxes and checkboxes for data entry.

Εικόνα 24: Tab page Παρακολούθηση Ασθενών με καρκίνο μαστού

Περιλαμβάνει 31 texbox, 11 combobox εισαγωγής στοιχείων. Ορισμένα textbox και combobox είναι απενεργοποιημένα αρχικά και ενεργοποιούνται εφόσον η τιμή του combobox που τους αντιστοιχεί είναι «Ναι» .

FormStats



Εικόνα 25: Form Στατιστικών

Αποτελεί την φόρμα στατιστικών. Περιλαμβάνει δύο comboBox και ένα ms chart. Τα δύο comboBox περιέχουν επιλογές για το τι είδους στατιστικά και σε ποια μορφή θα τα προβάλει το ms chart.

Φόρμα Backup



Εικόνα 26: Form Αντιγράφων ασφαλείας

Η φόρμα backup περιλαμβάνει 2 κουμπιά, ένα για δημιουργία αντιγράφου ασφαλείας της βάσης δεδομένων και ένα για την επαναφορά της βάσης από υπάρχον αντίγραφο ασφαλείας. Στο πάτημα των κουμπιών ανοίγει ένας file explorer για την επιλογή της θέσης της αποθήκευσης ενός καινούριου αντιγράφου ασφαλείας ή για την εύρεση της θέσης ενός υπάρχοντος αντιγράφου ασφαλείας.

Φόρμα Print

ID	Όνοματεπώνυμο	Ημερομηνία	Τηλέφωνο
6	Καπριδάκη Αργυρώ	Χανιά	00000

ΚΛΙΝΙΚΗ ΕΞΕΤΑΣΗ

Κλινικά ευρήματα

Μαστός		Θηλή	
Δεξιό	Αριστερό	Δεξιό	Αριστερό
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Κυστικό σφίξιμο	Μέγεθος (cm)	Εισολική θηλής	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Συμπαγές σφίξιμο	Μέγεθος (cm)	Εσχάρα θηλής	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Διέχυτα οζίδια		Ερυθρότητα θηλής	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Τοπική οξείδρα	Μέγεθος (cm)	Ασμετρικό έκκριμα θηλής	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Διέχυτη ευαισθησία		Μη αμετρικό έκκριμα θηλής	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Τοπική ευαισθησία		Ανωμαλίες θηλάδας/άλλω	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ουλή			Είδος <input type="text"/>
<input type="checkbox"/>	<input type="checkbox"/>		
Εισολική δέρματος			
<input type="checkbox"/>	<input type="checkbox"/>		
Διήθηση δέρματος			
<input type="checkbox"/>	<input type="checkbox"/>		
Peau d'orange			
<input type="checkbox"/>	<input type="checkbox"/>		
Συμφύεται στην περιτονία			
<input type="checkbox"/>	<input type="checkbox"/>		

Προηγούμενες επεμβάσεις

Προηγούμενες επεμβάσεις	
Δεξιό	Αριστερό
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

Εικόνα 27: Form Εκτύπωσης

Αποτελεί την φόρμα της εκτύπωσης και περιλαμβάνει όλες τις αναφορές των Crystal Reports για τα στοιχεία ασθενών και των εξετάσεών τους. Παρέχουμε την δυνατότητα, μέσω checkboxes να επιλεχθούν όλες οι εξετάσεις, ορισμένες εξ αυτών ή καμία. Στο επάνω μέρος της φόρμας τοποθετήσαμε εργαλεία πλοήγησης όπου μπορεί να γίνει η μετάβαση στην προηγούμενη ή επόμενη καρτέλα, να γίνει ανανέωση της φόρμας.

Select Printer

Add Printer Lexmark X63 LexmarkFax Microsoft Office Doc... Microsoft XPS Document ...

Status: Offline Print to file Preferences

Location: Find Printer...

Comment:

Page Range

All Selection Current Page

Pages: 1-6

Enter either a single page number or a single page range. For example, 5-12

Number of copies: 1

Collate

Print Cancel Apply

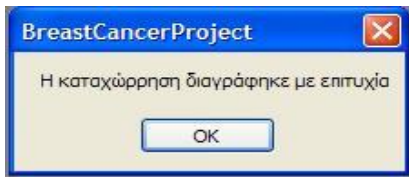
Εικόνα 28: Παράθυρο επιλογών εκτύπωσης

Πατώντας το εικονίδιο της εκτύπωσης της φόρμας Print ανοίγει ένα παράθυρο επιλογών της εκτύπωσης. Εδώ μπορεί να επιλεχθεί ο εκτυπωτής που θα διεκπεραιώσει την εκτύπωση, ποιες από τις σελίδες και πόσες είναι να εκτυπωθούν.

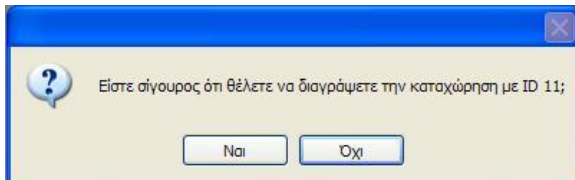
5.2.2 Παράθυρα διαλόγου μηνυμάτων

Πολλές φορές είναι απαραίτητο να μεταφέρουμε στους χρήστες μηνύματα επιβεβαίωσης, προειδοποίησης ή ενημέρωσης για μία ενέργεια τους μέσω ενός παραθύρου διαλόγου. Τα παράθυρα διαλόγου μηνυμάτων θα πρέπει να προσδιορίζουν με σαφήνεια τον σκοπό του χρήστη στην πρώτη γραμμή του κειμένου του παραθύρου διαλόγου ώστε να μεταδώσουμε τις πληροφορίες που θέλουμε με αποτελεσματικό τρόπο. Για το σκοπό αυτό στην εφαρμογή μας έχουμε χρησιμοποιήσει παράθυρα επιβεβαίωσης και ενημέρωσης. Παραθέτουμε ορισμένα ενδεικτικά μηνύματα.

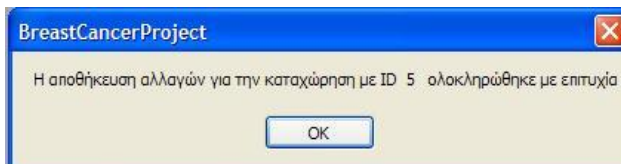
“Η καταχώρηση διαγράφηκε με επιτυχία”. Μήνυμα ενημέρωσης ότι η καταχώρηση ασθενούς διαγράφηκε με επιτυχία από την βάση δεδομένων.



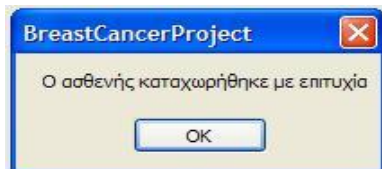
“Είστε σίγουρος ότι θέλετε να διαγράψετε την καταχώρηση με ID” Μήνυμα επιβεβαίωσης ότι η καταχώρηση με το συγκεκριμένο ID είναι εκείνη που πρόκειται να διαγραφεί. Απαιτείται η επιβεβαίωση από τον χρήστη.



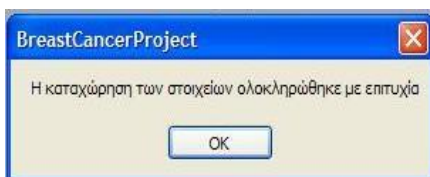
“Η αποθήκευση αλλαγών για την καταχώρηση με ID ολοκληρώθηκε με επιτυχία”. Μήνυμα ενημέρωσης ότι αποθηκεύτηκαν οι αλλαγές στην καταχώρηση με ID που επιλέχθηκε.



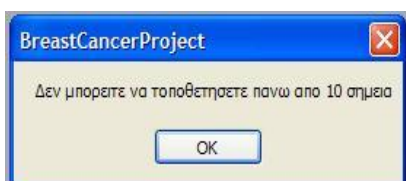
“Ο ασθενής καταχωρήθηκε με επιτυχία”. Μήνυμα ενημέρωσης ότι η καταχώρηση ασθενούς ολοκληρώθηκε με επιτυχία στην βάση δεδομένων.



"Η καταχώρηση των στοιχείων ολοκληρώθηκε με επιτυχία" Μήνυμα ενημέρωσης ότι καταχωρήθηκαν τα στοιχεία των εξετάσεων στην βάση δεδομένων.



"Δεν μπορείτε να τοποθετήσετε πάνω από 10 σημεία". Μήνυμα ενημέρωσης ξεπεράστηκε ο μέγιστος αριθμός σημείων στην εικόνα του διαγνωστικού ελέγχου.

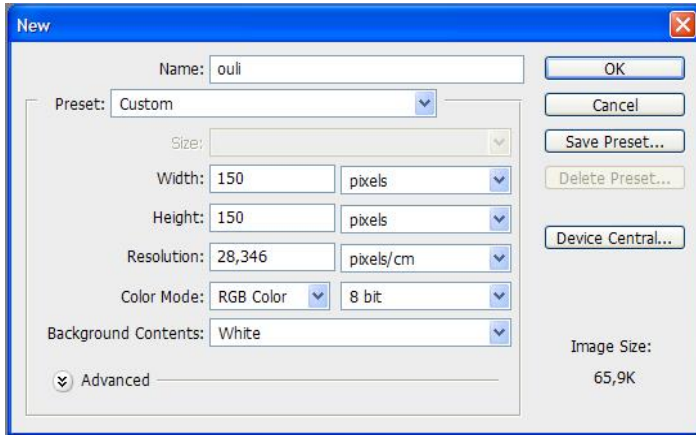


"Ο διαγνωστικός έλεγχος καταχωρήθηκε με επιτυχία". Μήνυμα ενημέρωσης ότι η εξέταση διαγνωστικού ελέγχου καταχωρήθηκε με επιτυχία στην βάση δεδομένων

5.3 Δημιουργία εικονιδίων και γραφικών της εφαρμογής

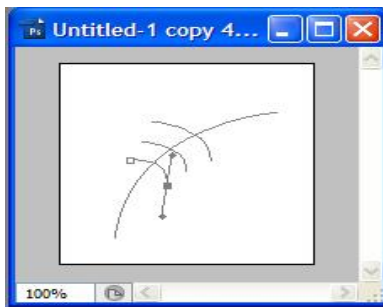
Για την δημιουργία της διεπαφής του προγράμματός μας χρειαστήκαμε να δημιουργήσουμε και να επεξεργαστούμε αρχεία εικόνων με την βοήθεια του Adobe Photoshop. Συγκεκριμένα στην καρτέλα «Διαγνωστικός Έλεγχος» χρειάστηκε να σχεδιάσουμε έξι εικονίδια που προσδιορίζουν τα ευρήματα αυτού του ελέγχου. Η πορεία εργασίας που ακολουθήσαμε ήταν η εξής:

Δημιουργήσαμε ένα κενό αρχείο μεγέθους 150x150 pixels.



Εικόνα 29: Παράθυρο δημιουργίας κενής εικόνας

Επιλέξαμε το εργαλείο pen και με αυτό σχεδιάσαμε το παρακάτω σχήμα (τύπου path).



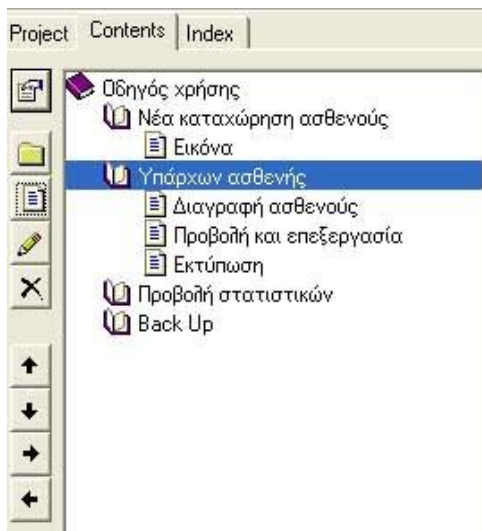
Εικόνα 30: Σχέδιο εικονιδίου

Στην συνέχεια επιλέξαμε Stroke Path και δώσαμε στο path πάχος τύπου pensil.



Εικόνα 31: Πάχος σχεδίου εικονιδίου

Κατά αυτόν τον τρόπο δημιουργήθηκαν όλα τα εικονίδια αυτής της καρτέλας. Για τις υπόλοιπες εικόνες χρειάστηκε να επεξεργαστούμε τα γραφικά στοιχεία τους όπως είναι η ανάλυση και το ιστόγραμμα προκειμένου να ταιριάζουν με το χρωματικό ύφος της εφαρμογής. Συγκεκριμένα το ιστόγραμμα αναπαριστά γραφικά, την φωτεινότητα σε σχέση με τον αριθμό των εικονοστοιχείων.



Εικόνα 33: Index του αρχείου βοήθειας.

Τέλος με την επιλογή compile HTML file δημιουργήθηκε το αρχείο βοήθειας με κατάληξη chm. Οι σελίδες HTML και οι εικόνες καθώς και το index της βοήθειας εμπεριέχονται στο αρχείο αυτό το οποίο έχει συμπιεστεί και καταλαμβάνει λιγότερο χώρο στο δίσκο

5.5 Ανάπτυξη Βάσης Δεδομένων

Η ανάπτυξη της βάσης δεδομένων έγινε εξ ολοκλήρου με το πρόγραμμα Ms Access όπου για κάθε καρτέλα με στοιχεία ασθενών και εξετάσεων δημιουργήθηκε ξεχωριστός πίνακας με στήλες τα δεδομένα των καρτελών. Επίσης στην Ms Access δημιουργήθηκαν ορισμένα SQL ερωτήματα για ορισμένα αποτελέσματα που απαιτούνταν για συγκεκριμένες λειτουργίες του προγράμματος.

5.5.1 Σκοπός δημιουργίας της βάσης δεδομένων και ερωτήματα που θέλουμε αυτή να απαντά.

Η δημιουργία της βάσης δεδομένων των ασθενών της εφαρμογής μας έχει ως σκοπό να δώσει τις απαραίτητες πληροφορίες για την καταγραφή και εξέλιξη της υγείας τους, καθώς και για την εξαγωγή στατιστικών συμπερασμάτων. Έτσι τα ερωτήματα και τα δεδομένα που θέλουμε να απαντά η βάση είναι τα παρακάτω:

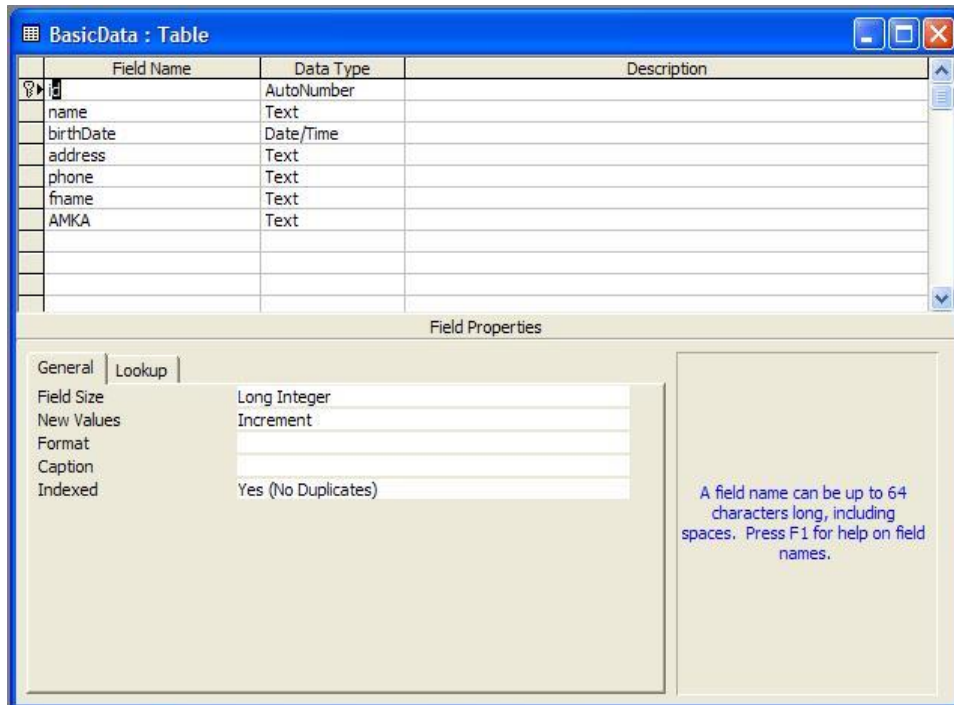
- Προσωπικά δεδομένα ασθενών.
- Δεδομένα παρούσας νόσου και οικογενειακού ιστορικού.
- Δεδομένα παραγόντων κινδύνου.
- Δεδομένα κλινικών εξετάσεων.
- Δεδομένα διαγνωστικού ελέγχου.
- Δεδομένα σταδιοποίησης καρκίνου.
- Δεδομένα παρακολούθησης ασθενών
- Στατιστικά ηλικιών ασθενών.
- Στατιστικά οικογενειακού ιστορικού.
- Στατιστικά νόσων.

5.5.2 Πίνακες, Γνωρίσματα (Στήλες) που περιέχονται σε αυτούς και περιορισμοί Γνωρισμάτων.

Γίνεται μία εκτενής αναφορά στους πίνακες της βάσης δεδομένων μας καθώς και στα γνωρίσματά, τους τα στοιχεία δηλαδή που κατακρατούνται στη βάση για κάθε ασθενή, καθώς και στους περιορισμούς που τίθενται σε αυτά, στις τιμές δηλαδή που μπορεί να πάρει το κάθε γνώρισμα. Θα ξεκινήσουμε την ανάλυσή μας από τον κυριότερο πίνακα της βάσης, αυτόν δηλαδή που κρατάει τα στοιχεία εκείνα τα οποία έχουν σχέση με τα προσωπικά δεδομένα του κάθε ασθενή.

Να σημειωθεί ακόμα ότι σχεδόν όλοι οι άλλοι πίνακες σχετίζονται με τον πίνακα αυτό (BasicData) μέσω σχέσεων (relationships). Ο πίνακας αυτός αποτελεί μία πατρική κατά κάποιο τρόπο οντότητα για τις υπόλοιπες μίας και περιέχει ξένα κλειδιά τα οποία είναι πρωτεύοντα κλειδιά στους άλλους πίνακες (οντότητες) της βάσης. Αυτό συμβαίνει για την υλοποίηση των σχέσεων (relationships) μεταξύ των δεδομένων των οντοτήτων της βάσης.

Πίνακας Στοιχεία Ασθενή



Εικόνα 34: Πίνακας BasicData

- **id** : Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- **AMKA** : Στο πεδίο αυτό αποθηκεύεται ο ΑΜΚΑ του ασθενή. Μπορεί να αποθηκευτεί ένα μεταβλητού μήκους αλφαριθμητικό με μέγιστο μήκος 50 χαρακτήρων. Το πεδίο αυτό μπορεί να λάβει μόνο χαρακτήρες μικρούς και μεγάλους. Δεν επιτρέπεται να υπάρχουν στο μεταβλητού μήκους αλφαριθμητικό νούμερα και σύμβολα όπως τελείες, κόμματα κ.λ.π. Επίσης δεν επιτρέπονται οι τιμές null.
- **name** : Στο πεδίο αυτό αποθηκεύεται το ονοματεπώνυμο του ασθενή. Μπορεί να αποθηκευτεί ένα μεταβλητού μήκους αλφαριθμητικό με μέγιστο μήκος 50 χαρακτήρων. Το πεδίο αυτό μπορεί να λάβει μόνο χαρακτήρες μικρούς και μεγάλους. Δεν επιτρέπεται να υπάρχουν στο μεταβλητού μήκους αλφαριθμητικό νούμερα και σύμβολα όπως τελείες, κόμματα κ.λ.π. Επίσης δεν επιτρέπονται οι τιμές null.
- **birthDate** : Εδώ εισάγεται η ημερομηνία γέννησης του ασθενή. Το πεδίο είναι χωρισμένο στα υποπεδία χρόνος, μήνας και ημέρα. Το κάθε πεδίο επιτρέπεται να λάβει μονάχα τιμές αριθμητικές ενώ γίνεται και έλεγχος για την ορθότητα των τιμών δηλαδή δεν επιτρέπεται η εισαγωγή μήνα μικρότερου από 1 και μεγαλύτερου από 12, για την ημέρα πάλι όχι τιμή μεγαλύτερη από 31 και μικρότερη του 1 εκτός και αν είναι Φεβρουάριος οπότε και δεν επιτρέπεται η εισαγωγή μεγαλύτερη από 29 ενώ για τον χρόνο ο μοναδικός έλεγχος που γίνεται είναι να είναι τετραψήφιος ο αριθμός. Null τιμές επιτρέπονται.
- **address** : Στο πεδίο αυτό αποθηκεύεται η διεύθυνση του ασθενή. Μπορεί να αποθηκευτεί ένα μεταβλητού μήκους αλφαριθμητικό με μέγιστο μήκος 50 χαρακτήρων. Το πεδίο αυτό μπορεί να λάβει μόνο χαρακτήρες μικρούς και μεγάλους. Null τιμές επιτρέπονται.

- **fname**: Στο πεδίο αυτό αποθηκεύεται το πατρώνυμο του ασθενή. Μπορεί να αποθηκευτεί ένα μεταβλητού μήκους αλφαριθμητικό με μέγιστο μήκος 50 χαρακτήρων. Το πεδίο αυτό μπορεί να λάβει μόνο χαρακτήρες μικρούς και μεγάλους. Null τιμές επιτρέπονται.
- **phone**: Στο πεδίο αυτό αποθηκεύεται το τηλέφωνο του ασθενή. Μπορεί να αποθηκευτεί ένα μεταβλητού μήκους αλφαριθμητικό με μέγιστο μήκος 50 χαρακτήρων. Το πεδίο αυτό μπορεί να λάβει μόνο χαρακτήρες μικρούς και μεγάλους. Null τιμές επιτρέπονται.

Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
id	AutoNumber	Μοναδικός κωδικός/ Πρωτεύων κλειδί	Ναι	Όχι
AMKA	Text	ΑΜΚΑ	Όχι	Όχι
name	Text	Όνοματεπώνυμο	Όχι	Όχι
birthDate	Date	Ημερομηνία γέννησης	Όχι	Ναι
address	Text	Διεύθυνση	Όχι	Ναι
phone	Text	Τηλέφωνο	Όχι	Ναι
fname	Text	Πατρώνυμο	Όχι	Ναι

Πίνακας 14: Πίνακας BasicData

Πίνακας DiseasesData

Στον πίνακα αυτό αποθηκεύονται τα στοιχεία της καρτέλας Παρούσα Νόσος/Ατομικό ιστορικό. Τα στοιχεία που κρατούνται για κάθε ασθενή φαίνονται από τα γνωρίσματα του πίνακα τα οποία είναι :

- id : Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- [Διάφορα Πεδία checkboxes] : Στα πεδία αυτά αποθηκεύονται δίτιμες τιμές που αφορούν την καρτέλα Παρούσα Νόσος/Ατομικό ιστορικό με την μορφή CheckBox.
- [Διάφορα Πεδία text] : Στα πεδία αυτά αποθηκεύονται τιμές που αφορούν την καρτέλα Παρούσα Νόσος/Ατομικό ιστορικό με την μορφή κειμένου.

Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
id	AutoNumber	Μοναδικός κωδικός/ Πρωτεύων κλειδί	Ναι	Όχι
Διάφορα Πεδία combo και check boxes	Check Box	Καταγραφή συμπερασμάτων και εξέτασης	Όχι	-
Διάφορα Πεδία text	Text	Καταγραφή συμπερασμάτων και εξέτασης	Όχι	Ναι

Πίνακας 15: Πίνακας DiseasesData

Πίνακας DangersData

Στον πίνακα αυτό αποθηκεύονται τα στοιχεία της καρτέλας Παράγοντες κινδύνου. Τα στοιχεία που κρατούνται για κάθε ασθενή φαίνονται από τα γνωρίσματα του πίνακα τα οποία είναι :

- id : Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- [Διάφορα Πεδία combo και check boxes] : Στα πεδία αυτά αποθηκεύονται δίτιμες τιμές που αφορούν την καρτέλα Παράγοντες κινδύνου με την μορφή CheckBox.
- [Διάφορα Πεδία text] : Στα πεδία αυτά αποθηκεύονται τιμές που αφορούν την καρτέλα Παράγοντες κινδύνου με την μορφή κειμένου.

Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
id	AutoNumber	Μοναδικός κωδικός/ Πρωτεύων κλειδί	Ναι	Όχι
Διάφορα Πεδία combo και check boxes	Check Box	Καταγραφή ιστορικού ασθενή	Όχι	-
Διάφορα Πεδία text	Text	Καταγραφή ιστορικού ασθενή	Όχι	Ναι

Πίνακας 16: Πίνακας DangersData

Πίνακας ClinicalData

Στον πίνακα αυτό αποθηκεύονται τα στοιχεία της καρτέλας Κλινική εξέταση. Τα στοιχεία που κρατούνται για κάθε ασθενή φαίνονται από τα γνωρίσματα του πίνακα τα οποία είναι :

- id : Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- [Διάφορα Πεδία combo και check boxes] : Στα πεδία αυτά αποθηκεύονται δίτιμες τιμές που αφορούν την καρτέλα Κλινικές εξετάσεις με την μορφή CheckBox.
- [Διάφορα Πεδία text] : Στα πεδία αυτά αποθηκεύονται τιμές που αφορούν την καρτέλα Κλινικές εξετάσεις με την μορφή κειμένου.

Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
id	AutoNumber	Μοναδικός κωδικός/ Πρωτεύων κλειδί	Ναι	Όχι
Διάφορα Πεδία combo και check boxes	Check Box	Καταγραφή συμπερασμάτων κλινικής εξέτασης	Όχι	-
Διάφορα Πεδία text	Text	Καταγραφή συμπερασμάτων κλινικής εξέτασης	Όχι	Ναι

Πίνακας 17: Πίνακας ClinicalData

Πίνακας DiagnosisData

Στον πίνακα αυτό αποθηκεύονται τα στοιχεία της καρτέλας Διαγνωστικού ελέγχου. Τα στοιχεία που κρατούνται για κάθε ασθενή φαίνονται από τα γνωρίσματα του πίνακα τα οποία είναι :

- id : Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- [Διάφορα Πεδία comboboxes και checkboxes] : Στα πεδία αυτά αποθηκεύονται δίτιμες τιμές που αφορούν την καρτέλα Διαγνωστικού ελέγχου με την μορφή checkBox.
- [Διάφορα Πεδία text] : Στα πεδία αυτά αποθηκεύονται τιμές που αφορούν την καρτέλα Διαγνωστικού ελέγχου με την μορφή κειμένου.

Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
id	AutoNumber	Μοναδικός κωδικός/ Πρωτεύων κλειδί	Ναι	Όχι
Διάφορα Πεδία combo και check boxes	Check Box	Καταγραφή συμπερασμάτων διαγνωστικού ελέγχου	Όχι	-
Διάφορα Πεδία text	Text	Καταγραφή συμπερασμάτων διαγνωστικού ελέγχου	Όχι	Ναι

Πίνακας 18: Πίνακας DiagnosisData

Πίνακας StadioData

Στον πίνακα αυτό αποθηκεύονται τα στοιχεία της καρτέλας Σταδιοποίηση καρκίνου. Τα στοιχεία που κρατούνται για κάθε ασθενή φαίνονται από τα γνωρίσματα του πίνακα τα οποία είναι :

- id : Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- [Διάφορα Πεδία comboboxes και checkboxes] : Στα πεδία αυτά αποθηκεύονται δίτιμες τιμές που αφορούν την καρτέλα Σταδιοποίηση καρκίνου με την μορφή checkBox.
- [Διάφορα Πεδία text] : Στα πεδία αυτά αποθηκεύονται τιμές που αφορούν την καρτέλα Σταδιοποίηση καρκίνου με την μορφή κειμένου.

Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
id	AutoNumber	Μοναδικός κωδικός/ Πρωτεύων κλειδί	Ναι	Όχι
Διάφορα Πεδία combo και check boxes	Check Box	Καταγραφή σταδιοποίησης καρκίνου	Όχι	-
Διάφορα Πεδία text	Text	Καταγραφή σταδιοποίησης καρκίνου	Όχι	Ναι

Πίνακας 19: Πίνακας StadioData

Πίνακας ObserveData

Στον πίνακα αυτό αποθηκεύονται τα στοιχεία της καρτέλας Παρακολούθηση ασθενούς. Τα στοιχεία που κρατούνται για κάθε ασθενή φαίνονται από τα γνωρίσματα του πίνακα τα οποία είναι :

- id : Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- [Διάφορα Πεδία combo και check boxes] : Στα πεδία αυτά αποθηκεύονται δίτιμες τιμές που αφορούν την καρτέλα Παρακολούθηση ασθενούς με την μορφή CheckBox.
- [Διάφορα Πεδία text] : Στα πεδία αυτά αποθηκεύονται τιμές που αφορούν την καρτέλα Παρακολούθηση ασθενούς με την μορφή κειμένου.

Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
id	AutoNumber	Μοναδικός κωδικός/ Πρωτεύον κλειδί	Ναι	Όχι
Διάφορα Πεδία combo και check boxes	Check Box	Καταγραφή παρακολούθησης ασθενούς	Όχι	-
Διάφορα Πεδία text	Text	Καταγραφή παρακολούθησης ασθενούς	Όχι	Ναι

Πίνακας 20: Πίνακας ObserveData

Πίνακας CoordinatesData

Στον πίνακα αυτό αποθηκεύονται τα στοιχεία της καρτέλας Παρακολούθηση ασθενούς. Τα στοιχεία που κρατούνται για κάθε ασθενή φαίνονται από τα γνωρίσματα του πίνακα τα οποία είναι :

- coordID: Το πεδίο αυτό αποτελεί το πρωτεύον κλειδί το πίνακα αυτού και κατά συνέπεια εμπίπτει στους περιορισμούς που εμπίπτουν όλα τα πρωτεύοντα κλειδιά. Δεν μπορεί να λάβει null τιμή ενώ η εισαγωγή του γίνεται αυτόματα από την εφαρμογή.
- diagnosisID : Στο πεδίο αυτό αποθηκεύεται ένας μοναδικός κωδικός που παίρνει την τιμή του από τον πίνακα του Διαγνωστικού ελέγχου. Ο τύπος του πεδίου αυτού είναι αριθμός και δεν μπορεί να λάβει τιμή null.
- Mpx : Στο πεδίο αυτό αποθηκεύεται οι τεταγμένη ενός σημείου που σχεδιάζεται στην καρτέλα του Διαγνωστικού ελέγχου. Ο τύπος του πεδίου αυτού είναι αριθμός και δεν μπορεί να λάβει τιμή null.
- Mpy : Στο πεδίο αυτό αποθηκεύεται οι τετμημένη ενός σημείου που σχεδιάζεται στην καρτέλα του Διαγνωστικού ελέγχου. Ο τύπος του πεδίου αυτού είναι αριθμός και δεν μπορεί να λάβει τιμή null.
- Paxos : Στο πεδίο αυτό αποθηκεύεται μιας μπάρας που επιλέγει το πάχος του σημείου στην καρτέλα του Διαγνωστικού ελέγχου. Ο τύπος του πεδίου αυτού είναι αριθμός και δεν μπορεί να λάβει τιμή null.

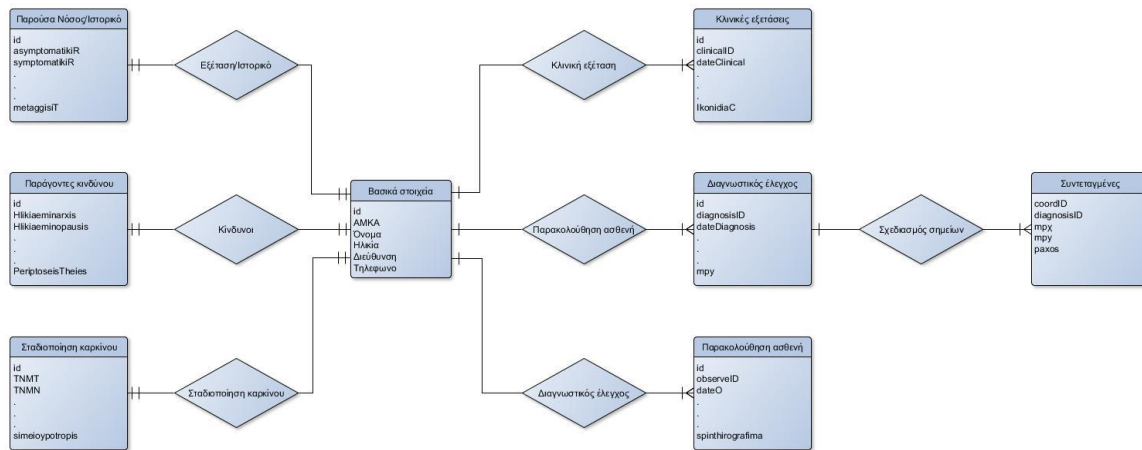
Πεδίο	Τύπος	Περιγραφή	Πρωτεύον Κλειδί	Null
coordID	AutoNumber	Μοναδικός κωδικός/ Πρωτεύον κλειδί	Ναι	Όχι
diagnosisID	Number	Κωδικός διασύνδεσης με Διαγνωστικό έλεγχο	Όχι	Όχι

mpx	Number	Τεταγμένη σημείου	Όχι	Όχι
mpy	Number	Τετμημένη σημείου	Όχι	Όχι
paxos	Number	Πάχος σημείου	Όχι	Όχι

Πίνακας 21: Πίνακας CoordinatesData

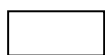
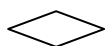



5.5.3 Διάγραμμα σχέσεων-οντοτήτων

Στην παράγραφο αυτή δίνεται γραφικά η σχεδίαση της βάσης μέσω του E-R (Entity Relationship Diagram) διαγράμματος και αναλύονται διεξοδικά οι σχέσεις ανάμεσα στους πίνακες (οντότητες) της βάσης και των δεδομένων τους. Πρέπει να αναφέρουμε ότι παραπάνω αναλύθηκαν οι βασικοί πίνακες της βάσης. Η ύπαρξη των πινάκων που χρησιμοποιούνται για την υλοποίηση των σύνθετων αυτών σχέσεων θα φανεί κατά την ανάλυση των σχέσεων αυτών. Οι διάφοροι τύποι σχέσεων που μπορούν να αναπτυχθούν ανάμεσα σε πίνακες και η σημασία τους αναλύονται κατά την αναλυτική ανάπτυξη των σχέσεων των πινάκων της βάσης.



Εικόνα 35: Διάγραμμα σχέσεων οντοτήτων

Στην παράγραφο αυτή γίνεται μία επεξήγηση του διαγράμματος οντοτήτων των συμβόλων του και τι σημαίνει το κάθε ένα. Έτσι λοιπόν έχουμε να εμφανίζονται στο διάγραμμα οντοτήτων τα παρακάτω σύμβολα τα οποία σημαίνουν:

- 
 • Με το σύμβολο αυτό δηλώνονται οι διάφορες οντότητες της βάσης. Κάθε σύμβολο αυτής της μορφής αντιστοιχεί σε ένα ξεχωριστό πίνακα της βάσης και το κείμενο που υπάρχει σε κάθε τέτοιο σύμβολο αντιστοιχεί στο όνομα του συγκεκριμένου πίνακα.
- 
 • πίνακα. Με το σύμβολο αυτό δηλώνουμε την σχέση (relationship) ανάμεσα στους πίνακες της βάσης. Παρατηρούμε ότι σύμβολα αυτής της μορφής υπάρχουν ανάμεσα σε δύο ή περισσότερους πίνακες.
- 
 • Με το σύμβολο αυτό δηλώνουμε σύνδεση (link) ανάμεσα στους πίνακες της βάσης τύπου από 0 έως 1 προς ένα (from zero to One), και γράφεται και ως 1:1.
- 
 • Με το σύμβολο αυτό δηλώνουμε σύνδεση ανάμεσα στους πίνακες της βάσης τύπου από 1 έως 1 (from One to One), και γράφεται και ως 1:1.
- 
 • Με το σύμβολο αυτό δηλώνουμε σύνδεση ανάμεσα στους πίνακες της βάσης τύπου από 0 έως πολλά (from zero to Many), και γράφεται και ως 1:N (To N μπορεί να είναι και 0).

Παρατηρούμε ότι κατά την γραφική συσχέτιση δύο πινάκων οι γραμμές που χρησιμοποιούμε ξεκινούν και καταλήγουν με κάποια σύμβολα. Όπως αναφέρουμε παραπάνω, με αυτόν τον τρόπο δηλώνεται ο τύπος της σχέσης ανάμεσα στους πίνακες. Στην βάση μας η σχέσεις αυτές είναι τύπου 1

προς 1, 1 προς 0 έως 1 και 1 προς 0 έως πολλά. Πρόκειται για σχέση δύο δρόμων και δηλώνει σε κάθε εισαγωγή του κάθε πίνακα πόσες εγγραφές του άλλου πίνακα σχετίζονται με τη συγκεκριμένη εφαρμογή. Αυτός ο έλεγχος γίνεται από και προς τις δύο κατευθύνσεις.

Για παράδειγμα για τη σχέση ανάμεσα στους πίνακες BasicData και DiseaseData. Από τη μεριά του πίνακα BasicData έχουμε ότι σε κάθε εισαγωγή στον πίνακα αυτό, δηλαδή για κάθε ασθενή, μπορεί να αντιστοιχεί καμία ή μία νόσος, δηλαδή έχουμε μία σχέση του τύπου 1:0-1. Από τη μεριά του πίνακα DiseaseData έχουμε ότι σε κάθε εγγραφή νόσου να αντιστοιχεί ένας και μόνο ένας ασθενής. Έχουμε δηλαδή μία σχέση του τύπου 1:1. Σε ένα άλλο παράδειγμα ας αναλύσουμε τη σχέση ανάμεσα στους πίνακες BasicData και ClinicalData. Από τη μεριά του πίνακα BasicData έχουμε ότι σε κάθε εισαγωγή στον πίνακα αυτό, δηλαδή για κάθε ασθενή, μπορεί να αντιστοιχούν πολλές κλινικές εξετάσεις, δηλαδή έχουμε μία σχέση του τύπου 1:N. Από τη μεριά του πίνακα ClinicalData έχουμε ότι σε κάθε εγγραφή νόσου να αντιστοιχεί ένας και μόνο ένας ασθενής. Έχουμε δηλαδή μία σχέση του τύπου 1:1.

5.5.4 Επεξήγηση των Σχέσεων των Πινάκων της Βάσης

Στην παράγραφο αυτή θα γίνει μία διεξοδική ανάλυση όλων των σχέσεων που έχουν αναπτυχθεί ανάμεσα στους πίνακες της βάσης. Οι σχέσεις λοιπόν που έχουν υλοποιηθεί θα αναλυθούν με βάση τους πίνακες που αυτές συσχετίζουν. Έχουμε λοιπόν τα ζευγάρια πινάκων τα οποία συσχετίζονται.

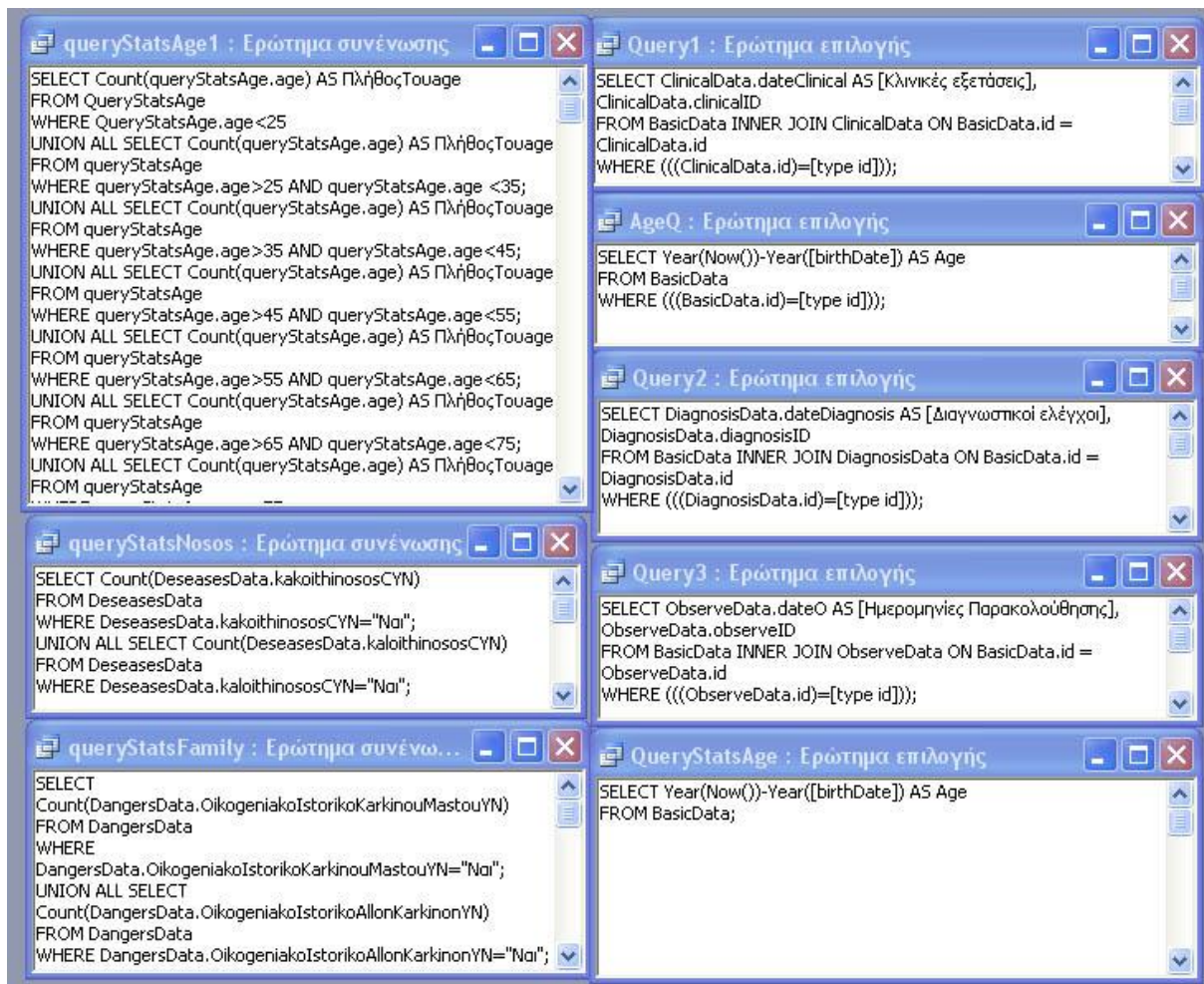
- **BasicData και DiseaseData.** Θεωρούμε για την συγκεκριμένη σχέση βλέποντας της από τη μεριά του πίνακα BasicData ότι πρώτα απ' όλα ο ασθενής υποβάλλεται σε εξέταση και τα συμπεράσματα είναι διαθέσιμα. Επιτρέπεται ο πίνακας DiseaseData να μη περιέχει εγγραφές που αφορούν τον συγκεκριμένο ασθενή, αν για κάποιο λόγο ο ιατρός δεν συμπληρώσει την καρτέλα «Παρούσα νόσος/ατομικό αναμνηστικό». Εξάλλου αυτή η καρτέλα μπορεί να συμπληρωθεί και αργότερα. Η καταχώρηση όμως των συμπερασμάτων γίνεται μία φορά και μετά επιτρέπονται μόνο διορθώσεις. Από την μεριά του πίνακα DiseaseData δεν μπορεί μια καταχώρησή του να αντιστοιχεί σε περισσότερους από έναν ασθενή. Για κάθε ασθενή δηλαδή, μπορεί να υπάρχει μία ή και καμία καταχώρηση της καρτέλας «Παρούσα νόσος/ατομικό αναμνηστικό»,για αυτόν τον λόγο η συγκεκριμένη σχέση είναι 1 έως 0 προς 1 (1:1). Η σχέση αυτή υλοποιείται με την εισαγωγή του πρωτεύοντος κλειδιού του πίνακα BasicData σαν ξένο κλειδί στον πίνακα DiseaseData. Το ξένο αυτό κλειδί εμπίπτει στους περιορισμούς που απαιτείται να εμπίπτουν όλα τα ξένα κλειδιά.

- **BasicData και DangersData:** Θεωρούμε για την συγκεκριμένη σχέση βλέποντας της από τη μεριά του πίνακα BasicData ότι επιτρέπεται ο πίνακας DangersData να μη περιέχει εγγραφές που αφορούν τον συγκεκριμένο ασθενή, αν για κάποιο λόγο ο ιατρός δεν συμπληρώσει την καρτέλα «Παράγοντες κινδύνου». Εξάλλου αυτή η καρτέλα μπορεί να συμπληρωθεί και αργότερα. Η καταχώρηση όμως των συμπερασμάτων γίνεται μία φορά και μετά επιτρέπονται μόνο διορθώσεις. Από την μεριά του πίνακα DangersData δεν μπορεί μια καταχώρησή του να αντιστοιχεί σε περισσότερους από έναν ασθενή. Για κάθε ασθενή δηλαδή, μπορεί να υπάρχει μία ή και καμία καταχώρηση της καρτέλας «Παράγοντες κινδύνου»,για αυτόν τον λόγο η συγκεκριμένη σχέση είναι 1 έως 0 προς 1 (1:1). Η σχέση αυτή υλοποιείται με την εισαγωγή του πρωτεύοντος κλειδιού του πίνακα BasicData σαν ξένο κλειδί στον πίνακα DangersData. Το ξένο αυτό κλειδί εμπίπτει στους περιορισμούς που απαιτείται να εμπίπτουν όλα τα ξένα κλειδιά.

- **BasicData και StadioData.** Θεωρούμε για την συγκεκριμένη σχέση βλέποντας της από τη μεριά του πίνακα BasicData ότι επιτρέπεται ο πίνακας StadioData να μη περιέχει εγγραφές που αφορούν τον συγκεκριμένο ασθενή, αν για κάποιο λόγο ο ιατρός δεν συμπληρώσει την καρτέλα «Σταδιοποίηση καρκίνου».Εξάλλου αυτή η καρτέλα μπορεί να συμπληρωθεί και αργότερα. Η καταχώρηση όμως των συμπερασμάτων γίνεται μία φορά και μετά επιτρέπονται μόνο διορθώσεις. Από την μεριά του πίνακα StadioData δεν μπορεί μια καταχώρησή του να αντιστοιχεί σε περισσότερους από έναν ασθενή. Για κάθε ασθενή δηλαδή, μπορεί να υπάρχει μία ή και καμία καταχώρηση της καρτέλας «Σταδιοποίηση καρκίνου»,για αυτόν τον λόγο η συγκεκριμένη σχέση είναι 1 έως 0 προς 1 (1:1). Η σχέση αυτή υλοποιείται με την εισαγωγή του πρωτεύοντος κλειδιού του πίνακα BasicData σαν ξένο κλειδί στον πίνακα StadioData. Το ξένο αυτό κλειδί εμπίπτει στους περιορισμούς που απαιτείται να εμπίπτουν όλα τα ξένα κλειδιά.

- **BasicData και ClinicalData:** Θεωρούμε για την συγκεκριμένη σχέση βλέποντας της από τη μεριά του πίνακα BasicData ότι επιτρέπεται ο πίνακας ClinicalData να μη περιέχει εγγραφές που αφορούν τον συγκεκριμένο ασθενή, αν για κάποιο λόγο ο ιατρός δεν συμπληρώσει την καρτέλα «Κλινικές εξετάσεις».Εξάλλου αυτή η καρτέλα μπορεί να συμπληρωθεί και αργότερα. Οι κλινικές εξετάσεις όμως μπορεί να είναι και περισσότερες από μία, για την ακρίβεια όσες και οι επισκέψεις του ασθενή. Η καταχώρηση κάθε μίας κλινικής εξέτασης γίνεται μία φορά και μετά επιτρέπονται μόνο διορθώσεις. Από την μεριά του πίνακα ClinicalData δεν μπορεί μια καταχώρησή του να αντιστοιχεί σε περισσότερους από έναν ασθενή. Για κάθε ασθενή δηλαδή, μπορεί να υπάρχουν πολλές ή και καμία καταχώρηση της καρτέλας «Κλινικές εξετάσεις»,για αυτόν τον λόγο η συγκεκριμένη σχέση είναι 1 προς 0 έως πολλά (1:N). Η σχέση αυτή υλοποιείται με την εισαγωγή του πρωτεύοντος κλειδιού του πίνακα BasicData σαν ξένο κλειδί στον πίνακα ClinicalData. Το ξένο αυτό κλειδί εμπίπτει στους περιορισμούς που απαιτείται να εμπίπτουν όλα τα ξένα κλειδιά.
- **BasicData και DiagnosisData:** Θεωρούμε για την συγκεκριμένη σχέση βλέποντας της από τη μεριά του πίνακα BasicData ότι επιτρέπεται ο πίνακας DiagnosisData να μη περιέχει εγγραφές που αφορούν τον συγκεκριμένο ασθενή, αν για κάποιο λόγο ο ιατρός δεν συμπληρώσει την καρτέλα «Διαγνωστικός έλεγχος».Εξάλλου αυτή η καρτέλα μπορεί να συμπληρωθεί και αργότερα. Οι κλινικές εξετάσεις όμως μπορεί να είναι και περισσότερες από μία, για την ακρίβεια όσες και οι επισκέψεις του ασθενή. Η καταχώρηση κάθε μίας κλινικής εξέτασης γίνεται μία φορά και μετά επιτρέπονται μόνο διορθώσεις. Από την μεριά του πίνακα DiagnosisData δεν μπορεί μια καταχώρησή του να αντιστοιχεί σε περισσότερους από έναν ασθενή. Για κάθε ασθενή δηλαδή, μπορεί να υπάρχουν πολλές ή και καμία καταχώρηση της καρτέλας «Διαγνωστικός έλεγχος»,για αυτόν τον λόγο η συγκεκριμένη σχέση είναι 1 προς 0 έως πολλά (1:N). Η σχέση αυτή υλοποιείται με την εισαγωγή του πρωτεύοντος κλειδιού του πίνακα BasicData σαν ξένο κλειδί στον πίνακα DiagnosisData. Το ξένο αυτό κλειδί εμπίπτει στους περιορισμούς που απαιτείται να εμπίπτουν όλα τα ξένα κλειδιά.
- **BasicData και ObserveData:** Θεωρούμε για την συγκεκριμένη σχέση βλέποντας της από τη μεριά του πίνακα BasicData ότι επιτρέπεται ο πίνακας ObserveData να μη περιέχει εγγραφές που αφορούν τον συγκεκριμένο ασθενή, αν για κάποιο λόγο ο ιατρός δεν συμπληρώσει την καρτέλα «Παρακολούθηση ασθενή».Εξάλλου αυτή η καρτέλα μπορεί να συμπληρωθεί και αργότερα. Οι κλινικές εξετάσεις όμως μπορεί να είναι και περισσότερες από μία, για την ακρίβεια όσες και οι επισκέψεις του ασθενή. Η καταχώρηση κάθε μίας κλινικής εξέτασης γίνεται μία φορά και μετά επιτρέπονται μόνο διορθώσεις. Από την μεριά του πίνακα ObserveData δεν μπορεί μια καταχώρησή του να αντιστοιχεί σε περισσότερους από έναν ασθενή. Για κάθε ασθενή δηλαδή, μπορεί να υπάρχουν πολλές ή και καμία καταχώρηση της καρτέλας «Παρακολούθηση ασθενή»,για αυτόν τον λόγο η συγκεκριμένη σχέση είναι 1 προς 0 έως πολλά (1:N). Η σχέση αυτή υλοποιείται με την εισαγωγή του πρωτεύοντος κλειδιού του πίνακα BasicData σαν ξένο κλειδί στον πίνακα ObserveData. Το ξένο αυτό κλειδί εμπίπτει στους περιορισμούς που απαιτείται να εμπίπτουν όλα τα ξένα κλειδιά.
- **DiagnosisData και CoordinatesData:** Θεωρούμε για την συγκεκριμένη σχέση βλέποντας της από τη μεριά του πίνακα DiagnosisData ότι επιτρέπεται ο πίνακας CoordinatesData να μη περιέχει εγγραφές που αφορούν τον συγκεκριμένο ασθενή, αν για κάποιο λόγο ο ιατρός δεν συμπληρώσει σημεία. Εξάλλου αυτή η καρτέλα μπορεί να συμπληρωθεί και αργότερα. Οι κλινικές εξετάσεις όμως μπορεί να είναι και περισσότερες από μία, για την ακρίβεια όσες και οι επισκέψεις του ασθενή. Η καταχώρηση κάθε μίας κλινικής εξέτασης γίνεται μία φορά και μετά επιτρέπονται μόνο διορθώσεις. Από την μεριά του πίνακα CoordinatesData δεν μπορεί μια καταχώρησή του να αντιστοιχεί σε περισσότερους από έναν ασθενή. Για κάθε ασθενή δηλαδή, μπορεί να υπάρχουν πολλές ή και καμία καταχώρηση σημείων, για αυτόν τον λόγο η συγκεκριμένη σχέση είναι 1 προς 0 έως πολλά (1:N). Η σχέση αυτή υλοποιείται με την εισαγωγή του πρωτεύοντος κλειδιού του πίνακα BasicData σαν ξένο κλειδί στον πίνακα CoordinatesData. Το ξένο αυτό κλειδί εμπίπτει στους περιορισμούς που απαιτείται να εμπίπτουν όλα τα ξένα κλειδιά.

5.5.5 SQL queries



Εικόνα 35: Ερωτήματα SQL της Access

Στην βάση μας χρησιμοποιούμε δύο ειδών ερωτημάτων (queries). Ερωτήματα επιλογής (Select queries) και ερωτήματα ενοποίησης (Union queries). Συγκεκριμένα τα ερωτήματα είναι τα εξής:

- Query1. Πρόκειται για ερώτημα που απαιτεί την εισαγωγή της παραμέτρου id που αντιστοιχεί στο id του ασθενή. Επιστρέφει τις κλινικές εξετάσεις που έχουν καταχωρηθεί για το συγκεκριμένο id ασθενή.
- Query2. Πρόκειται για ερώτημα που απαιτεί την εισαγωγή της παραμέτρου id που αντιστοιχεί στο id του ασθενή. Επιστρέφει τους διαγνωστικούς ελέγχους που έχουν καταχωρηθεί για το συγκεκριμένο id ασθενή.
- Query3. Πρόκειται για ερώτημα που απαιτεί την εισαγωγή της παραμέτρου id που αντιστοιχεί στο id του ασθενή. Επιστρέφει τους παρακολουθήσεις που έχουν καταχωρηθεί για το συγκεκριμένο id ασθενή.
- QueryStatsAge. Πρόκειται για ερώτημα που επιστρέφει τις ηλικίες των ασθενών υπολογίζοντας τες αφαιρώντας την ημερομηνία γέννησης από την ημερομηνία που εκτελέστηκε το query.
- QueryStatsAge1. Πρόκειται για ερώτημα που επιστρέφει τον αριθμό συγκεκριμένων ορίων ηλικίας ασθενών που είναι καταχωρημένες στην βάση.
- queryStatsFamily. Πρόκειται για ερώτημα που επιτρέπει τον αριθμό των καταχωρήσεων που έχουν οικογενειακό ιστορικό καρκίνου του μαστού.

- queryStatsNosos. Πρόκειται για ερώτημα που επιστρέφει τον αριθμό κακοήθων και καλοηθών ευρημάτων των εξετάσεων που σημειώνονται για ασθενείς στην βάση.

5.6 Υλοποίηση κώδικα

Η υλοποίηση κώδικα αφορά την υλοποίηση όλων των συναρτήσεων , μεθόδων , κλάσεων που εκτελούνται κατά τις διάφορες λειτουργίες της εφαρμογής όπως πλοήγηση , reset , νέα καταχώρηση , αποθήκευση κτλ.

5.6.1 Δημιουργία κώδικα για interface.

Η υλοποίηση του κώδικα του interface της εφαρμογής περιλαμβάνει την δημιουργία κλάσεων και λειτουργιών για τη διαχείριση όλων των κομματιών που το αποτελούν όπως πλοήγηση, καθαρισμός πεδίων , ζωγραφική σε εικόνα, μηνύματα επιβεβαίωσης και άλλα.

Καθαρισμός πεδίων

Περιλαμβάνει τη δημιουργία των κλάσεων που καλούμε για να καθαρίσουμε τα πεδία από τιμές. Είναι σημαντικό για την αποφυγή καταχώρησης ανεπιθύμητων δεδομένων στη βάση μας τα πεδία να καθαρίζουν πριν από κάθε νέα καταχώρηση ή επιστροφή στην αρχική οθόνη της εφαρμογής. Για αυτό το λόγο δημιουργήσαμε τις functions ClearTab και resetgbx .

```

For Each ctl In TabToClear.Controls
    If ctl.GetType.Name = "GroupBox" Then
        Dim gbxCtrls As Object
        For Each gbxCtrls In ctl.controls
            If TypeOf gbxCtrls Is TextBox Then
                gbxCtrls.Text = ""
            ElseIf TypeOf gbxCtrls Is CheckBox Then
                gbxCtrls.checked = False
            ElseIf TypeOf gbxCtrls Is ComboBox Then
                gbxCtrls.SelectedIndex = -1
            End If
        Next
    End If
    If TypeOf ctl Is TextBox Then
        ctl.Text = ""
    ElseIf TypeOf ctl Is CheckBox Then
        ctl.checked = False
    ElseIf TypeOf ctl Is ComboBox Then
        ctl.SelectedIndex = -1
    End If
Next

```

Η Cleartab δέχεται σαν όρισμα το όνομα του tab page που θέλουμε να αδειάσουμε τα πεδία του. Σκανάρει κάθε control του tab page και αδειάζει τα πεδία του διαδοχικά. Αντίστοιχα η resetgbx δέχεται σαν όρισμα το όνομα του groupbox για τον καθαρισμό των πεδίων του. Η ανάγκη για τη δημιουργία ξεχωριστής function για καθαρισμό κάθε groupbox έγκειται στο γεγονός ότι στην εφαρμογή κάποια groupbox ενεργοποιούνται και απενεργοποιούνται ανάλογα με το εάν η τιμή κάποιου control είναι true ή false αντίστοιχα.

Μηνύματα επιβεβαίωσης και ενημέρωσης

Για την αποφυγή μη αναστρέψιμων συνεπειών από λανθασμένες ενέργειες χρηστών καθώς και για την παροχή ενημέρωσης στο χρήστη για τις ενέργειες που πρέπει να κάνει για να συνεχίσει, δημιουργήσαμε παράθυρα διαλόγου. Κάποια μηνύματα που εμφανίζονται όπως κατά τη διαγραφή καταχώρησης απαιτούν από το χρήστη επιβεβαίωση προκειμένου να συνεχίσει η ενέργεια. Ενδεικτικά παραθέτουμε μέρος των μηνυμάτων.

```

MsgBox("Η αποθήκευση αλλαγών για την καταχώρηση με ID " & rowID & " ολοκληρώθηκε με επιτυχία")
MsgBox("Καταχωρήστε τουλάχιστον ένα όνομα για να συνεχίσετε")
MsgBox("Οι παρακάτω καρτέλες δεν περιέχουν εγγραφές:" & Chr(13) & Chr(13) & NoRecords, , "")

```

```

MsgBox("Δεν μπορείτε να τοποθετησετε πανω απο 10 σημεια")
MsgBox("Το πεδίο 'Όνοματεπώνυμο' πρέπει να είναι συμπληρωμένο")
MessageBox.Show("Είστε σίγουρος ότι θέλετε να διαγράψετε την καταχώρηση με ID
" & rowID & ";", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2)
MessageBox.Show("Είστε σίγουρος ότι θέλετε να τερματίσετε το πρόγραμμα;", "",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) = DialogResult.No Then
e.Cancel = True

```

Εικόνα και εικονίδια

Περιλαμβάνει δημιουργία κλάσεων και λειτουργιών για τη δυνατότητα σχεδίασης σημείων επάνω στην εικόνα καθώς και το σχεδιασμό ενός combobox που περιλαμβάνει εικονίδια. Οι κλάσεις που αποτελούν όλο το σύστημα του σχεδιασμού πάνω στην **εικόνα είναι οι trckPinelo_Scroll, PictureBox2_MouseLeave, PictureBox2_MouseMove, PictureBox2_MouseDown, PictureBox2_Paint**, και **resetCoordinates**. Για την εμφάνιση του combobox με τα εικονίδια φτιάξαμε τις κλάσεις **cmbIkonidia_DrawItem** και **cmbIkonidia_MeasureItem**.

Η trckPinelo_Scroll διαχειρίζεται την scrollbar που καθορίζει το πάχος του πινέλου.

```
Me.Cursor = Cursors.Default
```

Η PictureBox2_MouseMove αλλάζει τον κέρσορα σε πινέλο όταν ο χρήστης περνάει με το ποντίκι του επάνω από την εικόνα.

```
Dim mycursor As New Icon(Application.StartupPath + "\brush.ico")
Me.Cursor = New Cursor(mycursor.Handle)
```

Η PictureBox2_MouseLeave επαναφέρει τον κέρσορα σε δείκτη όταν όταν ο χρήστης τον απομακρύνει από την εικόνα.

```

coord.X = e.X
coord.Y = e.Y
If i <= 9 Then
    coordlist(1, i) = coord.X
    coordlist(0, i) = coord.Y
    pinelo(i) = trckPinelo.Value
    PictureBox2.Invalidate()
    i = i + 1
Else
    MsgBox("Δεν μπορείτε να τοποθετησετε πανω απο 10 σημεια")
    i = 0
    PictureBox2.Invalidate()
End If

```

Η PictureBox2_MouseDown διαχειρίζεται το αριστερό κλικ του ποντικιού επάνω στην εικόνα όπου συλλέγει τις συντεταγμένες του κέρσορα πάνω στην εικόνα και καλεί το event της paint. Επίσης παρακολουθεί τον αριθμό των κλικς στην εικόνα και αποθηκεύει όλες τις συντεταγμένες μέχρι 10 σημείων σε ένα πίνακα συντεταγμένων.

```

Dim pen1 As New System.Drawing.Pen(Color.Red, 100)
For count As Integer = 0 To 9
    e.Graphics.DrawEllipse(Pens.Red, coordlist(1, count) - 15,
    coordlist(0, count) + 5, pinelo(count), pinelo(count))
    e.Graphics.FillEllipse(Brushes.Red, coordlist(1, count) - 15,
    coordlist(0, count) + 5, pinelo(count), pinelo(count))
Next

```

Στην PictureBox2_Paint συμβαίνει ο σχεδιασμός των σημείων και γραφική απεικόνιση τους στην οθόνη με βάση τις συντεταγμένες του ποντικιού όταν πατήθηκε το κουμπί του και της τιμής που είχε η scrollbar την δεδομένη στιγμή. Στην function resetCoordinates αδειάζουμε τις τιμές του πίνακα συντεταγμένων και καλούμε ξανα την paint για να ζωγραφίσει ξανα την εικόνα χωρίς τα σημεία.

Μενού , κουμπί γρήγορης εκκίνησης και περιήγησης

Περιλαμβάνει υλοποίηση κώδικα για την περιήγηση στις διάφορες οθόνες της εφαρμογής καθώς και την διαχείριση των στοιχείων που θα είναι εμφανή σε κάθε λειτουργία της. Οι κλάσεις που δημιουργήσαμε για την λειτουργία των κουμπιών γρήγορης εκκίνησης είναι οι **btnStats_Click**, **btnPrint_Click**, **btnSearch_Click**, **btnPlus_Click**, **btnExit_Click** και **btnHelp_Click** που αποσκοπούν στο να προβάλλουν τις διάφορες λειτουργίες του προγράμματος όπως την εμφάνιση της φόρμας στατιστικών και της φόρμας εκτύπωσης, την προβολή της λειτουργίας Νέος ασθενής και της λειτουργίας Υπάρχων ασθενής καθώς και την προβολή του εγχειριδίου χρήσης και τέλος την έξοδο του προγράμματος.

```

TabControlBasicData.SelectTab(TabPage2)
TabControlKartel.Visible = False
TabControlKartel.SelectTab(TabDisease)
GroupBoxClinicDate.Visible = False
GroupBoxDiagnosis.Visible = False
cmbYparxonOnoma.Visible = True
txtYparxonOnoma.Visible = False
gbxObserve.Visible = False
btnSaveAll.Visible = False
btnReturn2.Visible = False
ClearTab(TabDisease)
ClearTab(TabDanger)
ClearTab(TabClinical)
ClearTab(TabDiagnosis)
ClearTab(TabStadio)
ClearTab(TabObserve)
ClearTab(TabPage1)
PanelSearch.Visible = True

```

Κατά το event click του κουμπιού btnSearch επιλέγεται το Tabpage2 του TabcontrolBasicData και το TabControlKartel γίνεται μη εμφανή. Καλείται επίσης η ClearTab για να καθαρίσει τυχόν τιμές που έχουν ξεμεινεί στα πεδία των tabpages του TabcontrolCartel και TabcontrolBasicData

Η δημιουργία του μενού αποσκοπεί σε εναλλακτικό τρόπο πρόσβασης στις διάφορες λειτουργίες του προγράμματος και περιλαμβάνει τις μεθόδους ΕισαγωγήΝεουΑσθενουςToolStripMenuItem_Click, ΑναζήτησηΑσθενούςToolStripMenuItem_Click, ΘέματαΤηςΒοήθειαςToolStripMenuItem_Click, ΠροβολήΣτατιστικώνToolStripMenuItem_Click, ΈξοδοςToolStripMenuItem_Click και ΕκτύπωσηToolStripMenuItem_Click.

```

BtnSearch_Click(sender, New System.EventArgs())

```

Καθώς οι επιλογές του μενού έχουν την ίδια λειτουργία με τα κουμπία γρήγορης εκκίνησης στο click event του Menu Item Αναζήτηση Ασθενούς καλούμε το event click του αντίστοιχου κουμπιού.

Για την πλοήγηση στις καρτέλες των εξετάσεων του TabControlKartel δημιουργήσαμε 12 μεθόδους κουμπιών που αποσκοπούν στην μετάβαση σε επόμενη και προηγούμενη καρτέλα. Επίσης δημιουργήσαμε δύο μεθόδους κουμπιών τις btnReturn_Click και btnReturn2_Click για την επιστροφή του προγράμματος σε προηγούμενες οθόνες.

```

If TabControlKartel.SelectedIndex > 0 Then
TabControlKartel.SelectedIndex -= 1
End If

```

Στη μέθοδο που εκτελεί τη μετάβαση στην προηγούμενη καρτέλα γίνεται έλεγχος εάν το επιλεγμένο index του TabControlKartel είναι μεγαλύτερο του 0 και έπειτα επιλέγει το index που είναι ένα νούμερο μικρότερο από το επιλεγμένο.

5.6.2 Έλεγχοι εγκυρότητας δεδομένων και περιορισμοί.

Η παράγραφος αυτή αφορά την υλοποίηση των μεθόδων και των συναρτήσεων για τον έλεγχο δεδομένων και τους περιορισμούς της εφαρμογής ως προς την μορφή των δεδομένων που προκειται να εισαχθούν ή να ανακτηθούν. Για την εύρεση της καταχώρησης στο DataGridView όλων των

καταχωρήσεων που αντιστοιχεί στο όνομα που πληκτρολογείται στο πεδίο του combobox cmbYparxonOnoma δημιουργήσαμε την μέθοδο **cmbYparxonOnoma_KeyPress** η οποία αντιδρά στο event Keypress.

```
Dim findString As String = String.Empty
findString = e.KeyChar
With cmbYparxonOnoma
    .SelectedIndex = cmbYparxonOnoma.FindStringExact(findString)
End With
```

Η μέθοδος αυτή ενεργοποιείται κάθε φορά που πληκτρολογείται χαρακτήρας στο πεδίο τιμής του cmbYparxonOnoma. Αποθηκεύονται διαδοχικά όλοι οι χαρακτήρες στη μεταβλητή findString και μέσω της FindStringExact αναζητείται η λέξη στο index του combobox.

```
checkTxtName_NeosAsthenis
txtPhone_NeosAsthenis_Click
searchString
```

Καθώς το όνομα του ασθενούς είναι το πλέον απαραίτητο δεδομένο στην εφαρμογή για να υπάρξει καταχώρηση ασθενούς, ο χρήστης οφείλει να συμπληρώσει ένα όνομα για να ξεκλειδώσουν τα υπόλοιπα πεδία. Για τον σκοπό αυτό δημιουργήσαμε την μέθοδο **TxtName_ModifiedChanged** που λειτουργεί στο event ModifiedChanged (εάν πληκτρολογήθηκε κάποια τιμή στο πεδίο τιμών του TxtName_NeosAsthenis).

```
If TxtName_NeosAsthenis.Text = "" Then
    txtAge_NeosAsthenis.Enabled = False
    txtAddress_NeosAsthenis.Enabled = False
    txtPhone_NeosAsthenis.Enabled = False
    btnNewRecord.Enabled = False
    btnReset.Enabled = False
End If
```

Δημιουργήσαμε μια μέθοδο την **checkTxtName_NeosAsthenis** η οποία δέχεται σαν όρισμα μια μεταβλητή boolean την flag για τον έλεγχο του πεδίου text του control Txt_NeosAsthenis. Όταν το όρισμα της μεθόδου είναι true συνεπάγεται το πεδίο να μην είναι συμπληρωμένο οπότε προτρέπει τον χρήστη να συμπληρώσει το πεδίο για να ενεργοποιηθούν τα υπόλοιπα πεδία και να συνεχίσει με τη λειτουργία.

```
If flag = True Then
    MsgBox("Το πεδίο 'Όνοματεπώνυμο' πρέπει να είναι συμπληρωμένο")
End If
```

Για τον έλεγχο εγκυρότητας του ΑΜΚΑ δημιουργήσαμε μια συνάρτηση την **IsValidAMKA** η οποία δέχεται σαν όρισμα έναν αριθμό ΑΜΚΑ και στη συνέχεια προβαίνει στις απαραίτητες μαθηματικές πράξεις για τον έλεγχο της εγκυρότητας του..

```
Dim IsEven As Boolean, i%, iDigit%, iSum%, sTMP$, sChr$
For i = 1 To Len(AMKANumber)
    sChr = Mid(AMKANumber, i, 1)
    If IsNumeric(sChr) Then sTMP = sTMP & sChr
Next
If Len(sTMP) <> 11 Then Exit Function
For i = 1 To Len(sTMP)
    iDigit = Mid(sTMP, i, 1)
    If IsEven Then
        iDigit = iDigit * 2
    If iDigit > 9 Then iDigit = iDigit - 9
    End If
    iSum = iSum + iDigit
    IsEven = Not IsEven
Next
IsValidAMKA = iSum Mod 10 = 0
End Function
```

5.6.3 Καταχώρηση, αποθήκευση και διαγραφή δεδομένων από τη βάση

Αποθήκευση δεδομένων

Στην εφαρμογή μας μπορούμε να κάνουμε αποθήκευση όλων των αλλαγών που έχουμε κάνει. Για αυτόν το σκοπό δημιουργήσαμε την μέθοδο btnSaveAll_Click. Στο event click του κουμπιού btnSaveALL η μέθοδος αυτή καταχωρεί όλα τα πεδία όλων των καρτελών του tabControlKartel του tabpage ΥπάρχωνΑσθενής του TabControlBasicData στη βάση αντικαθιστώντας τις υπάρχοντες.

```
Dim sql As String = "SELECT id, name as [Όνοματεπώνυμο], phone as [Τηλέφωνο], address as [Διεύθυνση] , age as [Ηλικία] FROM BasicData ORDER BY id "
```

```
Dim da = New OleDb.OleDbDataAdapter(sql, con)
da.MissingSchemaAction = MissingSchemaAction.AddWithKey
Dim cb As New OleDb.OleDbCommandBuilder(da)
da.Fill(dsTabs, "BasicData")
dsTabs.Tables("BasicData").Rows.Find(rowID).Item(1)=txtYparxonOnoma.Text
If IsNumeric(txtAgeSearch.Text) = True Then
dsTabs.Tables("BasicData").Rows.Find(rowID).Item(4)=
CInt(txtAgeSearch.Text)
ElseIf txtAgeSearch.Text = "" Then
dsTabs.Tables("BasicData").Rows.Find(rowID).Item(4) = 0
Else
MsgBox("Το πεδίο ηλικία πρέπει να είναι αριθμός. Το πεδίο
ηλικία δεν άλλαξε")
End If
```

Στη μέθοδο εκτελείται ένα sql query προκειμένου να ταυτιστούν τα πεδία τιμών που πρόκειται να καταχωρηθούν με τα πεδία τιμών της βάσης. Στη συνέχεια γεμίζει ένας dataset με τον πίνακα BasicData και καταχωρούνται σε αυτόν οι καινούριες τιμές στη θέση rowID δηλαδή του ID της γραμμής που επιλέχθηκε από το αρχικό Datagridview και που αποτελεί την καταχώρηση ασθενούς. Γίνεται ένας έλεγχος εάν συγκεκριμένα πεδία είναι αριθμός ή κενά.

```
dsTabs.Tables("BasicData").Rows.Find(rowID).Item(2)=txtPhoneSearch.Text
dsTabs.Tables("BasicData").Rows.Find(rowID).Item(3)=txtAddressSearch.Text
da.Update(dsTabs, "BasicData")
con.Close()
FillDT()
con.Open()
dsTabs.Tables.Remove("BasicData")
```

Γίνεται ανανέωση του πίνακα και καλείται η FillDT για τον ενσυγχρονισμό των δεδομένων. Το dataset αδειάζει από τιμές του πίνακα BasicData.

```
FillDGVs(dgvClinical, "Query1", rowID)
FillDGVs(dgvDiagnosis, "Query2", rowID)
FillDGVs(dgvObserve, "Query3", rowID)
```

Η αποθήκευση ολοκληρώνεται και καλούνται οι FillDGVS για να ανανεώσουν τα datagridviews των εξετάσεων.

Διαγραφή Δεδομένων

Στην εφαρμογή παρέχουμε την δυνατότητα διαγραφής ολόκληρης καταχώρησης από τη βάση καθώς επίσης και τη μεμονωμένη διαγραφή των εξετάσεων ενός ασθενούς. Για την διαγραφή μιας καταχώρησης ασθενούς από τη βάση δεδομένων δημιουργήσαμε την μέθοδο **BtnDeleteR_Click** που λειτουργεί στο event click του κουμπιού BtnDeleteR. Η μέθοδος αυτή διαγράφει την καταχώρηση με βάση το rowID της καταχώρησης του DataGridView που είναι επιλεγμένη. Για αποφυγή μη αντιστρεψίμων ενεργειών, ο χρήστης έχει τη δυνατότητα να επιλέγει μία εγγραφή για διαγραφή κάθε φορά.

```
Dim test As Integer = DT.RowCount
dsDT.Tables(0).Rows.Find(rowID).Delete()
```

```

If DT.RowCount = rowIndex Then
  If DT.RowCount = 0 Then
    BtnDeleteR.Enabled = False
    btnViewAndProcess.Enabled = False
    ClearTab(TabPage2)
    lblID.Text = ""
    GoTo noROW
  Else
    rowIndex = rowIndex - 1
    DT.Rows(DT.RowCount - 1).Selected = True
  End If
End If

```

Η μέθοδος ελέγχει εάν η γραμμή που πρόκειται να διαγραφεί είναι η τελευταία και εάν είναι όντως η τελευταία γραμμή την διαγράφει και προσδιορίζει το rowIndex στην πρωτελευταία γραμμή και την επιλέγει.

```

rowID = DT.Rows(rowIndex).Cells(0).Value.ToString noROW:
con.Open()
Dim da As OleDb.OleDbDataAdapter
Dim sql As String = "SELECT id as [id], name as
[Όνοματεπώνυμο], phone as [Τηλέφωνο], address as [Διεύθυνση] FROM
BasicData"
da = New OleDb.OleDbDataAdapter(sql, con)
Dim cb As New OleDb.OleDbCommandBuilder(da)
da.Update(dsDT.Tables("BasicData"))
MsgBox("Η καταχώρηση διαγράφηκε με επιτυχία")

```

Προσδιορίζει το rowID στην νέα τελευταία γραμμή. Φορτώνεται ένα dataset με τα καινούρια δεδομένα και με την ανανέωση του θα καταχωρηθούν οι αλλαγές και στην βάση.

Για την διαγραφή των εξετάσεων ενός ασθενούς δημιουργήσαμε την μέθοδο **DeleteFromDGV** η οποία δέχεται σαν ορίσματα το dataGridView της εξέτασης που εναπόκειται να γίνει η διαγραφή, ένα string για τον προσδιορισμό του πίνακα της βάσης που αντιστοιχεί στην εξέταση και ένα string που αποτελεί ένα sql query δημιουργημένο στην access. Η μέθοδος αυτή διαγράφει την καταχώρηση της εξέτασης που επιλέχθηκε από τα DataGridViews των εξετάσεων.

```

Dim sql As String = "SELECT * FROM " & DGVrelatedTable & " WHERE id="
& rowID & ""
Dim DGVda = New OleDb.OleDbDataAdapter(sql, con)
DGVda.MissingSchemaAction = MissingSchemaAction.AddWithKey
DGVda.Fill(dsDGV)
dsDGV.Tables(0).Rows.Find(DGV.Rows(DGVRowIndex).Cells(1).Value.
ToString).Delete()
Dim cb As New OleDb.OleDbCommandBuilder(DGVda)
DGVda.Update(dsDGV.Tables(0))

```

Εκτελείται ένα sql query για να επιστραφούν τα δεδομένα του πίνακα της εξέτασης με το row ID ασθενούς. Εκτελείται διαγραφή της καταχώρησης από τον πίνακα. Φορτώνονται τα καινούρια δεδομένα σε ένα dataset και με την ανανέωση του θα καταχωρηθούν οι αλλαγές και στην βάση. Στη συνέχεια καλείται η FillDgvs που θα προβάλει το dataGridView με τα καινούρια δεδομένα.

Καταχώρηση δεδομένων

Για την καταχώρηση των δεδομένων στην βάση, δημιουργήσαμε 6 μεθόδους τις **SubmitDangersData, SubmitDiagnosisData, SubmitDeseasesData, SubmitStadioData, SubmitObserveData, και SubmitClinicalData**. Κάθε μέθοδος δέχεται σαν όρισμα το ID ασθενούς και εκτελεί την καταχώρηση τιμών των πεδίων της καρτέλας της εξέτασης που της έχει ανατεθεί στην βάση δεδομένων.

```

Dim dsClinicalData As New DataSet
Dim sql As String = "SELECT * FROM ClinicalData WHERE id=" & PatientID
& ""
Dim da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(dsClinicalData, "ClinicalData")

```

Επιλέγουμε μέσω sql query όλες τις καταχωρήσεις του πίνακα ClinicalData με το PatientID το ID ασθενούς που θέλουμε να καταχωρήσουμε την εξέταση και τις φορτώνουμε στο dataset dsClinicalData.

```
Dim regDate As Date = Date.Today
Dim dsNewRow As DataRow
dsNewRow = dsClinicalData.Tables("ClinicalData").NewRow()
dtpClinical.Value = New DateTime(dtpClinical.Value.Year,
dtpClinical.Value.Month, dtpClinical.Value.Day, DateTime.Now.Hour,
DateTime.Now.Minute, DateTime.Now.Second)
dsNewRow.Item("dateClinical") = dtpClinical.Value
```

Δημιουργούμε μια μεταβλητή τύπου date για την αποθήκευση της σημερινής ημερομηνίας της εξέτασης εάν δεν έχει επιλεγεί ημερομηνία από το datetimerpicker της εξέτασης. Προσαρμόζουμε την ημερομηνία που επιλέχθηκε στην μορφή που θέλουμε και την καταχωρούμε στο dataset.

```
da.Update(dsClinicalData, "ClinicalData")
FillDGVs(dgvClinical, "Query1", PatientID)
dgvClinical.ClearSelection()
dgvClinical.Rows(dgvClinical.RowCount - 1).Selected = True
```

Ανανεώνουμε τα δεδομένα στη βάση μας με τα καινούρια από το dataset και καλούμε την FillDGVs για την εκ νέου απεικόνισή τους στο datagridview της κλινικής εξέτασης. Καθαρίζουμε το datagridview από επιλεγμένες γραμμές και επιλέγουμε την τελευταία.

5.6.4 Προβολή δεδομένων

Για την προβολή των δεδομένων της βάσης επιλέξαμε controls όπως combobox και datagridviews. Τα δεδομένα επίσης προβάλλονται στα ίδια πεδία που γίνεται και η καταχώρηση τους καθώς επίσης σε μορφή στατιστικών και σε μορφή προεσκόπισης εκτύπωσης.

Datagridviews

Μέσω DataGridViews επιλέξαμε να προβάλλουμε τις καταχωρήσεις των ασθενών καθώς και των εξετάσεων κλινική εξέταση, διαγνωστικός έλεγχος και παρακολούθηση ασθενών. Δημιουργήσαμε μεθόδους ώστε να ανανεώνονται με δεδομένα από την βάση, και επιλέγοντας τις καταχωρήσεις που περιέχουν μπορεί να γίνει η προβολή ή η διαγραφή της καταχώρησης. Συγκεκριμένα για το DataGridView DT δημιουργήσαμε την μέθοδο FillDT με την οποία ο πίνακας γεμίζει με δεδομένα:

```
Dim sql As String = "SELECT id , name as [Όνοματεπώνυμο], phone as
[Τηλέφωνο], address as [Διεύθυνση] , age as [Ηλικία] FROM BasicData ORDER
BY id "
con.Open()
Dim da = New OleDb.OleDbDataAdapter(sql, con)
da.MissingSchemaAction = MissingSchemaAction.AddWithKey
da.Fill(dsDT, "BasicData")
DT.AutoGenerateColumns = True
DT.DataSource = dsDT.Tables(0)
con.Close()
```

Για την προβολή των δεδομένων στα datagridviews δημιουργήσαμε τις μεθόδους FillDT και FillDGVs. Η FillDT γεμίζει ένα πίνακα με τις όλες τις καταχωρήσεις των ασθενών και τις προβάλλει στο datagridview. Η FillDGVs γεμίζει ένα πίνακα με τις καταχωρήσεις των εξετάσεων και έπειτα προβάλλει τις ημερομηνίες τους μέσω των datagridviews των εξετάσεων. Οι dgvClinical, dgvDiagnosis και dgvObserve γεμίζουν με την ίδια μέθοδο στην οποία δώσαμε τρία ορίσματα τα οποία είναι το DGV για το όνομα του DataGridView που είναι να γεμίσει, το query το οποίο είναι το SQL ερώτημα που θα εκτελεστεί από τον αντάπτορα της σύνδεσης και το PatientID το οποίο είναι το id του ασθενούς του οποίου τα δεδομένα χρειαζόμαστε.

```
Dim dsDGVs As DataSet = New DataSet()
Dim cmd As New OleDb.OleDbCommand
con.Open()
cmd.CommandType = CommandType.StoredProcedure
```



```

cmd.Parameters.Add("@id", OleDb.OleDbType.VarChar).Value = PatientID
cmd.Connection = con
cmd.CommandText = query
Dim da As OleDb.OleDbDataAdapter = New OleDb.OleDbDataAdapter(cmd)
da.Fill(dsDGVs)
DGV.DataSource = dsDGVs.Tables(0)
DGV.Columns(1).Visible = False
DGV.AutoGenerateColumns = False
con.Close()

```

Το αντικείμενο του OleDb που εκτελείται είναι τύπου Stored Procedure καθώς τα queries που εκτελούνται βρίσκονται αποθηκευμένα στην βάση δεδομένων. Επειδή σε αυτά βάλαμε παραμέτρους, προσθέτουμε τις ίδιες παραμέτρους μέσω του αντικειμένου command cmd. Επίσης σε κάθε κλικ σε γραμμή του DT εκτελείται η DT_CellMouseDown η οποία εναρμονίζει όλα τα στοιχεία του interface με αυτά του επιλεγμένου ασθενή.

```

TabControlKartel.Visible = True
PanelSearch.Visible = False
TabControlKartel.SelectTab(TabDesease)
btnSaveAll.Visible = True
cmbYparxonOnoma.Visible = False
txtYparxonOnoma.Visible = True
btnReturn2.Visible = True
loadDataOnDoubleClick()

```

Combobox

Κάθε πάτημα πλήκτρου στο combobox cmbYparxonOnoma εκτελεί την μέθοδο cmbYparxonOnoma_KeyPress η οποία αναζητά στις ήδη υπάρχουσες καταχωρήσεις για ταύτιση του κειμένου που έχει πληκτρολογήθει. Εμφανίζει επίσης μία dropdown λίστα του combobox με τονισμένες της καταχωρήσεις που είναι πιο κοντινές στο κείμενο που πληκτρολογήθηκε.

```

Dim findString As String = String.Empty
findString = e.KeyChar
With cmbYparxonOnoma
    .SelectedIndex = cmbYparxonOnoma.FindStringExact(findString)
End With
cmbYparxonOnoma.DroppedDown = True

```

Σε περίπτωση που αλλάξουμε το περιεχόμενο του cmbYparxonOnoma από την dropdown λίστα του, εκτελείται η cmbYparxonOnoma_SelectedIndexChanged η οποία με βάση το id του ασθενή που επιλέχθηκε εναρμονίζει όλα τα στοιχεία του interface με αυτά του επιλεγμένου ασθενή.

```

Dim rowIndex1 As Integer
If cmbYparxonOnoma.SelectedIndex >= 0 Then
    rowIndex1 = cmbYparxonOnoma.SelectedIndex
    txtYparxonOnoma.Text =
    DT.Rows(rowIndex1).Cells("Όνοματεπώνυμο").Value.ToString
    lblID.Text = DT.Rows(rowIndex1).Cells("id").Value.ToString
    txtAddressSearch.Text =
    DT.Rows(rowIndex1).Cells("Διεύθυνση").Value.ToString
    txtPhoneSearch.Text =
    DT.Rows(rowIndex1).Cells("Τηλέφωνο").Value.ToString
    txtAgeSearch.Text = DT.Rows(rowIndex1).Cells("Ηλικία").Value.ToString
    DT.Rows(rowIndex1).Selected = True
End If

```

MsCharts

Για την εμφάνιση των στατιστικών δημιουργήσαμε την μέθοδο ReturnResultsByAgeTraffic. Τα ορίσματα της είναι ένα query το ερώτημα που θα εκτελεστεί από την βάση ,ένας πίνακας συντεταγμένων για την προβολή τους στο chart και ένα string που αποτελεί τον τίτλο του chart. Η μέθοδος αυτή εκτελεί ένα query και με τα δεδομένα που επιστρέφει γεμίζει το γράφημα των στατιστικών. Ταυτόχρονα ελέγχει εάν η εμφάνιση θα γίνεται με την μορφή στηλών ή πίτας.

```

Dim dsStats As DataSet = New DataSet()

```

```

Dim cmd As New OleDb.OleDbCommand
cmd.CommandType = CommandType.StoredProcedure
cmd.Connection = Form1.con
cmd.CommandText = query
Dim da As OleDb.OleDbDataAdapter = New OleDb.OleDbDataAdapter(cmd)
da.Fill(dsStats)

```

Δημιουργούμε ένα dataset και έπειτα εκτελούμε ένα query μέσω storedprocedure. Γεμίζουμε το dataset με τα αποτελέσματα που επέστρεψε το query.

```

Dim Yval(Pinakas.Length - 1) As Integer
For count As Integer = 0 To (Pinakas.Length - 1)
    Yval(count) = dsStats.Tables(0).Rows(count).Item(0)
Next
mscTrafficByAge.ChartAreas.Clear()
mscTrafficByAge.Series.Clear()
Dim area As New ChartArea("AREA")
mscTrafficByAge.ChartAreas.Add(area)
Dim series As Series = New Series(strTitle)
If cmbViewType.Text = "Πίτα" Then
    series.ChartType = SeriesChartType.Pie
End If
If cmbViewType.Text = "Στήλες" Then
    series.ChartType = SeriesChartType.Column
End If
For i As Integer = 0 To (Pinakas.Length - 1)
    series.Points.AddXY(Pinakas(i), Yval(i))
Next

```

Δημιουργούμε ένα πίνακα για τις τιμές του άξονα y. Καθορίζουμε τον τύπο του γραφήματος με βάση την τιμή του combobox. Δημιουργούμε τα σημεία του γραφήματος τοποθετώντας για κάθε εγγραφή του πίνακα yval την τιμή του πίνακα pinakas που της αντιστοιχεί.

Για τις επιλογές των πληροφοριών προς εμφάνιση στο γράφημα μας τοποθετήσαμε το combobox cmbStats.

```

Select Case cmbStats.SelectedIndex
Case 0
    Dim PinStatLabelsAge() As String = {"<25", "25 με 35", "35 με 45", "45 με 55", "55 με 65", "65 με 75", ">75"}
    Dim strTitle1 As String = "Συχνότητα προσέλευσης ανα ηλικιακή ομάδα"
    ReturnResultsByAgeTraffic("queryStatsAge1", PinStatLabelsAge, strTitle1)

```

Εάν το index του combobox είναι 0 δημιουργούμε ένα πίνακα από string για τον καθορισμό των ετικετών του άξονα y. Δημιουργούμε ένα string για τον τίτλο του γραφήματος και καλούμε την ReturnResults με αυτά τα ορίσματα και το query προκειμένου να απεικονίσουμε το γράφημα.

```

Case 1
    Dim PinStatLabelsFamilyhis() As String = {"καρκίνος μαστού ", "άλλων καρκίνων"}
    Dim strTitle2 As String = "Συχνότητα εμφάνισης οικογενειακού ιστορικού καρκίνου του μαστού και άλλων καρκίνων"
    ReturnResultsByAgeTraffic("queryStatsFamily", PinStatLabelsFamilyhis, strTitle2)

```

Ομοίως εάν το index του combobox είναι 1 δημιουργούμε ένα πίνακα από string για τον καθορισμό των ετικετών του άξονα y, ένα string για τον τίτλο του γραφήματος και καλούμε την ReturnResults με το αντίστοιχο query προκειμένου να απεικονίσουμε το γράφημα.

Print

Για την προεπισκόπηση της εκτύπωσης δημιουργήσαμε την φόρμα PrintForm. Η κλάση της περιέχει της εντολές και τις μεθόδους που μας δίνουν την δυνατότητα επιλογής των δεδομένων προς εκτύπωση καθώς και την μορφή τους. Η PrintForm περιέχει μία αναφορά (Crystal Report) που την

ονομάσαμε `PrintReport`, και η οποία χωρίζεται σε 6 υποαναφορές όσες δηλαδή και οι καρτέλες της εφαρμογής.

```
PrintReport1.Subreports.Item(2).RecordSelectionFormula = "{BasicData.id}=" &
Form1.rowID & ""
PrintReport1.Subreports.Item(1).RecordSelectionFormula = "{BasicData.id}=" &
Form1.rowID & ""
PrintReport1.Subreports.Item(0).RecordSelectionFormula = "{BasicData.id}=" &
Form1.rowID & ""
PrintReport1.Subreports.Item(3).RecordSelectionFormula = "{BasicData.id}=" &
Form1.rowID & ""
PrintReport1.Subreports.Item(4).RecordSelectionFormula = "{BasicData.id}=" &
Form1.rowID & ""
PrintReport1.Subreports.Item(5).RecordSelectionFormula = "{BasicData.id}=" &
Form1.rowID & ""
```

Ξεκινώντας την εκτέλεσή της η φόρμα φορτώνει τις υπό-αναφορές πάντα με βάση το `id` του ασθενή που έχουμε επιλέξει. Για την φόρτωση των δεδομένων των υποφορμών δημιουργήσαμε την μέθοδο **reportLoad** η οποία έχει δύο ορίσματα: `pinakas` το όνομα του πίνακα της βάσης από τον οποίο θέλουμε να φορτώσουμε τα δεδομένα, και `x` ο αριθμός που αντιστοιχεί στην υπό αναφορά της φόρμας.

```
Dim dsReport As New BCPDataSet
Dim Sql = "SELECT * FROM BasicData"
Dim daBasicData = New OleDb.OleDbDataAdapter(Sql, Form1.con)
daBasicData.MissingSchemaAction = MissingSchemaAction.AddWithKey
daBasicData.Fill(dsReport, "BasicData")
Sql = "SELECT * FROM " & pinakas & ""
Dim daKartela = New OleDb.OleDbDataAdapter(Sql, Form1.con)
daKartela.MissingSchemaAction = MissingSchemaAction.AddWithKey
daKartela.Fill(dsReport, pinakas)
PrintReport1.Subreports.Item(x).SetDataSource(dsReport)
```

Η φόρμα μας παρέχει την δυνατότητα μέσω `checkboxes` και `comboboxes` να επιλέξουμε συγκεκριμένες εξετάσεις προς εκτύπωση. Έτσι για εξέταση με μοναδική καταχώρηση πχ καρτέλα «Παρούσα νόσος» ελέγχουμε ένα επιλέχθηκε μέσω του `checkbox` για να την προβάλλουμε.

```
If chkDiseases.Checked = False Then
    PrintReport1.Section3.SectionFormat.EnableSuppress = True
    CrystalReportViewer1.RefreshReport()
Else
    PrintReport1.Section3.SectionFormat.EnableSuppress = False
    CrystalReportViewer1.RefreshReport()
End If
```

Ενώ για εξετάσεις με πολλαπλές καταχωρήσεις π.χ. «Κλινικές εξετάσεις» ελέγχουμε ποια εξέταση έχει επιλεγεί μέσω του `combobox` και την προβάλλουμε.

```
If cmbClinical.Text = "Καμία" Then
    PrintReport1.DetailSection3.SectionFormat.EnableSuppress = True
    CrystalReportViewer1.RefreshReport()
Else
    If PrintReport1.DetailSection3.SectionFormat.EnableSuppress =
    True Then
        PrintReport1.DetailSection3.SectionFormat.EnableSuppress = False
    End If
    Dim ClinicalID = Form1.dgvClinical.Rows(cmbClinical.SelectedIndex -
    1).Cells(1).Value.ToString()
    PrintReport1.Subreports.Item(0).RecordSelectionFormula =
    "{BasicData.id}=" & Form1.rowID & " AND {ClinicalData.ClinicalID}=" &
    ClinicalID & ""
    CrystalReportViewer1.RefreshReport()
End If
```

5.6.5 Αντίγραφο ασφαλείας

Για λόγους προστασίας των δεδομένων από κακή χρήση του προγράμματος ή καταστροφή του φυσικού χώρου που έχουν αποθηκευτεί, παρέχουμε τη δυνατότητα στο χρήστη μέσω του προγράμματος να δημιουργεί αντίγραφο ασφαλείας της βάσης δεδομένων. Γι' αυτό δημιουργήσαμε την φόρμα FormBackUp που ανοίγει στο πάτημα του κουμπιού συντόμευσης btnBck. Η φόρμα περιέχει 2 κουμπιά τα btnBackUp και btnRestore για τις λειτουργίες δημιουργία αντιγράφων και επαναφορά από αντίγραφο ασφαλείας αντίστοιχα.

```
Dim FileToCopy As String
Dim NewCopy As String
FileToCopy = System.AppDomain.CurrentDomain.BaseDirectory() & "BCP.mdb"
If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then
    NewCopy = FolderBrowserDialog1.SelectedPath & "\" &
    String.Format("{0:dd_MM_yyyy}", DateTime.Now) & "\BCP.mdb"
If Not System.IO.Directory.Exists(FolderBrowserDialog1.SelectedPath &
"\\" & String.Format("{0:dd_MM_yyyy}", DateTime.Now)) Then
System.IO.Directory.CreateDirectory(FolderBrowserDialog1.SelectedPath &
"\\" & String.Format("{0:dd_MM_yyyy}", DateTime.Now))
End If
If System.IO.File.Exists(FileToCopy) = True Then
System.IO.File.Copy(FileToCopy, NewCopy)
MsgBox("Το αρχείο αντιγράφηκε")
End If
End If
```

Κατά το πάτημα του κουμπιού btnBackUp δημιουργούνται 2 μεταβλητές string για την αποθήκευση των διευθύνσεων της βάσης δεδομένων και της διεύθυνσης που θα δημιουργηθεί το αντίγραφο ασφαλείας. Στην διεύθυνση που επιλέγει ο χρήστης μέσω του FolderBrowserDialog δημιουργείται ένας φάκελος που περιέχει το αντίγραφο της βάσης δεδομένων με όνομα την ημερομηνία που έγινε η ενέργεια. Ελέγχεται εάν είναι έγκυρη η διεύθυνση και έπειτα γίνεται η αντιγραφή και η ενημέρωση του χρήστη ότι το αρχείο αντιγράφηκε.

```
Dim FileToCopy As String
Dim NewCopy As String
If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then
FileToCopy = FolderBrowserDialog1.SelectedPath & "\BCP.mdb"
NewCopy = System.AppDomain.CurrentDomain.BaseDirectory() & "BCP.mdb"
If System.IO.File.Exists(FileToCopy) = True Then
System.IO.File.Copy(FileToCopy, NewCopy, True)
MsgBox("Η επαναφορά ολοκληρώθηκε")
End If
End If
```

Κατά το πάτημα του κουμπιού btnRestore δημιουργούνται 2 μεταβλητές string για την αποθήκευση των διευθύνσεων του αντιγράφου ασφαλείας της βάσης δεδομένων και της διεύθυνσης της βάσης δεδομένων που χρησιμοποιείται από το πρόγραμμα. Από τη διεύθυνση που επιλέγει ο χρήστης μέσω του FolderBrowserDialog γίνεται η αντιγραφή της βάσης δεδομένων και η αντικατάστασή της στη διεύθυνση του προγράμματος. Ελέγχεται εάν είναι έγκυρη η διεύθυνση του αντιγράφου και έπειτα γίνεται η αντιγραφή και η ενημέρωση του χρήστη ότι η επαναφορά της βάσης δεδομένων ολοκληρώθηκε.

5.6.6 Υλοποίηση κώδικα διασύνδεσης με τη βάση δεδομένων

Όπως έχουμε προαναφέρει, ένα αντικείμενο σύνδεσης (Connection Object) αναπαριστά μία φυσική διασύνδεση με την βάση δεδομένων. Για να δημιουργήσουμε ένα Connection Object χρειαζόμαστε να δηλώσουμε έναν πάροχο OLEDB (OLEDB Provider). Στην εφαρμογή μας χρησιμοποιούμε το αντικείμενο σύνδεσης όποτε θέλουμε να έχουμε πρόσβαση στην βάση δεδομένων μας. Στις πρώτες γραμμές του κώδικά μας, κατά τον ορισμό της κλάσης Form1, δημιουργούμε το αντικείμενο con που παράγεται από την κλάση OleDbConnection, και τις string μεταβλητές dbProvider και dbSource. Το αντικείμενο αυτό και οι μεταβλητές ανήκουν στα αντικείμενα και

μεταβλητές της Form1, οπότε καθ' όλη την διάρκεια εκτέλεσης του προγράμματός μας θα υπάρχει μόνο ένα στιγμιότυπο (instance) από αυτά.

```
Public con As New OleDb.OleDbConnection
Private dbProvider As String = "PROVIDER=Microsoft.Jet.OLEDB.4.0;"
Private dbSource As String = "Data Source = " &
System.AppDomain.CurrentDomain.BaseDirectory() & "BCP.mdb"
```

Στην μεταβλητή dbProvider αποδίδουμε την τιμή **PROVIDER=Microsoft.Jet.OLEDB.4.0** που είναι ο τρόπος σύνδεσης που θα χρησιμοποιήσουμε και στην μεταβλητή dbSource το μονοπάτι της βάσης μας. Η μέθοδος BaseDirectory() αντιστοιχεί στο μονοπάτι όπου βρίσκεται και το εκτελέσιμο αρχείο μας. Κατά την φόρτωση της φόρμας Form1 οπότε και εκτελείται η μέθοδος Form1_Load χρησιμοποιούμε την ιδιότητα ConnectionString του αντικειμένου con για να αποδώσουμε το μονοπάτι της βάσης μας ώστε να πραγματοποιηθεί η σύνδεσή μας.

```
con.ConnectionString = dbProvider & dbSource
```

Με αυτές τις εντολές επιτυγχάνουμε, με την εκτέλεση του προγράμματός μας, φόρτωσης της φόρμας μας και πριν ο χρήστης να αλληλεπιδράσει με την διεπαφή μας, να έχουμε δημιουργήσει την σύνδεσή μας και να έχουμε αποδώσει σε αυτήν τον τρόπο και το μονοπάτι σύνδεσης. Παρόλα αυτά όμως δεν έχουμε ακόμα πρόσβαση στα δεδομένα της βάσης αφού για να γίνει αυτό θα πρέπει να εκτελεστεί η μέθοδος con.Open().

Το παρατεταμένο άνοιγμα της σύνδεσης εγκυμονεί κινδύνους αλλοίωσης των δεδομένων και κατανάλωση πόρων του συστήματος γι' αυτό είναι προτιμότερο η σύνδεσή μας να ανοίγει όταν την χρειαζόμαστε και να κλείνει μέχρι την ανάγκη επαναχρησιμοποίησής της. Παρακάτω έχουμε μία τυπική χρησιμοποίηση της σύνδεσής μας όπου επιθυμούμε να απλά να φορτώσουμε κάποια δεδομένα από την βάση μας για να γεμίσουμε έναν πίνακα δεδομένων:

```
Dim sql As String = "SELECT id , name as [Όνοματεπώνυμο], fname as
[Πατρώνυμο] , phone as [Τηλέφωνο], address as [Διεύθυνση] , birthDate as
[Ημερομηνία γέννησης], AMKA as [AMKA] FROM BasicData ORDER BY id "
con.Open()
Dim da = New OleDb.OleDbDataAdapter(sql, con)
da.MissingSchemaAction = MissingSchemaAction.AddWithKey
da.Fill(dsDT, "BasicData")
DT.AutoGenerateColumns = True
DT.DataSource = dsDT.Tables(0)
con.Close()
```

Ανοίγουμε την σύνδεσή μας και κατόπιν δημιουργούμε έναν αντάπτορα δεδομένων ο οποίος μας παρέχει τα δεδομένα που προκύπτουν από το sql ερώτημα που θέσαμε ως όρισμα. Κατόπιν ενημερώνουμε τον αντάπτορα για την στήλη όπου περιέχεται το πρωτεύον κλειδί.

```
da.MissingSchemaAction = MissingSchemaAction.AddWithKey
```

Στην συνέχεια ο πίνακας δεδομένων γεμίζει από τα δεδομένα του dataset. Τα δεδομένα επιστρέφονται από ένα απλό SQL ερωτήματος, όμως στην εφαρμογή μας χρειαστήκαμε να λάβουμε δεδομένα μέσω ερωτήματος παραμέτρου (parameter SQL), όπου ένα από τα κριτήρια (στην προκειμένη περίπτωση το id) δεν είναι γνωστό εξ αρχής και εξαρτάται από τις επιλογές του χρήστη.

```
Dim cmd As New OleDb.OleDbCommand
con.Open()
cmd.CommandType = CommandType.StoredProcedure
cmd.Parameters.Add("@id", OleDb.OleDbType.VarChar).Value = PatientID
cmd.Connection = con
cmd.CommandText = query
Dim da As OleDb.OleDbDataAdapter = New OleDb.OleDbDataAdapter(cmd)
con.Close()
```

Δημιουργούμε μία εκτελέσιμη OleDb εντολή. Ορίζουμε ότι τα δεδομένα θα ληφθούν από την εκτέλεση ερωτήματος της βάσης. Ορίζουμε ότι το ερώτημα θα λάβει ως παράμετρο το ID του ασθενή.

Η σύνδεσή μας θα γίνει μέσω της διασύνδεσης con. Ο τύπος εντολής είναι query και έπειτα γίνεται εκτέλεση της εντολής.

5.7 Δημιουργία αρχείου εγκατάστασης

Ανοίγοντας το πρόγραμμα Inno Setup εμφανίζεται η επιλογή «create script file using wizard». Ακολουθώντας τον οδηγό αυτό μπορούμε να δημιουργήσουμε το βασικό μέρος του Script μας. Συμπληρώνοντας ζητούμενα όπως το εκτελέσιμο αρχείο μας, εικονίδια εφαρμογής και συμπεριλαμβανόμενα αρχεία, δημιουργείται το Script μας στο οποίο μπορούμε να προσθέσουμε κώδικα στην συνέχεια. Για να κάνουμε compile το αρχείο πατάμε στο κουμπί με το πράσινο βελάκι το αρχείο αποθηκεύεται στον φάκελο output στα έγγραφα.

```
[Files]
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio
2008\Projects\BreastCancerProject\bin\Debug\BreastCancerProject.exe";
DestDir: "{app}";
```

5.8 Διαδικασία ελέγχου

Μετά το πέρας της πρώτης φάση υλοποίησης της εφαρμογής μας η εγκατάστασή της δοκιμάστηκε σε διάφορα συστήματα με σκοπό να διαπιστωθούν τυχούσες ατέλειες και παραβλέψεις.

1) Σε διάφορες αναλύσεις οθόνης διαπιστώσαμε ότι σε μερικές από αυτές οι διαστάσεις των παραθύρων της εφαρμογής, δεν συμβάδιζαν με τους κανόνες μια εύχρηστη διεπαφής. Αποφασίσαμε να επιλέξουμε σταθερές και προκαθορισμένες διαστάσεις των παραθύρων της εφαρμογής μας με σκοπό να είναι εμφανή όλες οι επιλογές της εφαρμογής. Παράλληλα παραμείναμε πιστοί στο πλάνο, της όσον το δυνατόν πιστότερης αναπαράστασης του χαρτιού στην διεπαφή μας.

2) Σε παλιότερα συστήματα διαπιστώθηκε ότι υπήρχε απουσία του Net Framework 3.5 το οποίο και είναι απαραίτητο για την σωστή εκτέλεση της εφαρμογής μας. Το πρόβλημα αυτό διορθώθηκε με πρόβλεψη εγκατάστασής του (αν είναι απόν) κατά την εγκατάσταση της εφαρμογής μας (dependencies).

3) Όπως ήταν φυσικό τα CrystalReports που χρησιμοποιούμε για την εμφάνιση και εκτύπωση των αναφορών της εφαρμογής, έλειπαν από την πλειονότητα των συστημάτων. Και σε αυτή την περίπτωση έγινε πρόβλεψη εγκατάστασης των απαιτούμενων αρχείων κατά την διάρκεια εγκατάστασης της εφαρμογής μας.

6. Αποτελέσματα

Το πρόγραμμα παρακολούθησης ασθενών καρκίνου του μαστού BCP που δημιουργήθηκε στην πτυχιακή εργασία αυτή, αποτελεί ένα ικανοποιητικό αποτέλεσμα με βάση τους στόχους που τέθηκαν πρωταρχικά. Η δημιουργία του πραγματοποιήθηκε με τη χρήση της αντικειμενοστραφούς γλώσσας προγραμματισμού, Visual Basic, σε συνδυασμό με το σύστημα διαχείρισης βάσεων δεδομένων της Access. Παράλληλα για τις ανάγκες υλοποίησης απαιτήθηκε και η χρήση λογισμικού όπως Microsoft HTML Help Workshop, Photoshop, Ms Chart, και Crystal Reports. Επιπρόσθετα χρησιμοποιήθηκαν συμπληρωματικά οι γλώσσες SQL, HTML και Crystal Syntax. Το BCP αποτελεί ένα εύχρηστο, γρήγορο και αξιόπιστο σύστημα πλήρως λειτουργικό, το οποίο μπορεί να χρησιμοποιηθεί από οποιοδήποτε ιατρείο καρκίνου του μαστού και από οποιονδήποτε χρήστη, ανεξαρτήτως των γνώσεών του σε πληροφοριακά συστήματα.

6.1 Συμπεράσματα

Στην πτυχιακή εργασία αυτή χρησιμοποιήθηκε η αντικειμενοστραφής γλώσσα προγραμματισμού Visual Basic, η οποία είναι μια απλή γλώσσα ειδικά ως προς τον εκτέλεση κώδικα. Ταυτόχρονα υποστηρίζει την εύκολη δημιουργία διεπαφών και την σύνδεσή τους με μεθόδους που υποστηρίζονται από τις εφαρμογές. Η διασύνδεση μια εφαρμογής με υλοποιημένη σε Visual Basic είναι εύκολη και αξιόπιστη και μπορεί να παρέχει την μεταβολή των δεδομένων της βάσης με την χρήση της SQL που είναι η κατεξοχήν γλώσσα διαχείρισης βάσεων δεδομένων. Η κύρια πλατφόρμα ανάπτυξης της Visual Basic είναι το Microsoft Visual Studio που είναι ένα IDE που παρέχει συνοδευτικά τα εργαλεία για την ανάπτυξη της εφαρμογής μας όπως είναι τα Crystal Reports για την δημιουργία των εκτυπώσεων.

Η βάση δεδομένων που χρησιμοποιήσαμε είναι η Microsoft Access, η οποία είναι ένα σύστημα διαχείρισης βάσεων δεδομένων που μπορεί εύκολα να αλληλεπιδράσει με λογισμικό βασισμένο σε συστήματα Windows. Είναι επίσης φιλική ως προς το πακέτο ανάπτυξης λογισμικού .NET Framework τις βιβλιοθήκες του οποίου χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής μας. Είναι το πιο δημοφιλές σύστημα, για αποθήκευση και διαχείριση δεδομένων του αριθμού που χρειάζεται ένα ιατρείο καρκίνου του μαστού. Μία βάση δεδομένων Access μπορεί να αποθηκεύσει έως και 2 gigabyte δεδομένων.

Το .NET Framework είναι ουσιαστικά μια πλατφόρμα ανάπτυξης αντικειμενοστραφών εφαρμογών που προσφέρει μια εκτεταμένη βιβλιοθήκη κλάσεων. Το χαρακτηριστικό της είναι ότι τα προγράμματα που φτιάχνει κάποιος μεταγλωττίζονται σε μια ενδιάμεση γλώσσα (όχι γλώσσα μηχανής αλλά ούτε και bytecode όπως στη java) και μετατρέπεται σε γλώσσα μηχανής κατά την πρώτη εκτέλεση. Η ενδιάμεση αυτή γλώσσα επιτρέπει σε εφαρμογές που είναι γραμμένες σε διαφορετικές γλώσσες να επικοινωνούν άμεσα μεταξύ τους γιατί πρακτικά παράγουν εκτελέσιμα που υπακούουν στο CLI (Common Language Infrastructure) & για την εκτέλεσή τους απαιτούν μόνο το (CLR) Common Language Runtime.

Το σύστημα είναι εύχρηστο, ελαφρύ, γρήγορο και συμβατό με όλες τις εκδόσεις Windows. Παράλληλα διατηρεί την όσο ο δυνατόν πιο πιστή αναπαράσταση των φύλλων χαρτιού που χρησιμοποιούταν για την συγκέντρωση δεδομένων. Τα φύλλα αυτά είναι ακολουθούν πρότυπα που χρησιμοποιούν όλα τα ιατρεία στην Ελλάδα, οπότε η αντικατάσταση λήψης δεδομένων από το πρόγραμμά μας γίνεται μέσω μιας οικίας διεπαφής.

6.2 Χρήση

Η εφαρμογή που αναπτύχθηκε, δίνει τη δυνατότητα σε έναν χρήστη να καταγράψει, να αποθηκεύσει και να επεξεργαστεί δεδομένα ασθενών που λαμβάνονται από αυτούς, είτε με την χρήση ερωτηματολογίου είτε με την εξέτασή τους. Δίνει παράλληλα την δυνατότητα διόρθωσης ή συμπλήρωσης δεδομένων παλιότερων επισκέψεων. Παρέχεται ακόμη δυνατότητα εξαγωγής στατιστικών που ήταν αδύνατη με την χρήση χαρτιού. Ο χρήστης όμως μπορεί ακόμη να λάβει δεδομένα σε εκτυπωμένη μορφή και δεν χρειάζεται να έχει ιδιαίτερες γνώσεις χρήσης υπολογιστών πέρα από τις βασικές. Ακόμη, το πρόγραμμά μας εμπεριέχει ανίχνευση λαθών, (προειδοποίηση λάθους αριθμού ψηφίων, λάθος ΑΜΚΑ κτλ) που η μέθοδος του ερωτηματολογίου δεν μπορεί να ανιχνεύσει.

6.3 Μελλοντική εργασία και επεκτάσεις

Η πτυχιακή εργασία αυτή θα μπορούσε μελλοντικά να έχει τις ακόλουθες επεκτάσεις:

- 1) Να γίνει χρήση των σύγχρονων τεχνολογιών ώστε μέσω scanner να καταχωρούνται απευθείας τα στοιχεία και απλά να επαληθεύονται από τον χρήστη.
- 2) Επανασχεδίαση του προγράμματος με τεχνικές multi-threading ώστε να εκμεταλλεύεται στο έπακρο την τεχνολογία των σύγχρονων επεξεργαστών.
- 3) Να γίνει η μετάφραση του στα Αγγλικά και σε άλλες γλώσσες.
- 4) Προσαρμογή του προγράμματος σε διάφορες αναλύσεις κατά την εγκατάσταση του προγράμματος και λειτουργία επιλογών απεικόνισης όπως μέγεθος και χρώμα γραμματοσειράς.
- 5) Επανασχεδίαση του προγράμματος για λειτουργία σε server με ταυτόχρονη πολλαπλή πρόσβαση από απομακρυσμένες θέσεις καθώς και χρήση πρωτοκόλλων αυξημένης ασφαλείας για την μεταφορά των ευαίσθητων ιατρικών δεδομένων. Προϋποθέτει δημιουργία λογαριασμών με αντίστοιχα δικαιώματα πρόσβασης ανάλογα με την ειδικότητα του κάθε χρήστη.
- 6) Επιπλέον λειτουργία για απλοποιημένη δημιουργία SQL queries μέσω γραφικού περιβάλλοντος για την επιστροφή των δεδομένων κατά την εξαγωγή των στατιστικών.
- 7) Επιπλέον δυνατότητα απεικόνισης των ευρημάτων του διαγνωστικού ελέγχου σε 3 διαστάσεις.

Βιβλιογραφία

- [1] Μανιάτης Πέτρος, Δημιουργία ιατρικής βάσης δεδομένων με την βοήθεια της Microsoft Access (2006).
- [2] Dan Appleman, Visual Basic Programmer's Guide to the Win32 API (2009)
- [3] Paul Deitel, H.M Deitel, and G. J. Ayer, Simply Visual Basic 2008 (2010)
- [4] Dan Appleman, Visual Basic Programmer's Guide to the Win32 API (2009)
- [5] Joyce Cox, Joan Lambert, Microsoft Access 2010 Βήμα βήμα (2011)
- [6] Αθανάσιος Σπυριδάκος, Αντικειμενοστραφής προγραμματισμός σε περιβάλλον Visual Basic.NET (2008)
- [7] Εμμανουήλ Σκορδαλάκης. Εισαγωγή στην Τεχνολογία Λογισμικού. Συμμετρία (1991)

Πηγές από το διαδίκτυο

- [1] <http://www.office.microsoft.com>
- [2] <http://www.databases.about.com/>
- [3] <http://www.vbtutor.net>
- [4] <http://www.vb6.us>
- [5] <http://www.crystalreports.com>
- [6] <http://www.businessobjects.com>
- [7] <http://www.ceid.upatras.gr/>
- [8] <http://www.w3counter.com/>
- [9] <http://office.microsoft.com>
- [10] <http://www.dmst.aueb.gr/>
- [11] <http://www.wikipedia.org>
- [12] <http://www.dotnetperls.com>
- [13] <http://www.codeproject.com>
- [14] <http://www.processing.org>
- [15] <http://www.jrssoftware.org>

ΠΑΡΑΡΤΗΜΑΤΑ

Παράρτημα Α – Πηγαίος κώδικας

Form1

```
Imports System.ComponentModel
Public Class Form1

    Public con As New OleDb.OleDbConnection
    Private dbProvider As String = "PROVIDER=Microsoft.Jet.OLEDB.4.0;"
    Private dbSource As String = "Data Source = " &
System.AppDomain.CurrentDomain.BaseDirectory() & "BCP.mdb"
    Private dsDT As New DataSet
    Public rowID As String
    Private rowIndex As Integer
    Private coordlist(1, 9) As Integer
    Private coord As Point = MousePosition
    Private pinelo(9) As Integer
    Private i As Integer = 0
    Private NoRecords As String

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        con.ConnectionString = dbProvider & dbSource
        FillDT()
        If DT.RowCount = 0 Then
            btnViewAndProccess.Enabled = False
            BtnDeleteR.Enabled = False
        Else
            rowID = DT.Rows(0).Cells(0).Value.ToString
        End If
        Dim ikonidiaText() As String = {"Συμπαγέζ ", "Κυστικό", "Διήθηση",
"Οζίδια ", "Εισολκή", "Ουλή"}
        Dim items(Me.ImageList1.Images.Count - 1) As String
        For i As Int32 = 0 To Me.ImageList1.Images.Count - 1
            items(i) = ikonidiaText(i)
        Next
        Me.cmbIkonidia.Items.AddRange(items)
        Me.cmbIkonidia.DropDownStyle = ComboBoxStyle.DropDownList
        Me.cmbIkonidia.DrawMode = DrawMode.OwnerDrawVariable
        Me.cmbIkonidia.ItemHeight = Me.ImageList1.ImageSize.Height
        Me.cmbIkonidia.Width = Me.ImageList1.ImageSize.Width + 75
        Me.cmbIkonidia.MaxDropDownItems = Me.ImageList1.Images.Count
    End Sub

    Private Sub cmbIkonidia_DrawItem(ByVal sender As Object, ByVal e As
System.Windows.Forms.DrawItemEventArgs) Handles cmbIkonidia.DrawItem
        If e.Index <> -1 Then
            e.Graphics.DrawImage(Me.ImageList1.Images(e.Index),
e.Bounds.Left, e.Bounds.Top)
            e.Graphics.DrawString(cmbIkonidia.Items(e.Index),
cmbIkonidia.Font, System.Drawing.Brushes.Black, e.Bounds.Left + 30,
e.Bounds.Top + 10)
        End If
    End Sub
```

```

Private Sub cmbIkonidia_MeasureItem(ByVal sender As Object, ByVal e As
System.Windows.Forms.MeasureItemEventArgs) Handles cmbIkonidia.MeasureItem
    e.ItemHeight = Me.ImageList1.ImageSize.Height
    e.ItemWidth = Me.ImageList1.ImageSize.Width
End Sub

'για να δείξει το data table στο panelSearch
Private Sub FillDT()
    Dim sql As String = "SELECT id , name as [Όνοματεπώνυμο], fname as
[Πατρώνυμο] ,phone as [Τηλέφωνο], address as [Διεύθυνση] , birthDate as
[Ημερομηνία γέννησης],AMKA as [AMKA] FROM BasicData ORDER BY id "
    con.Open()
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    ' Ενημερώνει το dsDT για την στήλη που περιέχει το πρωτεύον κλειδί
    da.MissingSchemaAction = MissingSchemaAction.AddWithKey
    da.Fill(dsDT, "BasicData")
    DT.AutoGenerateColumns = True
    DT.DataSource = dsDT.Tables(0)
    con.Close()
End Sub

Public Sub FillDGVs(ByVal DGV As DataGridView, ByVal query As String,
ByVal PatientID As String)
    Dim dsDGVs As DataSet = New DataSet()
    Dim cmd As New OleDb.OleDbCommand
    con.Open()
    'storedprecedure epeidi ta querys einai idi apothikeumena stin vasi
    cmd.CommandType = CommandType.StoredProcedure
    'vazoume parametrous epeidi valame kai sto query tis vasis
parametro
    cmd.Parameters.Add("@id", OleDb.OleDbType.VarChar).Value =
PatientID
    cmd.Connection = con
    cmd.CommandText = query
    Dim da As OleDb.OleDbDataAdapter = New OleDb.OleDbDataAdapter(cmd)
    da.Fill(dsDGVs)
    DGV.DataSource = dsDGVs.Tables(0)
    DGV.Columns(1).Visible = False
    DGV.AutoGenerateColumns = False
    con.Close()
    If DGV.RowCount >= 1 Then '>1 Γιατί έτσι κι αλλιώς η μία εγγραφή
επιλέγεται.
        DGV.CurrentCell = DGV.Item(0, DGV.RowCount - 1)
    End If
End Sub

Private Sub DT_CellMouseDoubleClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
DT.CellMouseDoubleClick
    TabControlKartel.Visible = True
    PanelSearch.Visible = False
    TabControlKartel.SelectTab(TabDesease)
    btnSaveAll.Visible = True
    cmbYparxonOnoma.Visible = False
    txtYparxonOnoma.Visible = True
    btnReturn2.Visible = True
    loadDataOnDoubleClick()
End Sub

Public Sub loadComboData()

```

```

        cmbYparxonOnoma.DataSource = New BindingSource(DT.DataSource,
Nothing)
        'cmbYparxonOnoma.DataSource = dsDT.Tables(0)
        cmbYparxonOnoma.DisplayMember = "Όνοματεπώνυμο"
    End Sub

'event gia na pianei to gramma pou pliktrologithike
    Private Sub cmbYparxonOnoma_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles cmbYparxonOnoma.KeyPress
        Dim findString As String = String.Empty
        findString = e.KeyChar
        With cmbYparxonOnoma
            ' otan patitheï koumpi mesa sto combobox psaxnei tis
kataxoriseis na vrei akrivos to idio onoma
            .SelectedIndex = cmbYparxonOnoma.FindStringExact(findString)
        End With
        cmbYparxonOnoma.DroppedDown = True
    End Sub

Private Sub cmbYparxonOnoma_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbYparxonOnoma.SelectedIndexChanged
    ' elegxos gia na min krasarei otan den vriksei kataxorisi sto row 1
    Dim rowIndex1 As Integer
    If cmbYparxonOnoma.SelectedIndex >= 0 Then
        rowIndex1 = cmbYparxonOnoma.SelectedIndex
        txtYparxonOnoma.Text =
DT.Rows(rowIndex1).Cells("Όνοματεπώνυμο").Value.ToString
        lblID.Text = DT.Rows(rowIndex1).Cells("id").Value.ToString
        txtAddressSearch.Text =
DT.Rows(rowIndex1).Cells("Διεύθυνση").Value.ToString
        txtPhoneSearch.Text =
DT.Rows(rowIndex1).Cells("Τηλέφωνο").Value.ToString
        txtAgeSearch.Text = DT.Rows(rowIndex1).Cells("Ημερομηνία
Γέννησης").Value.ToString
        txtFnameSearch.Text =
DT.Rows(rowIndex1).Cells("Πατρώνυμο").Value.ToString
        txtAMKASearch.Text =
DT.Rows(rowIndex1).Cells("AMKA").Value.ToString
        Dim cmd As New OleDb.OleDbCommand
        'con.Open()
        'storedprecedure epeidi ta queries einai idi apothikeumena stin
vasi
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Parameters.Add("@id", OleDb.OleDbType.VarChar).Value =
rowID
        'vazoume parametrous epeidi valame kai sto query tis vasis
parametro
        cmd.Connection = con
        cmd.CommandText = "AgeQ"
        Dim dsAge As New DataSet
        Dim da As OleDb.OleDbDataAdapter = New
OleDb.OleDbDataAdapter(cmd)
        da.Fill(dsAge)
        lblAgeS.Text = dsAge.Tables(0).Rows(0).Item(0).ToString
        'con.Close()
        DT.Rows(rowIndex1).Selected = True
    End If
End Sub

```

```

Private Sub BtnDeleteR_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnDeleteR.Click
    Dim MSGResult As DialogResult = MessageBox.Show("Είστε σίγουρος ότι
θέλετε να διαγράψετε την καταχώρηση με ID " & rowID & ";", "",
MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2)
    If MSGResult = DialogResult.Yes Then
        Dim test As Integer = DT.RowCount
        dsDT.Tables(0).Rows.Find(rowID).Delete()
        If DT.RowCount = rowIndex Then 'IF που τρέχει όταν η προς
διαγραφή γραμμή είναι η τελευταία.
            If DT.RowCount = 0 Then
                BtnDeleteR.Enabled = False
                btnViewAndProcess.Enabled = False
                ClearTab(TabPage2)
                lblID.Text = ""
                GoTo noROW
            Else
                rowIndex = rowIndex - 1 'προσδιορίζει το rowIndex στην
νέα τελευταία γραμμή.
                DT.Rows(DT.RowCount - 1).Selected = True 'Επιλέγει την
νέα τελευταία γραμμή.
            End If
        End If
    End If

    rowID = DT.Rows(rowIndex).Cells(0).Value.ToString 'προσδιορίζει
το rowID στην νέα τελευταία γραμμή.
noROW:
    con.Open()
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String = "SELECT id as [id], name as
[Όνοματεπώνυμο], phone as [Τηλέφωνο], address as [Διεύθυνση] FROM BasicData"
    da = New OleDb.OleDbDataAdapter(sql, con)
    Dim cb As New OleDb.OleDbCommandBuilder(da) 'Εντολή η οποία
επιτρέπει τον da να εκτελεί εντολές μεταξύ των οποίων και την delete.
    da.Update(dsDT.Tables("BasicData")) 'Με την ανανέωση του
DataSet θα καταχωρηθούν οι αλλαγές και στην βάση.
    MsgBox("Η καταχώρηση διαγράφηκε με επιτυχία")
    con.Close()
End If
End Sub

Private Sub DeleteFromDGV(ByVal DGV As DataGridView, ByVal DGVrelatedTable
As String, ByVal query As String)
    If DGV.RowCount <> 0 Then
        Dim DGVRowIndex = DGV.SelectedRows(0).Index
        Dim MSGResult As DialogResult = MessageBox.Show("Είστε σίγουρος
ότι θέλετε να διαγράψετε την καταχώρηση με ID " &
DGV.Rows(DGVRowIndex).Cells(0).Value.ToString & ";", "",
MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2)
        If MSGResult = DialogResult.Yes Then
            con.Open()
            Dim dsDGV As New DataSet
            Dim sql As String = "SELECT * FROM " & DGVrelatedTable & "
WHERE id=" & rowID & ""
            Dim DGVda = New OleDb.OleDbDataAdapter(sql, con)
            DGVda.MissingSchemaAction = MissingSchemaAction.AddWithKey
            DGVda.Fill(dsDGV)
            dsDGV.Tables(0).Rows.Find(DGV.Rows(DGVRowIndex).Cells(1).Value.ToString).De
lete()
        End If
    End If
End Sub

```

```

        If DGVRowIndex = DGV.RowCount - 1 Then 'IF που τρέχει όταν
η προς διαγραφή γραμμή είναι η τελευταία.
        DGVRowIndex = DGVRowIndex - 1 'Προσδιορίζει το rowIndex
στην νέα τελευταία γραμμή.
        End If
        Dim cb As New OleDb.OleDbCommandBuilder(DGVda) 'Εντολή η
οποία επιτρέπει τον da να εκτελεί εντολές μεταξύ των οποίων και την delete.
        DGVda.Update(dsDGV.Tables(0)) 'Με την ανανέωση του DataSet
θα καταχωρηθούν οι αλλαγές και στην βάση.

        MsgBox("Η καταχώρηση διαγράφηκε με επιτυχία")
        con.Close()
    End If
    FillDGVs(DGV, query, rowID)
    If DGV.RowCount <> 0 Then 'Η if αυτή ελέγχει αν η διαγραφίσα
καταχώρηση είναι η τελευταία και επιλέγει την αμέσως προηγούμενη.
        DGV.CurrentCell = DGV.Item(0, DGVRowIndex)
        Select Case DGVrelatedTable
            Case "Clinicaldata"
RetrieveClinicalData(DGV.SelectedRows(0).Cells(1).Value)
            Case "DiagnosisData"
RetrieveDiagnosisData(DGV.SelectedRows(0).Cells(1).Value)
            Case "CoordinatesData"
RetrieveCoordinatesData(DGV.SelectedRows(0).Cells(1).Value)
            Case "ObserveData"
RetrieveObserveData(DGV.SelectedRows(0).Cells(1).Value)
        End Select
    Else
        Select Case DGVrelatedTable
            Case "ClinicalData"
                ClearTab(TabClinical)
            Case "DiagnosisData"
                ClearTab(TabDiagnosis)
                resetCoordinates()
            Case "ObserveData"
                ClearTab(TabObserve)
        End Select
    End If
Else
    MsgBox("Δεν υπάρχει καταχώρηση για διαγραφή")
End If
End Sub

Private Sub btnSaveAll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSaveAll.Click
    Dim dsTabs As New DataSet 'to new arxikopoiei ta dataset
    con.Open()
    Dim sql As String = "SELECT id , name as [Όνοματεπώνυμο], fname as
[Πατρώνυμο] , phone as [Τηλέφωνο], address as [Διεύθυνση] , birthDate as
[Ημερομηνία γέννησης], AMKA as [AMKA] FROM BasicData ORDER BY id"
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.MissingSchemaAction = MissingSchemaAction.AddWithKey
    Dim cb As New OleDb.OleDbCommandBuilder(da)
    da.Fill(dsTabs, "BasicData")
    dsTabs.Tables("BasicData").Rows.Find(rowID).Item(1) =
txtYparxonOnoma.Text
    If IsValidAMKA(txtAMKASearch.Text) = True Then
        dsTabs.Tables("BasicData").Rows.Find(rowID).Item("AMKA") =
txtAMKASearch.Text
        txtAgeSearch.Text = getDatefromAMKA(txtAMKASearch.Text)
    End If
End Sub

```

```

        If IsNumeric(searchString(txtAMKA_NeosAsthenis.Text, "AMKA"))
Then
        MsgBox("Ο αριθμός Α.Μ.Κ.Α που πληκτρολογήσατε έχει ήδη
καταχωρηθεί σε ασθενή")
        End If
    Else
        MsgBox("Ο αριθμός Α.Μ.Κ.Α που πληκτρολογήσατε δεν είναι έγκυρος.
Ο αριθμός Α.Μ.Κ.Α δεν καταχωρήθηκε")
        End If
        If IsDate(txtAgeSearch.Text) = True Then
            dsTabs.Tables("BasicData").Rows.Find(rowID).Item("Ημερομηνία
γέννησης") = txtAgeSearch.Text
        Else
            MsgBox("Το πεδίο Ημερομηνία γέννησης πρέπει να είναι σε μορφή
*/*/*. Το πεδίο Ημερομηνία γέννησης δεν άλλαξε")
            End If
            dsTabs.Tables("BasicData").Rows.Find(rowID).Item("Τηλέφωνο") =
txtPhoneSearch.Text
            dsTabs.Tables("BasicData").Rows.Find(rowID).Item("Διεύθυνση") =
txtAddressSearch.Text
            dsTabs.Tables("BasicData").Rows.Find(rowID).Item("Πατρώνυμο") =
txtFnameSearch.Text
            da.Update(dsTabs, "BasicData")
            con.Close()
            FillDT()
            con.Open()
            dsTabs.Tables.Remove("BasicData")
            'επεξεργασία στοιχείων παρούσας νοσου
            sql = "SELECT * FROM DeseasesData WHERE id=" & rowID & ""
            Dim daDeseasesData = New OleDb.OleDbDataAdapter(sql, con)
            daDeseasesData.MissingSchemaAction = MissingSchemaAction.AddWithKey
            daDeseasesData.Fill(dsTabs, "DeseasesData")
            Dim cbDeseasesData As New OleDb.OleDbCommandBuilder(daDeseasesData)
            If dsTabs.Tables(0).Rows.Count = 0 Then
                con.Close()
                SubmitDeseasesData(rowID)
                con.Open()
            Else
                dsTabs.Tables(0).Rows.Find(rowID).Item("asymptomatikiri") =
chkAsymptomatikiri.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("Symptomatikiri") =
chkSymptomatikiri.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("MasxaliaiesL") =
chkMasxaliaiesL.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("MasxaliaiesR") =
chkMasxaliaiesR.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("AllaL") =
chkAllaL.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("AllaR") =
chkAllaR.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("Ponoskiri") =
chkPonoskiri.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("PonoskiL") =
chkPonoskiL.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiL")
= chkDermatikiparamorfosiL.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiR")
= chkDermatikiparamorfosiR.Checked
                dsTabs.Tables(0).Rows.Find(rowID).Item("PonosmikiL") =
chkPonosmikiL.Checked

```

```

        dsTabs.Tables(0).Rows.Find(rowID).Item("PonosmikiR") =
chkPonosmikiR.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("AnomaliathilisR") =
chkAnomaliathilisR.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("AnomaliathilisL") =
chkAnomaliathilisL.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("MiaimatiroL") =
chkMiaimatiroL.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("MiaimatiroR") =
chkMiaimatiroR.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("DenanaferetaiL") =
chkDenanaferetaiL.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("DenanaferetaiR") =
chkDenanaferetaiR.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("DenanaferetaiR") =
chkAimatiroR.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("OgidioL") =
chkOgidioL.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("OgidioR") =
chkOgidioR.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("AimatiroL") =
chkAimatiroL.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("Symptomatikil") =
chkSymptomatikil.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("Asymptomatikil") =
chkAsymptomatikil.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("AlleskakoitheiesT") =
txtAlleskakoitheiesT.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AllaLD") =
txtAllaLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AllaRD") =
txtAllaRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("MasxaliaiesLD") =
txtMasxaliaiesLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("MasxaliaiesRD") =
txtMasxaliaiesRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("PonoskiLD") =
txtPonoskiLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("PonoskiRD") =
txtPonoskiRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("PonosmikiLD") =
txtPonosmikiLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("PonosmikiRD") =
txtPonosmikiRD.Text
        sTabs.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiLD")
= txtDermatikiparamorfosiLD.Text
        sTabs.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiRD")
= txtDermatikiparamorfosiRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AnomaliathilisLD") =
txtAnomaliathilisLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AnomaliathilisRD") =
txtAnomaliathilisRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("DenanaferetaiLD") =
txtDenanaferetaiLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("DenanaferetaiRD") =
txtDenanaferetaiRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("MiaimatiroLD") =
txtMiaimatiroLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("MiaimatiroRD") =
txtMiaimatiroRD.Text

```



```

        dsTabs.Tables(0).Rows.Find(rowID).Item("AimatiroLD") =
txtAimatiroLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AimatiroRD") =
txtAimatiroRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("OgidioLD") =
txtOgidioLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("OgidioRD") =
txtOgidioRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("SymptomatikiLD") =
txtSymptomatikiLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("SymptomatikiRD") =
txtSymptomatikiRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AsymptomatikiLD") =
txtAsymptomatikiLD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AsymptomatikiRD") =
txtAsymptomatikiRD.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("MetaggisiT") =
txtMetaggisiT.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AllergiesT") =
txtAllergiesT.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("LipsifarmakwnT") =
txtLipsifarmakwnT.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AllopathologikoT") =
txtAllopathologikoT.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("kakoithinososT") =
txtKakoithinosos.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("kaloithinososT") =
txtKaloithinosos.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("allergiesCYN") =
cmbAllergiesYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("MetaggisiCYN") =
cmbMetaggisiYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("LipsifarmakwnCYN") =
cmbLipsifarmakwnYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AllopathologikoC") =
cmbAllopathologiko.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AllopathologikoCYN") =
cmbAllopathologikoYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AlleskakoiitheiesC") =
cmbAlleskakoiitheies.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("AlleskakoiitheiesCYN") =
cmbAlleskakoiitheiesYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("kakoithinososCYN") =
cmbKakoithinososYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("kakoithinososCRL") =
cmbKakoithinososRL.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("kaloithinososCYN") =
cmbKaloithinososYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("kaloithinososCRL") =
cmbKaloithinososRL.Text
        daDiseasesData.Update(dsTabs, "DiseasesData")
    End If
    dsTabs.Tables.Remove("DiseasesData") 'me to remove adeiazei to
table
    'επεξεργασία στοιχειων παραγοντων κινδυνου
    sql = "SELECT * FROM DangersData WHERE id=" & rowID & ""
    Dim daDangersData = New OleDb.OleDbDataAdapter(sql, con)
    daDangersData.MissingSchemaAction = MissingSchemaAction.AddWithKey
    daDangersData.Fill(dsTabs, "DangersData")
    Dim cbDangersData As New OleDb.OleDbCommandBuilder(daDangersData)
    If dsTabs.Tables(0).Rows.Count = 0 Then

```

```

con.Close()
SubmitDangersData(rowID)
con.Open()
Else
dsTabs.Tables(0).Rows.Find(rowID).Item("EidosKarkinouVathmosSiggenias") =
txtEidosKarkinouVathmosSiggenias.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("HlikiaOlokliromenisKuisis") =
txtHlikiaOlokliromenisKuisis.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ArithmosKuisseon") =
txtArithmosKuisseon.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("Apovoles") =
txtApovoles.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("Hlikiaeminopausis") =
txtHlikiaeminopausis.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("Hlikiaeminarxis") =
txtHlikiaeminarxis.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononPillDur") =
txtProsfatiLipsiOrmononPillDur.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononPill") =
txtProsfatiLipsiOrmononPill.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononOtherDur") =
txtProsfatiLipsiOrmononOtherDur.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononOther") =
txtProsfatiLipsiOrmononOther.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononEidos") =
txtProsfatiLipsiOrmononEidos.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononEidos") =
txtProigoumeniLipsiOrmononEidos.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononPill") =
txtProigoumeniLipsiOrmononPill.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononOtherDur") =
txtProigoumeniLipsiOrmononOtherDur.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononOther") =
txtProigoumeniLipsiOrmononOther.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononPillDur") =
txtProigoumeniLipsiOrmononPillDur.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("Mitera") =
txtMitera.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("PeriptoseisTheies") =
txtPeriptoseisTheies.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("Alloi") = txtAlloi.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("Adelfi") =
txtAdelfi.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("PeriptoseisAdelfes") =
txtPeriptoseisAdelfes.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("TheiaApoPatera") =
txtTheiaApoPatera.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("TheiaDeuterouVathmou")
= txtTheiaDeuterouVathmou.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("GiagiaApoMitera") =
txtGiagiaApoMitera.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("GiagiaApoPatera") =
txtGiagiaApoPatera.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("Progiagia") =
txtProgiagia.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("TheiaApoMitera") =
txtTheiaApoMitera.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("OikogeniakoIstorikoKarkinouMastouYN")
= cmbOikogeniakoIstorikoKarkinouMastouYN.Text
dsTabs.Tables(0).Rows.Find(rowID).Item("OikogeniakoIstorikoAllonKarkinonYN")
= cmbOikogeniakoIstorikoAllonKarkinonYN.Text

```

```

        dsTabs.Tables(0).Rows.Find(rowID).Item("DiakopiThilasmouYN") =
cmbDiakopiThilasmouYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("ThilasateYN") =
cmbThilasateYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononYN") =
cmbProigoumeniLipsiOrmononYN.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononYN") =
cmbProsfatiLipsiOrmononYN.Text
        daDangersData.Update(dsTabs, "DangersData")
    End If
    dsTabs.Tables.Remove("DangersData")
    'επεξεργασία στοιχείων κλινικής εξέτασης
    If dgvClinical.Rows.Count <> 0 Then
        Dim clinicalID =
dgvClinical.SelectedRows(0).Cells(1).Value.ToString
        sql = "SELECT * FROM ClinicalData WHERE clinicalID=" &
clinicalID & ""
        Dim daClinicalData = New OleDb.OleDbDataAdapter(sql, con)
        daClinicalData.MissingSchemaAction =
MissingSchemaAction.AddWithKey
        daClinicalData.Fill(dsTabs, "ClinicalData")
        Dim cbClinicalData As New
OleDb.OleDbCommandBuilder(daClinicalData)
        dtpClinical.Value = New DateTime(dtpClinical.Value.Year,
dtpClinical.Value.Month, dtpClinical.Value.Day, DateTime.Now.Hour,
DateTime.Now.Minute, DateTime.Now.Second)
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("dateClinical") =
dtpClinical.Value
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("YperklidioiMegethos") =
txtYperklidioiMegethos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("YpoklidioiMegethos") =
txtYpoklidioiMegethos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiAkinitoiMegethos")
= txtMasxaliaioiAkinitoiMegethos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiKinitoiMegethos") =
txtMasxaliaioiKinitoiMegethos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("AnomaliesThilaiasEidos") =
txtAnomaliesThilaiasEidos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("TopikaOksidiaMegethos") =
txtTopikaOksidiaMegethos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("KlinikaEurimataYN") =
cmbKlinikaEurimataYN.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("SumpagesOgidioMegethos") =
txtSumpagesOgidioMegethos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("KistikoOgidioMegethos") =
txtKistikoOgidioMegethos.Text
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("SimfietaiPeritoniaL") =
chkSimfietaiPeritoniaL.Checked
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("SimfietaiPeritoniaR") =
chkSimfietaiPeritoniaR.Checked
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("SimfietaiThorakikoTixomaL") =
chkSimfietaiThorakikoTixomaL.Checked
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("SimfietaiThorakikoTixomaR") =
chkSimfietaiThorakikoTixomaR.Checked
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("EisolikiDermatosL") =
chkEisolikiDermatosL.Checked
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("PeauDOrangeL") =
chkPeauDOrangeL.Checked
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("PeauDOrangeR") =
chkPeauDOrangeR.Checked
        dsTabs.Tables(0).Rows.Find(clinicalID).Item("EisolikiDermatosR") =

```

```

chkEisolikiDermatosR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("DiithisiDermatosL") =
chkDiithisiDermatosL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("DiithisiDermatosR") =
chkDiithisiDermatosR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("EisolikiThilisR")
= chkEisolikiThilisR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiKinitoiL") =
chkMasxaliaioiKinitoiL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiKinitoiR") =
chkMasxaliaioiKinitoiR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiAkinitoiL") =
chkMasxaliaioiAkinitoiL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiAkinitoiR") =
chkMasxaliaioiAkinitoiR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("YperklidioiL") =
chkYperklidioiL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("YperklidioiR") =
chkYperklidioiR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("YpoklidioiL") =
chkYpoklidioiL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("YpoklidioiR") =
chkYpoklidioiR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("EsxaraThilisL") =
chkEsxaraThilisL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("EsxaraThilisR") =
chkEsxaraThilisR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("ErithrotitaThilisL") =
chkErithrotitaThilisL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("ErithrotitaThilisR") =
chkErithrotitaThilisR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("AimatiroEkrimaL")
= chkAimatiroEkrimaL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("AnomaliesThiliasL") =
chkAnomaliesThiliasL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("AnomaliesThiliasR") =
chkAnomaliesThiliasR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("AimatiroEkrimaR")
= chkAimatiroEkrimaR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MiAimatiroEkrimaL") =
chkMiAimatiroEkrimaL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MiAimatiroEkrimaR") =
chkMiAimatiroEkrimaR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("EisolikiThilisL")
= chkEisolikiThilisL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("SumpagesOgidioL")
= chkSumpagesOgidioL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("SumpagesOgidioR")
= chkSumpagesOgidioR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("DiaxitaOksidiaL")
= chkDiaxitaOksidiaL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("DiaxitaOksidiaR")
= chkDiaxitaOksidiaR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("TopikaOksidiaL") =
chkTopikaOksidiaL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("TopikiEuesthisiaL") =
chkTopikiEuesthisiaL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("TopikiEuesthisiaR") =
chkTopikiEuesthisiaR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("TopikaOksidiaR") =
chkTopikaOksidiaR.Checked

```

```

dsTabs.Tables(0).Rows.Find(clinicalID).Item("DiaxitiEuesthisiaL") =
chkDiaxitiEuesthisiaL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("DiaxitiEuesthisiaR") =
chkDiaxitiEuesthisiaR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("KistikoOgidioL") =
chkKistikoOgidioL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("KistikoOgidioR") =
chkKistikoOgidioR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("OuliL") =
chkOuliL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("OuliR") =
chkOuliR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("ProigoumenesEpemvaseisYN") =
cmbProigoumenesEpemvaseisYN.Text
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("ViopsiaL") =
chkViopsiaL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("ViopsiaR") =
chkViopsiaR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("OgkektomiL") =
chkOgkektomiL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("OgkektomiR") =
chkOgkektomiR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("TmimatikiMastektomiL") =
chkTmimatikiMastektomiL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("TmimatikiMastektomiR") =
chkTmimatikiMastektomiR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("OlikiMastektomiR")
= chkOlikiMastektomiR.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("OlikiMastektomiL") =
chkOlikiMastektomiL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaiosKatharismosL") =
chkMasxaliaiosKatharismosL.Checked
dsTabs.Tables(0).Rows.Find(clinicalID).Item("MasxaliaiosKatharismosR") =
chkMasxaliaiosKatharismosR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("EpanepemvasiL") =
chkEpanepemvasiL.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("EpanepemvasiR") =
chkEpanepemvasiR.Checked
    dsTabs.Tables(0).Rows.Find(clinicalID).Item("IkonidiaC") =
cmbIkonidia.SelectedIndex
    daClinicalData.Update(dsTabs, "ClinicalData")
    dsTabs.Tables.Remove("ClinicalData")
Else
    con.Close()
    SubmitClinicalData(rowID)
    con.Open()
End If
'επεξεργασία στοιχείων διαγνωστικού ελεγχού
If dgvDiagnosis.Rows.Count <> 0 Then
    Dim diagnosisID = dgvDiagnosis.SelectedRows(0).Cells(1).Value
    sql = "SELECT * FROM DiagnosisData WHERE diagnosisID=" &
diagnosisID & ""
    Dim daDiagnosisData = New OleDb.OleDbDataAdapter(sql, con)
    daDiagnosisData.MissingSchemaAction =
MissingSchemaAction.AddWithKey
    daDiagnosisData.Fill(dsTabs, "DiagnosisData")
    Dim cbDiagnosisData As New
OleDb.OleDbCommandBuilder(daDiagnosisData)
    dtpDiagnosis.Value = New DateTime(dtpDiagnosis.Value.Year,
dtpDiagnosis.Value.Month, dtpDiagnosis.Value.Day, DateTime.Now.Hour,
DateTime.Now.Minute, DateTime.Now.Second)

```

```

        dsTabs.Tables(0).Rows.Find(diagnosisID).Item("dateDiagnosis") =
dtpDiagnosis.Value
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaAneparkisL") =
chkViopsiaAneparkisL.Checked
        dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaThesiL") =
chkViopsiaThesiL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKakoithiR") =
chkViopsiaKakoithiR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKaloithiR") =
chkViopsiaKaloithiR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKakoithiL") =
chkViopsiaKakoithiL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaAneparkisR") =
chkViopsiaAneparkisR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKaloithiL") =
chkViopsiaKaloithiL.Checked
        dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaThesiR") =
chkViopsiaThesiR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosFisiologikiR") =
chkYperixosFisiologikiR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosFisiologikiL") =
chkYperixosFisiologikiL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosDisplasiaL") =
chkYperixosDisplasiaL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosDisplasiaR") =
chkYperixosDisplasiaR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosApotitanoseisL") =
chkYperixosApotitanoseisL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosApotitanoseisR") =
chkYperixosApotitanoseisR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosKakoithiaR") =
chkYperixosKakoithiaR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosKakoithiaL") =
chkYperixosKakoithiaL.Checked
        dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosYpoptoR")
= chkYperixosYpoptoR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosKaloithiaR") =
chkYperixosKaloithiaR.Checked
        dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosYpoptoL")
= chkYperixosYpoptoL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosKaloithiaL") =
chkYperixosKaloithiaL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEntosR") =
chkApotitanoseisEntosR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEntosL") =
chkApotitanoseisEntosL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEktosR") =
chkApotitanoseisEktosR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEktosL") =
chkApotitanoseisEktosL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaFisiologikiR") =
chkMastografiaFisiologikiR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaFisiologikiL") =
chkMastografiaFisiologikiL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDisplasiaL") =
chkMastografiaDisplasiaL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDisplasiaR") =
chkMastografiaDisplasiaR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaApotitanoseisL") =
chkMastografiaApotitanoseisL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaApotitanoseisR") =

```

```

chkMastografiaApotitanoseisR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKakoithiaR") =
chkMastografiaKakoithiaR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKakoithiaL") =
chkMastografiaKakoithiaL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaYpoptoR") =
chkMastografiaYpoptoR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKaloithiaR") =
chkMastografiaKaloithiaR.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaYpoptoL") =
chkMastografiaYpoptoL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKaloithiaL") =
chkMastografiaKaloithiaL.Checked
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNAAristeraKitarologikiL") =
txtFNAAristeraKitarologikiL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNAAristeraKitarologikiR") =
txtFNAAristeraKitarologikiR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNAAristeraThesiL") =
txtFNAAristeraThesiL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNAAristeraThesiR") =
txtFNAAristeraThesiR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaKitarologikiR") =
txtFNADexiaKitarologikiR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaKitarologikiL") =
txtFNADexiaKitarologikiL.Text
    dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaThesiL")
= txtFNADexiaThesiL.Text
    dsTabs.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaThesiR")
= txtFNADexiaThesiR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraMegethosR") =
txtYperixosAristeraMegethosR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraMegethosL") =
txtYperixosAristeraMegethosL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraThesiL") =
txtYperixosAristeraThesiL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraThesiR") =
txtYperixosAristeraThesiR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaMegethosR") =
txtYperixosDexiaMegethosR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaMegethosL") =
txtYperixosDexiaMegethosL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaThesiL") =
txtYperixosDexiaThesiL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaThesiR") =
txtYperixosDexiaThesiR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraMegethosR")
) = txtMastografiaAristeraMegethosR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraMegethosL")
) = txtMastografiaAristeraMegethosL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraThesiL") =
txtMastografiaAristeraThesiL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraThesiR") =
txtMastografiaAristeraThesiR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaMegethosR") =
txtMastografiaDexiaMegethosR.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaMegethosL") =
txtMastografiaDexiaMegethosL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaThesiL") =
txtMastografiaDexiaThesiL.Text
dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaThesiR") =
txtMastografiaDexiaThesiR.Text

```

```

dsTabs.Tables(0).Rows.Find(diagnosisID).Item("MastografiaApotitanoseisYN")
= cmbMastografiaApotitanoseisYN.Text
    daDiagnosisData.Update(dsTabs, "DiagnosisData")
    dsTabs.Tables.Remove("DiagnosisData")
Else
    con.Close()
    SubmitDiagnosisData(rowID)
    Dim diagID = dgvDiagnosis.SelectedRows(0).Cells(1).Value
    SubmitCoordinatesData(diagID)
    con.Open()
End If
'επεκσεργασια στοιχειων coordinatesData
If dgvDiagnosis.Rows.Count <> 0 Then
    Dim diagID = dgvDiagnosis.SelectedRows(0).Cells(1).Value
    sql = "SELECT * FROM CoordinatesData WHERE diagnosisID=" &
diagID & ""
    Dim daCoordinatesData = New OleDb.OleDbDataAdapter(sql, con)
    daCoordinatesData.MissingSchemaAction =
MissingSchemaAction.AddWithKey
    daCoordinatesData.Fill(dsTabs, "CoordinatesData")
    Dim cbCoordinatesData As New
OleDb.OleDbCommandBuilder(daCoordinatesData)
    If dsTabs.Tables(0).Rows.Count <> 0 Then
        For i As Integer = 0 To 9
            dsTabs.Tables(0).Rows(i).Item("mpx") = coordlist(1, i)
            dsTabs.Tables(0).Rows(i).Item("mpy") = coordlist(0, i)
            dsTabs.Tables(0).Rows(i).Item("paxos") = pinelo(i)
            daCoordinatesData.Update(dsTabs, "CoordinatesData")
        Next
    End If
    dsTabs.Tables.Remove("CoordinatesData")
End If
'επεξεργασια στοιχειων Σταδιοπολισης καρκινου
sql = "SELECT * FROM StadioData WHERE id=" & rowID & ""
Dim daStadioData = New OleDb.OleDbDataAdapter(sql, con)
daStadioData.MissingSchemaAction = MissingSchemaAction.AddWithKey
daStadioData.Fill(dsTabs, "StadioData")
Dim cbStadioData As New OleDb.OleDbCommandBuilder(daStadioData)
If dsTabs.Tables(0).Rows.Count = 0 Then
    con.Close()
    SubmitStadioData(rowID)
    con.Open()
Else
    dsTabs.Tables(0).Rows.Find(rowID).Item("TNMT") = txtTNMT.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("TNMN") = txtTNMN.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("TNMM") = txtTNMM.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("stadio") =
cmbStadio.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("porogenes") =
txtPorogenes.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("loviakos") =
txtLoviakos.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("thilis") = txtThilis.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("istoallo") =
txtIstoallo.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("apoleiavarous") =
txtApoleiavarous.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("dispnoia") =
txtDispnoia.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("vixas") = txtVixas.Text
    dsTabs.Tables(0).Rows.Find(rowID).Item("algos") = txtAlgos.Text

```



```

        dsTabs.Tables(0).Rows.Find(rowID).Item("metastallo") =
txtMetastallo.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("anapneustikoF") =
chkAnapneustikoF.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("anapneustikoO") =
chkAnapneustikoO.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("anapneustikoM") =
chkAnapneustikoM.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("ostaF") =
chkOstaF.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("ostaO") =
chkOstaO.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("ostaM") =
chkOstaM.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("koiliaF") =
chkKoiliaF.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("koiliaO") =
chkKoiliaO.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("koiliaM") =
chkKoiliaM.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("KNSF") =
chkKNSF.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("KNSO") =
chkKNSO.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("KNSM") =
chkKNSM.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("allaF") =
chkAllaF.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("allaO") =
chkAllaO.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("allaM") =
chkAllaM.Checked
        dsTabs.Tables(0).Rows.Find(rowID).Item("xeirourgikiC") =
cmbXeirourgiki.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("sumpliromatikitherapeiaC") =
cmbSumpliromatikitherapeia.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("katastasiC") =
cmbKatastasi.Text
        dsTabs.Tables(0).Rows.Find(rowID).Item("simeioypotropis") =
txtSimeioypotropis.Text
        daStadioData.Update(dsTabs, "StadioData")
    End If
    dsTabs.Tables.Remove("StadioData")
    'επεξεργασία στοιχείων Παρακολούθησης ασθενών
    If dgvObserve.Rows.Count <> 0 Then
        Dim observeID = dgvObserve.SelectedRows(0).Cells(1).Value
        sql = "SELECT * FROM ObserveData WHERE observeID=" & observeID
& ""
        Dim daObserveData = New OleDb.OleDbDataAdapter(sql, con)
        daObserveData.MissingSchemaAction =
MissingSchemaAction.AddWithKey
        daObserveData.Fill(dsTabs, "ObserveData")
        Dim cbDiagnosisData As New
OleDb.OleDbCommandBuilder(daObserveData)
        dtpObserve.Value = New DateTime(dtpObserve.Value.Year,
dtpObserve.Value.Month, dtpObserve.Value.Day, DateTime.Now.Hour,
DateTime.Now.Minute, DateTime.Now.Second)
        dsTabs.Tables(0).Rows.Find(observeID).Item("dateO") =
dtpObserve.Value
        dsTabs.Tables(0).Rows.Find(observeID).Item("minesmetaC") =
cmbMinesmeta.Text

```

```

        dsTabs.Tables(0).Rows.Find(observeID).Item("trauma") =
txtTrauma.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("masxali") =
txtMasxali.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("sustoixo") =
txtSustoixo.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("plastiki") =
txtPlastiki.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("aisthanetai") =
txtAisthanetai.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("Hct") = txtHct.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("Hb") = txtHb.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("WBC") = txtWBC.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("NEUTR") =
txtNEUTR.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("LYMPH") =
txtLYMPH.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("PLT") = txtPLT.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("SGOT") =
txtSGOT.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("GGT") = txtGGT.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("ALP") = txtALP.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("LDH") = txtLDH.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("CA125") =
txtCA125.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("CA153") =
txtCA153.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("CA199") =
txtCA199.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("CEA") = txtCEA.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("mastografia") =
txtMastografia.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("ypermastwn") =
txtYpermastwn.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("yperanokatokoilias")
= txtYperanokatokoilias.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("aktinothorakos") =
txtAktinothorakos.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("aksonikithorakos")
= txtAksonikithorakos.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("aksonikikoilias") =
txtAksonikikoilias.Text
        dsTabs.Tables(0).Rows.Find(observeID).Item("spinthirografima")
= txtSpinthirografima.Text
        daObserveData.Update(dsTabs, "ObserveData")
        dsTabs.Tables.Remove("ObserveData")
    Else
        con.Close()
        SubmitObserveData(rowID)
        con.Open()
    End If
    MsgBox("Η αποθήκευση αλλαγών για την καταχώρηση με ID " & rowID &
" ολοκληρώθηκε με επιτυχία")
    con.Close()
    FillDGVs(dgvClinical, "Query1", rowID)
    FillDGVs(dgvDiagnosis, "Query2", rowID)
    '---> Γέμισμα των
DGVs
    FillDGVs(dgvObserve, "Query3", rowID)
End Sub

```

```

Private Sub btnNewRecord_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnNewRecord.Click
    Dim sql As String
    'Dim dsDT As New DataSet
    Dim ds1 As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    If btnNewRecord.Text = "Καταχώρηση" Then
        If TxtName_NeosAsthenis.Text <> "" Then
            con.Open()
            sql = "SELECT id , name as [Όνοματεπώνυμο], fname as
[Πατρώνυμο] ,phone as [Τηλέφωνο], address as [Διεύθυνση] , birthDate as
[Ημερομηνία γέννησης],AMKA as [AMKA] FROM BasicData ORDER BY id"
            da = New OleDb.OleDbDataAdapter(sql, con)
            Dim cb As New OleDb.OleDbCommandBuilder(da)
            Dim dsNewRow As DataRow
            dsNewRow = dsDT.Tables("BasicData").NewRow()
            dsNewRow.Item("Όνοματεπώνυμο") = TxtName_NeosAsthenis.Text
            dsNewRow.Item("Τηλέφωνο") = txtPhone_NeosAsthenis.Text
            dsNewRow.Item("Διεύθυνση") = txtAddress_NeosAsthenis.Text
            dsNewRow.Item("Πατρώνυμο") = txtFname_NeosAsthenis.Text
            If IsValidAMKA(txtAMKA_NeosAsthenis.Text) = True Then
                dsNewRow.Item("AMKA") = txtAMKA_NeosAsthenis.Text
                txtAge_NeosAsthenis.Text =
getDatefromAMKA(txtAMKA_NeosAsthenis.Text)
                If IsNumeric(searchString(txtAMKA_NeosAsthenis.Text,
"AMKA")) Then
                    MsgBox("Ο αριθμός Α.Μ.Κ.Α που πληκτρολογήσατε έχει
ήδη καταχωρηθεί σε ασθενή")
                End If
            Else
                MsgBox("Ο αριθμός Α.Μ.Κ.Α που πληκτρολογήσατε δεν είναι
έγκυρος. Ο αριθμός Α.Μ.Κ.Α δεν καταχωρήθηκε")
                End If
            If IsDate(txtAge_NeosAsthenis.Text) = True Then
                dsNewRow.Item("Ημερομηνία γέννησης") =
txtAge_NeosAsthenis.Text
            Else
                MsgBox("Το πεδίο Ημερομηνία γέννησης πρέπει να είναι σε
μορφή */*/*. Το πεδίο Ημερομηνία γέννησης δεν άλλαξε")
                End If
            dsDT.Tables("BasicData").Rows.Add(dsNewRow)
            da.Update(dsDT, "BasicData")
            If DT.RowCount = 1 Then
                BtnDeleteR.Enabled = True
                btnViewAndProcess.Enabled = True
                da.MissingSchemaAction = MissingSchemaAction.AddWithKey
                dsDT.Tables.Remove("BasicData")
                da.Fill(dsDT, "BasicData")
                DT.DataSource = dsDT.Tables(0)
                rowID = DT.Rows(0).Cells(0).Value.ToString
            End If
            con.Close()
            MsgBox("Ο ασθενής καταχωρήθηκε με επιτυχία")
            TabControlKartel.Enabled = True
            TabControlKartel.Visible = True
        Else
            MsgBox("Καταχωρήστε τουλάχιστον ένα όνομα για να
συνεχίσετε")
        End If
    End If
    End If
    gbxStoixeiaNeouAsthenous.Visible = False

```

```

gbxStoixeiaAsthenous.Visible = True
lblActualName.Text = TxtName_NeosAsthenis.Text
lblActualAge.Text = txtAge_NeosAsthenis.Text
lblActualAddress.Text = txtAddress_NeosAsthenis.Text
lblActualPhone.Text = txtPhone_NeosAsthenis.Text
If btnNewRecord.Text = "Αποθήκευση" Then
    Dim numofrows As Integer
    numofrows = dsDT.Tables("BasicData").Rows.Count - 1
    Dim lastid As String
    lastid = dsDT.Tables("BasicData").Rows(numofrows).Item("id")
    SubmitDeseasesData(lastid)
    SubmitDangersData(lastid)
    SubmitDiagnosisData(lastid)
    SubmitClinicalData(lastid)
    SubmitStadioData(lastid)
    SubmitObserveData(lastid)
    'submit coordinates
    Dim ds As New DataSet
    Dim sql1 = "SELECT * FROM DiagnosisData WHERE id=" & lastid &
""
    Dim da1 = New OleDb.OleDbDataAdapter(sql1, con)
    da1.Fill(ds, "DiagnosisData")
    Dim numofrows1 As Integer
    numofrows1 = ds.Tables("DiagnosisData").Rows.Count - 1
    Dim lastdiagID As String
    lastdiagID =
ds.Tables("DiagnosisData").Rows(numofrows1).Item("diagnosisID")
    SubmitCoordinatesData(lastdiagID)
    MsgBox("Η καταχώρηση των στοιχείων ολοκληρώθηκε με επιτυχία")
    btnReset_Click(sender, New System.EventArgs())
End If
If btnNewRecord.Text = "Αποθήκευση" Then
    btnReturn_Click(sender, New System.EventArgs())
Else
    btnNewRecord.Text = "Αποθήκευση"
End If
btnReturn.Enabled = True
End Sub

Public Sub SubmitClinicalData(ByVal PatientID As String)
    Dim dsClinicalData As New DataSet
    Dim sql As String = "SELECT * FROM ClinicalData WHERE id=" &
PatientID & "" 'Προσθέτοντας ως PatientID ως όρισμα, κάνουμε το ds μικρότερο
σε μέγεθος.
    con.Open()
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(dsClinicalData, "ClinicalData")
    Dim cb As New OleDb.OleDbCommandBuilder(da)
    Dim regDate As Date = Date.Today
    Dim dsNewRow As DataRow
    dsNewRow = dsClinicalData.Tables("ClinicalData").NewRow()
    dtpClinical.Value = New DateTime(dtpClinical.Value.Year,
dtpClinical.Value.Month, dtpClinical.Value.Day, DateTime.Now.Hour,
DateTime.Now.Minute, DateTime.Now.Second)
    dsNewRow.Item("dateClinical") = dtpClinical.Value
    dsNewRow.Item("id") = PatientID
    dsNewRow.Item("KlinikaEurimataYN") = cmbKlinikaEurimataYN.Text
    dsNewRow.Item("ProigoumenesEpemvaseisYN") =
cmbProigoumenesEpemvaseisYN.Text
    dsNewRow.Item("KistikoOgidioR") = chkKistikoOgidioR.CheckState

```

```

        dsNewRow.Item("SimfietaiPeritoniaL") =
chkSimfietaiPeritoniaL.CheckState
        dsNewRow.Item("SimfietaiPeritoniaR") =
chkSimfietaiPeritoniaR.CheckState
        dsNewRow.Item("SimfietaiThorakikoTixomaL") =
chkSimfietaiThorakikoTixomaL.CheckState
        dsNewRow.Item("SimfietaiThorakikoTixomaR") =
chkSimfietaiThorakikoTixomaR.CheckState
        dsNewRow.Item("OuliL") = chkOuliL.CheckState
        dsNewRow.Item("OuliR") = chkOuliR.CheckState
        dsNewRow.Item("EisolikiDermatosL") =
chkEisolikiDermatosL.CheckState
        dsNewRow.Item("PeauDOrangeL") = chkPeauDOrangeL.CheckState
        dsNewRow.Item("PeauDOrangeR") = chkPeauDOrangeR.CheckState
        dsNewRow.Item("EisolikiDermatosR") =
chkEisolikiDermatosR.CheckState
        dsNewRow.Item("DiithisiDermatosL") =
chkDiithisiDermatosL.CheckState
        dsNewRow.Item("DiithisiDermatosR") =
chkDiithisiDermatosR.CheckState
        dsNewRow.Item("YperklidioiMegethos") = txtYperklidioiMegethos.Text
        dsNewRow.Item("YpoklidioiMegethos") = txtYpoklidioiMegethos.Text
        dsNewRow.Item("MasxaliaioiKinitoiL") =
chkMasxaliaioiKinitoiL.CheckState
        dsNewRow.Item("MasxaliaioiKinitoiR") =
chkMasxaliaioiKinitoiR.CheckState
        dsNewRow.Item("MasxaliaioiAkinitoiL") =
chkMasxaliaioiAkinitoiL.CheckState
        dsNewRow.Item("MasxaliaioiAkinitoiR") =
chkMasxaliaioiAkinitoiR.CheckState
        dsNewRow.Item("YperklidioiL") = chkYperklidioiL.CheckState
        dsNewRow.Item("YperklidioiR") = chkYperklidioiR.CheckState
        dsNewRow.Item("YpoklidioiL") = chkYpoklidioiL.CheckState
        dsNewRow.Item("YpoklidioiR") = chkYpoklidioiR.CheckState
        dsNewRow.Item("MasxaliaioiAkinitoiMegethos") =
txtMasxaliaioiAkinitoiMegethos.Text
        dsNewRow.Item("MasxaliaioiKinitoiMegethos") =
txtMasxaliaioiKinitoiMegethos.Text
        dsNewRow.Item("AnomaliesThilaiasEidos") =
txtAnomaliesThilaiasEidos.Text
        dsNewRow.Item("EsxaraThilisL") = chkEsxaraThilisL.CheckState
        dsNewRow.Item("EsxaraThilisR") = chkEsxaraThilisR.CheckState
        dsNewRow.Item("ErithrotitaThilisL") =
chkErithrotitaThilisL.CheckState
        dsNewRow.Item("ErithrotitaThilisR") =
chkErithrotitaThilisR.CheckState
        dsNewRow.Item("AimatiroEkrimaL") = chkAimatiroEkrimaL.CheckState
        dsNewRow.Item("AnomaliesThilaiasL") =
chkAnomaliesThilaiasL.CheckState
        dsNewRow.Item("AnomaliesThilaiasR") =
chkAnomaliesThilaiasR.CheckState
        dsNewRow.Item("AimatiroEkrimaR") = chkAimatiroEkrimaR.CheckState
        dsNewRow.Item("MiAimatiroEkrimaL") =
chkMiAimatiroEkrimaL.CheckState
        dsNewRow.Item("MiAimatiroEkrimaR") =
chkMiAimatiroEkrimaR.CheckState
        dsNewRow.Item("EisolikiThilisL") = chkEisolikiThilisL.CheckState
        dsNewRow.Item("EisolikiThilisR") = chkEisolikiThilisR.CheckState
        dsNewRow.Item("TopikaOksidiaMegethos") =
txtTopikaOksidiaMegethos.Text
        dsNewRow.Item("SumpagesOgidioL") = chkSumpagesOgidioL.CheckState

```

```

        dsNewRow.Item("SumpagesOgidioR") = chkSumpagesOgidioR.CheckState
        dsNewRow.Item("DiaxitaOksidiaL") = chkDiaxitaOksidiaL.CheckState
        dsNewRow.Item("DiaxitaOksidiaR") = chkDiaxitaOksidiaR.CheckState
        dsNewRow.Item("TopikaOksidiaL") = chkTopikaOksidiaL.CheckState
        dsNewRow.Item("TopikiEuesthisiaL") =
chkTopikiEuesthisiaL.CheckState
        dsNewRow.Item("TopikiEuesthisiaR") =
chkTopikiEuesthisiaR.CheckState
        dsNewRow.Item("TopikaOksidiaR") = chkTopikaOksidiaR.CheckState
        dsNewRow.Item("DiaxitiEuesthisiaL") =
chkDiaxitiEuesthisiaL.CheckState
        dsNewRow.Item("DiaxitiEuesthisiaR") =
chkDiaxitiEuesthisiaR.CheckState
        dsNewRow.Item("KistikoOgidioL") = chkKistikoOgidioL.CheckState
        dsNewRow.Item("KistikoOgidioR") = chkKistikoOgidioR.CheckState
        dsNewRow.Item("SumpagesOgidioMegethos") =
txtSumpagesOgidioMegethos.Text
        dsNewRow.Item("KistikoOgidioMegethos") =
txtKistikoOgidioMegethos.Text
        dsNewRow.Item("ViopsiaL") = chkViopsiaL.Checked
        dsNewRow.Item("ViopsiaR") = chkViopsiaR.Checked
        dsNewRow.Item("OgkektomiL") = chkOgkektomiL.Checked
        dsNewRow.Item("OgkektomiR") = chkOgkektomiR.Checked
        dsNewRow.Item("TmimatikiMastektomiL") =
chkTmimatikiMastektomiL.Checked
        dsNewRow.Item("TmimatikiMastektomiR") =
chkTmimatikiMastektomiR.Checked
        dsNewRow.Item("OlikiMastektomiR") = chkOlikiMastektomiR.Checked
        dsNewRow.Item("OlikiMastektomiL") = chkOlikiMastektomiL.Checked
        dsNewRow.Item("MasxaliaiosKatharismosL") =
chkMasxaliaiosKatharismosL.Checked
        dsNewRow.Item("MasxaliaiosKatharismosR") =
chkMasxaliaiosKatharismosR.Checked
        dsNewRow.Item("EpanepemvasiL") = chkEpanepemvasiL.Checked
        dsNewRow.Item("EpanepemvasiR") = chkEpanepemvasiR.Checked
        dsNewRow.Item("IkonidiaC") = cmbIkonidia.SelectedIndex
        dsClinicalData.Tables("ClinicalData").Rows.Add(dsNewRow)
        da.Update(dsClinicalData, "ClinicalData")
        con.Close()
        FillDGVs(dgvClinical, "Query1", PatientID) 'Για να εμφανιστεί η νέα
καταχώρηση.
        dgvClinical.ClearSelection() 'Ξεπιλέγει τις επιλεγμένες σειρές.
        dgvClinical.Rows(dgvClinical.RowCount - 1).Selected = True
'Επιλέγει την τελευταία σειρά.
    End Sub

```

```

Public Sub SubmitDangersData(ByVal PatientID As String)
    Dim dsNewRow As DataRow
    Dim dsDangersData As New DataSet
    Dim sql = "SELECT * FROM DangersData WHERE id=" & PatientID & ""
    con.Open()
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(dsDangersData, "DangersData")
    Dim cb As New OleDb.OleDbCommandBuilder(da)
    dsNewRow = dsDangersData.Tables("DangersData").NewRow()
    dsNewRow.Item("id") = PatientID
    dsNewRow.Item("EidosKarkinouVathmosSiggenias") =
txtEidosKarkinouVathmosSiggenias.Text
    dsNewRow.Item("OikogeniakoIstorikoAllonKarkinonYN") =
cmbOikogeniakoIstorikoAllonKarkinonYN.Text
    dsNewRow.Item("DiakopiThilasmouYN") = cmbDiakopiThilasmouYN.Text

```

```

        dsNewRow.Item("ThilasateYN") = cmbThilasateYN.Text
        dsNewRow.Item("HlikiaOlokliromenisKuisis") =
txtHlikiaOlokliromenisKuisis.Text
        dsNewRow.Item("ArithmosKuisseon") = txtArithmosKuisseon.Text
        dsNewRow.Item("Apovoles") = txtApovoles.Text
        dsNewRow.Item("ProigoumeniLipsiOrmononYN") =
cmbProigoumeniLipsiOrmononYN.Text
        dsNewRow.Item("ProsfatiLipsiOrmononYN") =
cmbProsfatiLipsiOrmononYN.Text
        dsNewRow.Item("OikogeniakoIstorikoKarkinouMastouYN") =
cmbOikogeniakoIstorikoKarkinouMastouYN.Text
        dsNewRow.Item("Hlikiaeminopausis") = txtHlikiaeminopausis.Text
        dsNewRow.Item("Hlikiaeminarxis") = txtHlikiaeminarxis.Text
        dsNewRow.Item("ProsfatiLipsiOrmononPill") =
txtProsfatiLipsiOrmononPill.Text
        dsNewRow.Item("ProsfatiLipsiOrmononOtherDur") =
txtProsfatiLipsiOrmononOtherDur.Text
        dsNewRow.Item("ProsfatiLipsiOrmononOther") =
txtProsfatiLipsiOrmononOther.Text
        dsNewRow.Item("ProsfatiLipsiOrmononEidos") =
txtProsfatiLipsiOrmononEidos.Text
        dsNewRow.Item("ProigoumeniLipsiOrmononPillDur") =
txtProigoumeniLipsiOrmononPillDur.Text
        dsNewRow.Item("ProigoumeniLipsiOrmononPill") =
txtProigoumeniLipsiOrmononPill.Text
        dsNewRow.Item("ProigoumeniLipsiOrmononOtherDur") =
txtProigoumeniLipsiOrmononOtherDur.Text
        dsNewRow.Item("ProigoumeniLipsiOrmononOther") =
txtProigoumeniLipsiOrmononOther.Text
        dsNewRow.Item("ProigoumeniLipsiOrmononEidos") =
txtProigoumeniLipsiOrmononEidos.Text
        dsNewRow.Item("PeriptoseisTheies") = txtPeriptoseisTheies.Text
        dsNewRow.Item("Alloi") = txtAlloi.Text
        dsNewRow.Item("Adelfi") = txtAdelfi.Text
        dsNewRow.Item("PeriptoseisAdelfes") = txtPeriptoseisAdelfes.Text
        dsNewRow.Item("TheiaApoPatera") = txtTheiaApoPatera.Text
        dsNewRow.Item("TheiaDeuterouVathmou") =
txtTheiaDeuterouVathmou.Text
        dsNewRow.Item("GiagiaApoMitera") = txtGiagiaApoMitera.Text
        dsNewRow.Item("GiagiaApoPatera") = txtGiagiaApoPatera.Text
        dsNewRow.Item("Progiagia") = txtProgiagia.Text
        dsNewRow.Item("TheiaApoMitera") = txtTheiaApoMitera.Text
        dsNewRow.Item("Mitera") = txtMitera.Text
        dsDangersData.Tables("DangersData").Rows.Add(dsNewRow)
        da.Update(dsDangersData, "DangersData")
        con.Close()
    End Sub

```

```

Public Sub SubmitDiagnosisData(ByVal PatientID As String)
    Dim dsNewRow As DataRow
    Dim dsDiagnosisData As New DataSet
    Dim sql = "SELECT * FROM DiagnosisData WHERE id=" & PatientID & ""
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    Dim regDate As Date = Date.Today
    con.Open()
    da.Fill(dsDiagnosisData, "DiagnosisData")
    Dim cb As New OleDb.OleDbCommandBuilder(da)
    dsNewRow = dsDiagnosisData.Tables("DiagnosisData").NewRow()
    dsNewRow.Item("id") = PatientID

```

```

        dtpDiagnosis.Value = New DateTime(dtpDiagnosis.Value.Year,
dtpDiagnosis.Value.Month, dtpDiagnosis.Value.Day, DateTime.Now.Hour,
DateTime.Now.Minute, DateTime.Now.Second)
        dsNewRow.Item("dateDiagnosis") = dtpDiagnosis.Value
        dsNewRow.Item("ViopsiaAneparkisL") =
chkViopsiaAneparkisL.CheckState
        dsNewRow.Item("ViopsiaThesiL") = chkViopsiaThesiL.CheckState
        dsNewRow.Item("ViopsiaKakoithiR") = chkViopsiaKakoithiR.CheckState
        dsNewRow.Item("ViopsiaKaloithiR") = chkViopsiaKaloithiR.CheckState
        dsNewRow.Item("ViopsiaKakoithiL") = chkViopsiaKakoithiL.CheckState
        dsNewRow.Item("ViopsiaAneparkisR") =
chkViopsiaAneparkisR.CheckState
        dsNewRow.Item("ViopsiaKaloithiL") = chkViopsiaKaloithiL.CheckState
        dsNewRow.Item("ViopsiaThesiR") = chkViopsiaThesiR.CheckState
        dsNewRow.Item("YperixosFisiologikiR") =
chkYperixosFisiologikiR.CheckState
        dsNewRow.Item("YperixosFisiologikiL") =
chkYperixosFisiologikiL.CheckState
        dsNewRow.Item("YperixosDisplasiaL") =
chkYperixosDisplasiaL.CheckState
        dsNewRow.Item("YperixosDisplasiaR") =
chkYperixosDisplasiaR.CheckState
        dsNewRow.Item("YperixosApotitanoseisL") =
chkYperixosApotitanoseisL.CheckState
        dsNewRow.Item("YperixosApotitanoseisR") =
chkYperixosApotitanoseisR.CheckState
        dsNewRow.Item("YperixosKakoithiaR") =
chkYperixosKakoithiaR.CheckState
        dsNewRow.Item("YperixosKakoithiaL") =
chkYperixosKakoithiaL.CheckState
        dsNewRow.Item("YperixosYpoptoR") = chkYperixosYpoptoR.CheckState
        dsNewRow.Item("YperixosKaloithiaR") =
chkYperixosKaloithiaR.CheckState
        dsNewRow.Item("YperixosYpoptoL") = chkYperixosYpoptoL.CheckState
        dsNewRow.Item("YperixosKaloithiaL") =
chkYperixosKaloithiaL.CheckState
        dsNewRow.Item("ApotitanoseisEntosR") =
chkApotitanoseisEntosR.CheckState
        dsNewRow.Item("ApotitanoseisEntosL") =
chkApotitanoseisEntosL.CheckState
        dsNewRow.Item("ApotitanoseisEktosR") =
chkApotitanoseisEktosR.CheckState
        dsNewRow.Item("ApotitanoseisEktosR") =
chkApotitanoseisEktosR.CheckState
        dsNewRow.Item("ApotitanoseisEktosL") =
chkApotitanoseisEktosL.CheckState
        dsNewRow.Item("MastografiaFisiologikiR") =
chkMastografiaFisiologikiR.CheckState
        dsNewRow.Item("MastografiaFisiologikiL") =
chkMastografiaFisiologikiL.CheckState
        dsNewRow.Item("MastografiaDisplasiaL") =
chkMastografiaDisplasiaL.CheckState
        dsNewRow.Item("MastografiaDisplasiaR") =
chkMastografiaDisplasiaR.CheckState
        dsNewRow.Item("FNAAristeraKitarologikiL") =
txtFNAAristeraKitarologikiL.Text
        dsNewRow.Item("FNAAristeraKitarologikiR") =
txtFNAAristeraKitarologikiR.Text
        dsNewRow.Item("FNAAristeraThesiL") = txtFNAAristeraThesiL.Text
        dsNewRow.Item("FNAAristeraThesiR") = txtFNAAristeraThesiR.Text

```



```

        dsNewRow.Item("FNADexiaKitarologikiR") =
txtFNADexiaKitarologikiR.Text
        dsNewRow.Item("FNADexiaKitarologikiL") =
txtFNADexiaKitarologikiL.Text
        dsNewRow.Item("FNADexiaThesiL") = txtFNADexiaThesiL.Text
        dsNewRow.Item("FNADexiaThesiR") = txtFNADexiaThesiR.Text
        dsNewRow.Item("YperixosAristeraMegethosR") =
txtYperixosAristeraMegethosR.Text
        dsNewRow.Item("YperixosAristeraMegethosL") =
txtYperixosAristeraMegethosL.Text
        dsNewRow.Item("YperixosAristeraThesiL") =
txtYperixosAristeraThesiL.Text
        dsNewRow.Item("YperixosAristeraThesiR") =
txtYperixosAristeraThesiR.Text
        dsNewRow.Item("YperixosDexiaMegethosR") =
txtYperixosDexiaMegethosR.Text
        dsNewRow.Item("YperixosDexiaMegethosL") =
txtYperixosDexiaMegethosL.Text
        dsNewRow.Item("YperixosDexiaThesiL") = txtYperixosDexiaThesiL.Text
        dsNewRow.Item("YperixosDexiaThesiR") = txtYperixosDexiaThesiR.Text
        dsNewRow.Item("MastografiaAristeraMegethosR") =
txtMastografiaAristeraMegethosR.Text
        dsNewRow.Item("MastografiaAristeraMegethosL") =
txtMastografiaAristeraMegethosL.Text
        dsNewRow.Item("MastografiaAristeraThesiL") =
txtMastografiaAristeraThesiL.Text
        dsNewRow.Item("MastografiaAristeraThesiR") =
txtMastografiaAristeraThesiR.Text
        dsNewRow.Item("MastografiaDexiaMegethosR") =
txtMastografiaDexiaMegethosR.Text
        dsNewRow.Item("MastografiaDexiaMegethosL") =
txtMastografiaDexiaMegethosL.Text
        dsNewRow.Item("MastografiaDexiaThesiL") =
txtMastografiaDexiaThesiL.Text
        dsNewRow.Item("MastografiaDexiaThesiR") =
txtMastografiaDexiaThesiR.Text
        dsNewRow.Item("MastografiaApotitanoseisYN") =
cmbMastografiaApotitanoseisYN.Text
        dsNewRow.Item("MastografiaApotitanoseisL") =
chkMastografiaApotitanoseisL.CheckState
        dsNewRow.Item("MastografiaApotitanoseisR") =
chkMastografiaApotitanoseisR.CheckState
        dsNewRow.Item("MastografiaKakoithiaR") =
chkMastografiaKakoithiaR.CheckState
        dsNewRow.Item("MastografiaKakoithiaL") =
chkMastografiaKakoithiaL.CheckState
        dsNewRow.Item("MastografiaYpoptoR") =
chkMastografiaYpoptoR.CheckState
        dsNewRow.Item("MastografiaKaloithiaR") =
chkMastografiaKaloithiaR.CheckState
        dsNewRow.Item("MastografiaYpoptoL") =
chkMastografiaYpoptoL.CheckState
        dsNewRow.Item("MastografiaKaloithiaL") =
chkMastografiaKaloithiaL.CheckState
        dsNewRow.Item("MastografiaKaloithiaL") =
chkMastografiaKaloithiaL.CheckState
        dsDiagnosisData.Tables("DiagnosisData").Rows.Add(dsNewRow)
        da.Update(dsDiagnosisData, "DiagnosisData")
        con.Close()
        FillDGVs(dgvDiagnosis, "Query2", PatientID) 'Για να εμφανιστεί η
νέα καταχώρηση.

```

```

        dgvDiagnosis.ClearSelection() 'Εξεπιλέγει τις επιλεγμένες σειρές.
        dgvDiagnosis.Rows(dgvDiagnosis.RowCount - 1).Selected = True
'Επιλέγει την τελευταία σειρά.
    End Sub

```

```

Public Sub SubmitDeseasesData(ByVal PatientID As String)
    Dim dsNewRow As DataRow
    Dim dsDeseasesData As New DataSet
    Dim sql = "SELECT * FROM DeseasesData WHERE id=" & PatientID & ""
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    con.Open()
    da.Fill(dsDeseasesData, "DeseasesData")
    Dim cb As New OleDb.OleDbCommandBuilder(da)
    dsNewRow = dsDeseasesData.Tables("DeseasesData").NewRow()
    dsNewRow.Item("id") = PatientID
    dsNewRow.Item("asymptomatikiR") = chkAsymptomatikiR.Checked
    dsNewRow.Item("symptomatikiR") = chkSymptomatikiR.Checked
    dsNewRow.Item("asymptomatikiL") = chkAsymptomatikiL.CheckState
    dsNewRow.Item("symptomatikiL") = chkSymptomatikiL.CheckState
    dsNewRow.Item("asymptomatikiRD") = txtAsymptomatikiRD.Text
    dsNewRow.Item("symptomatikiRD") = txtSymptomatikiRD.Text
    dsNewRow.Item("asymptomatikiLD") = txtAsymptomatikiLD.Text
    dsNewRow.Item("symptomatikiLD") = txtSymptomatikiLD.Text
    dsNewRow.Item("ogidioR") = chkOgidioR.Checked
    dsNewRow.Item("ogidioL") = chkOgidioL.Checked
    dsNewRow.Item("ogidioRD") = txtOgidioRD.Text
    dsNewRow.Item("ogidioLD") = txtOgidioLD.Text
    dsNewRow.Item("aimatiroR") = chkAimatiroR.Checked
    dsNewRow.Item("aimatiroL") = chkAimatiroL.Checked
    dsNewRow.Item("aimatiroRD") = txtAimatiroRD.Text
    dsNewRow.Item("aimatiroLD") = txtAimatiroLD.Text
    dsNewRow.Item("miaaimatiroR") = chkMiaaimatiroR.Checked
    dsNewRow.Item("miaaimatiroL") = chkMiaaimatiroL.Checked
    dsNewRow.Item("miaaimatiroRD") = txtMiaaimatiroRD.Text
    dsNewRow.Item("miaaimatiroLD") = txtMiaaimatiroLD.Text
    dsNewRow.Item("denanaferetaiR") = chkDenanaferetaiR.Checked
    dsNewRow.Item("denanaferetaiL") = chkDenanaferetaiL.Checked
    dsNewRow.Item("denanaferetaiRD") = txtDenanaferetaiRD.Text
    dsNewRow.Item("denanaferetaiLD") = txtDenanaferetaiLD.Text
    dsNewRow.Item("anomaliathilisR") = chkAnomaliathilisR.Checked
    dsNewRow.Item("anomaliathilisL") = chkAnomaliathilisL.Checked
    dsNewRow.Item("anomaliathilisRD") = txtAnomaliathilisRD.Text
    dsNewRow.Item("anomaliathilisLD") = txtAnomaliathilisLD.Text
    dsNewRow.Item("dermatikiparamorfosiR") =
chkDermatikiparamorfosiR.Checked
    dsNewRow.Item("dermatikiparamorfosiL") =
chkDermatikiparamorfosiL.Checked
    dsNewRow.Item("dermatikiparamorfosiRD") =
txtDermatikiparamorfosiRD.Text
    dsNewRow.Item("dermatikiparamorfosiLD") =
txtDermatikiparamorfosiLD.Text
    dsNewRow.Item("ponosmikiR") = chkPonosmikiR.Checked
    dsNewRow.Item("ponosmikiL") = chkPonosmikiL.Checked
    dsNewRow.Item("ponosmikiRD") = txtPonosmikiRD.Text
    dsNewRow.Item("ponosmikiLD") = txtPonosmikiLD.Text
    dsNewRow.Item("ponoskiR") = chkPonoskiR.Checked
    dsNewRow.Item("ponoskiL") = chkPonoskiL.Checked
    dsNewRow.Item("ponoskiRD") = txtPonoskiRD.Text
    dsNewRow.Item("ponoskiLD") = txtPonoskiLD.Text
    dsNewRow.Item("masxaliaiesR") = chkMasxaliaiesR.Checked
    dsNewRow.Item("masxaliaiesL") = chkMasxaliaiesL.Checked

```

```

dsNewRow.Item("masxaliaiesRD") = txtMasxaliaiesRD.Text
dsNewRow.Item("masxaliaiesLD") = txtMasxaliaiesLD.Text
dsNewRow.Item("allaR") = chkAllaR.Checked
dsNewRow.Item("allaL") = chkAllaL.Checked
dsNewRow.Item("allaRD") = txtAllaRD.Text
dsNewRow.Item("allaLD") = txtAllaLD.Text
dsNewRow.Item("kakoithinososCYN") = cmbKakoithinososYN.Text
dsNewRow.Item("kakoithinososCRL") = cmbKakoithinososRL.Text
dsNewRow.Item("kakoithinososT") = txtKakoithinosos.Text
dsNewRow.Item("kaloithinososCYN") = cmbKaloithinososYN.Text
dsNewRow.Item("kaloithinososCRL") = cmbKaloithinososRL.Text
dsNewRow.Item("kaloithinososT") = txtKaloithinosos.Text
dsNewRow.Item("alleskakoiitheiesCYN") = cmbAlleskakoiitheiesYN.Text
dsNewRow.Item("alleskakoiitheiesC") = cmbAlleskakoiitheies.Text
dsNewRow.Item("alleskakoiitheiesT") = txtAlleskakoiitheiesT.Text
dsNewRow.Item("allopathologikoCYN") = cmbAllopathologikoYN.Text
dsNewRow.Item("allopathologikoC") = cmbAllopathologiko.Text
dsNewRow.Item("allopathologikoT") = txtAllopathologikoT.Text
dsNewRow.Item("lipsifarmakwnCYN") = cmbLipsifarmakwnYN.Text
dsNewRow.Item("lipsifarmakwnT") = txtLipsifarmakwnT.Text
dsNewRow.Item("allergiesCYN") = cmbAllergiesYN.Text
dsNewRow.Item("allergiesT") = txtAllergiesT.Text
dsNewRow.Item("metaggisiCYN") = cmbMetaggisiYN.Text
dsNewRow.Item("metaggisiT") = txtMetaggisiT.Text

dsDiseasesData.Tables("DiseasesData").Rows.Add(dsNewRow)
da.Update(dsDiseasesData, "DiseasesData")
con.Close()
End Sub

```

```

Public Sub SubmitStadioData(ByVal PatientID As String)
    Dim dsStadioData As New DataSet
    Dim da As New OleDb.OleDbDataAdapter
    con.Open()
    ' Καταχώρηση στοιχείων στο table StadioData
    Dim sql = "SELECT * FROM StadioData WHERE id=" & PatientID & ""
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(dsStadioData, "StadioData")
    Dim cb As New OleDb.OleDbCommandBuilder(da)
    Dim dsNewRow As DataRow
    dsNewRow = dsStadioData.Tables("StadioData").NewRow()
    dsNewRow.Item("id") = PatientID
    dsNewRow.Item("TNMT") = txtTNMT.Text
    dsNewRow.Item("TNMN") = txtTNMN.Text
    dsNewRow.Item("TNMM") = txtTNMM.Text
    dsNewRow.Item("stadio") = cmbStadio.Text
    dsNewRow.Item("porogenes") = txtPorogenes.Text
    dsNewRow.Item("loviakos") = txtLoviakos.Text
    dsNewRow.Item("thilis") = txtThilis.Text
    dsNewRow.Item("istoallo") = txtIstoallo.Text
    dsNewRow.Item("apoleiavarous") = txtApoleiavarous.Text
    dsNewRow.Item("dispnoia") = txtDispnoia.Text
    dsNewRow.Item("vixas") = txtVixas.Text
    dsNewRow.Item("algos") = txtAlgos.Text
    dsNewRow.Item("metastallo") = txtMetastallo.Text
    dsNewRow.Item("anapneustikoF") = chkAnapneustikoF.Checked
    dsNewRow.Item("anapneustikoO") = chkAnapneustikoO.Checked
    dsNewRow.Item("anapneustikoM") = chkAnapneustikoM.Checked
    dsNewRow.Item("ostaF") = chkOstaF.Checked
    dsNewRow.Item("ostaO") = chkOstaO.Checked
    dsNewRow.Item("ostaM") = chkOstaM.Checked

```

```

dsNewRow.Item("koiliaF") = chkKoiliaF.Checked
dsNewRow.Item("koiliaO") = chkKoiliaO.Checked
dsNewRow.Item("koiliaM") = chkKoiliaM.Checked
dsNewRow.Item("KNSF") = chkKNSF.Checked
dsNewRow.Item("KNSO") = chkKNSO.Checked
dsNewRow.Item("KNSM") = chkKNSM.Checked
dsNewRow.Item("allaF") = chkAllaF.Checked
dsNewRow.Item("allaO") = chkAllaO.Checked
dsNewRow.Item("allaM") = chkAllaM.Checked
dsNewRow.Item("xeirourgikiC") = cmbXeirourgiki.Text
dsNewRow.Item("sumpliromatikitherapeiaC") =
cmbSumpliromatikitherapeia.Text
dsNewRow.Item("katastasiC") = cmbKatastasi.Text
dsNewRow.Item("simeioypotropis") = txtSimeioypotropis.Text
dsStadioData.Tables("StadioData").Rows.Add(dsNewRow)
da.Update(dsStadioData, "StadioData")
con.Close()
End Sub

Public Sub SubmitObserveData(ByVal PatientID As String)
Dim sql As String = "SELECT * FROM ObserveData WHERE id=" &
PatientID & ""
Dim dsObserveData As New DataSet
Dim regDate As Date = Date.Today
con.Open()
Dim da As New OleDb.OleDbDataAdapter
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(dsObserveData, "ObserveData")
Dim cb As New OleDb.OleDbCommandBuilder(da)
Dim dsNewRow As DataRow
dsNewRow = dsObserveData.Tables("ObserveData").NewRow()
dsNewRow.Item("id") = PatientID
dtpObserve.Value = New DateTime(dtpObserve.Value.Year,
dtpObserve.Value.Month, dtpObserve.Value.Day, DateTime.Now.Hour,
DateTime.Now.Minute, DateTime.Now.Second)
dsNewRow.Item("dateO") = dtpObserve.Value
dsNewRow.Item("minesmetaC") = cmbMinesmeta.Text
dsNewRow.Item("trauma") = txtTrauma.Text
dsNewRow.Item("masxali") = txtMasxali.Text
dsNewRow.Item("sustoixo") = txtSustoixo.Text
dsNewRow.Item("plastiki") = txtPlastiki.Text
dsNewRow.Item("aisthanetai") = txtAisthanetai.Text
dsNewRow.Item("Hct") = txtHct.Text
dsNewRow.Item("Hb") = txtHb.Text
dsNewRow.Item("WBC") = txtWBC.Text
dsNewRow.Item("NEUTR") = txtNEUTR.Text
dsNewRow.Item("LYMPH") = txtLYMPH.Text
dsNewRow.Item("PLT") = txtPLT.Text
dsNewRow.Item("SGOT") = txtSGOT.Text
dsNewRow.Item("GGT") = txtGGT.Text
dsNewRow.Item("ALP") = txtALP.Text
dsNewRow.Item("LDH") = txtLDH.Text
dsNewRow.Item("CA125") = txtCA125.Text
dsNewRow.Item("CA153") = txtCA153.Text
dsNewRow.Item("CA199") = txtCA199.Text
dsNewRow.Item("CEA") = txtCEA.Text
dsNewRow.Item("mastografia") = txtMastografia.Text
dsNewRow.Item("ypermastwn") = txtYpermastwn.Text
dsNewRow.Item("yperanokatokoilias") = txtYperanokatokoilias.Text
dsNewRow.Item("aktinothorakos") = txtAktinothorakos.Text
dsNewRow.Item("aksonikithorakos") = txtAksonikithorakos.Text

```

```

dsNewRow.Item("aksonikikoilias") = txtAksonikikoilias.Text
dsNewRow.Item("spinthirografima") = txtSpinthirografima.Text
dsObserveData.Tables("ObserveData").Rows.Add(dsNewRow)
da.Update(dsObserveData, "ObserveData")
con.Close()

FillDGVs(dgvObserve, "Query3", PatientID) 'Για να εμφανιστεί η νέα
καταχώρηση.
dgvObserve.ClearSelection() 'Εξεπιλέγει τις επιλεγμένες σειρές.
dgvObserve.Rows(dgvObserve.RowCount - 1).Selected = True 'Επιλέγει
την τελευταία σειρά.
End Sub

Private Sub dgvClinical_CellDoubleClick(ByVal sender As Object, ByVal e
As System.Windows.Forms.DataGridViewCellEventArgs) Handles
dgvClinical.CellDoubleClick
RetrieveClinicalData(dgvClinical.SelectedRows(0).Cells(1).Value)

End Sub

Private Sub dgvDiagnosis_CellDoubleClick(ByVal sender As Object, ByVal e
As System.Windows.Forms.DataGridViewCellEventArgs) Handles
dgvDiagnosis.CellDoubleClick
RetrieveDiagnosisData(dgvDiagnosis.SelectedRows(0).Cells(1).Value)
RetrieveCoordinatesData(dgvDiagnosis.SelectedRows(0).Cells(1).Value)
End Sub

Private Sub dgvObserve_CellDoubleClick(ByVal sender As Object, ByVal e
As System.Windows.Forms.DataGridViewCellEventArgs) Handles
dgvObserve.CellDoubleClick
RetrieveObserveData(dgvObserve.SelectedRows(0).Cells(1).Value)
End Sub

Private Sub loadDataOnDoubleClick()
FillDGVs(dgvClinical, "Query1", rowID)
FillDGVs(dgvDiagnosis, "Query2", rowID) 'Γέμισμα των DGVs
FillDGVs(dgvObserve, "Query3", rowID) '
RetrieveDiseasesData()
RetrieveDangersData() 'Ανάκτηση δεδομένων για τις καρτέλες με
μονές εγγραφές.
RetrieveStadioData()
If dgvClinical.RowCount > 0 Then
Dim clinicalID As Integer =
dgvClinical.Rows(dgvClinical.RowCount - 1).Cells(1).Value
RetrieveClinicalData(clinicalID)
' Ανάκτηση δεδομένων για την καρτέλα ClinicalData.
Else
NoRecords = NoRecords & "Κλινικές εξετάσεις" & vbNewLine
End If
If dgvDiagnosis.RowCount > 0 Then
Dim diagnosisID As Integer =
dgvDiagnosis.Rows(dgvDiagnosis.RowCount - 1).Cells(1).Value
RetrieveDiagnosisData(diagnosisID)
RetrieveCoordinatesData(diagnosisID)
Else
NoRecords = NoRecords & "Διαγνωστικός έλεγχος" & vbNewLine
End If
If dgvObserve.RowCount > 0 Then

Dim observeID As Integer = dgvObserve.Rows(dgvObserve.RowCount
- 1).Cells(1).Value
RetrieveObserveData(observeID)

```

```

Else
    NoRecords = NoRecords & "Παρακολούθηση Ασθενων με καρκίνο
μαστού"
End If
    If Len(NoRecords) <> 0 Then
        MsgBox("Οι παρακάτω καρτέλες δεν περιέχουν εγγραφές:" & Chr(13)
& Chr(13) & NoRecords, , "")
        NoRecords = String.Empty '--> Για να είναι καθαρή η μεταβλητή
για το επόμενο DoubleClick στο DT.
    End If
End Sub

Private Sub RetrieveDiseasesData()
    Dim sql As String = "SELECT * FROM DiseasesData where id=" & rowID
& ""
    Dim dsDiseasesData As New DataSet
    con.Open()
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.MissingSchemaAction = MissingSchemaAction.AddWithKey
    da.Fill(dsDiseasesData)
    If dsDiseasesData.Tables(0).Rows.Count > 0 Then
        chkAsymptomatikiR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("asymptomatikiR")
        chkSymptomatikiR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("SymptomatikiR")
        chkMasxaliaiesL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("MasxaliaiesL")
        chkMasxaliaiesR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("MasxaliaiesR")
        chkAllaL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("AllaL")
        chkAllaR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("AllaR")
        chkPonoskiR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("PonoskiR")
        chkPonoskiL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("PonoskiL")
        chkDermatikiparamorfosiL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiL")
        chkDermatikiparamorfosiR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiR")
        chkPonosmikiL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("PonosmikiL")
        chkPonosmikiR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("PonosmikiR")
        chkAnomaliathilisR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("AnomaliathilisR")
        chkAnomaliathilisL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("AnomaliathilisL")
        chkMiaimatiroL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("MiaimatiroL")
        chkMiaimatiroR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("MiaimatiroR")
        chkDenanaferetaiL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("DenanaferetaiL")
        chkDenanaferetaiR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("DenanaferetaiR")
        chkAimatiroR.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("DenanaferetaiR")
        chkOgidioL.Checked =
dsDiseasesData.Tables(0).Rows.Find(rowID).Item("OgidioL")
    End If
End Sub

```

```

        chkOgidioR.Checked =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("OgidioR")
        chkAimatiroL.Checked =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AimatiroL")
        chkSymptomatikiL.Checked =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("SymptomatikiL")
        chkAsymptomatikiL.Checked =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AsymptomatikiL")
        txtAlleskakoitheiesT.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AlleskakoitheiesT").ToString
        txtAllaLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AllaLD")
        txtAllaRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AllaRD")
        txtMasxaliaiesLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("MasxaliaiesLD")
        txtMasxaliaiesRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("MasxaliaiesRD")
        txtPonoskiLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("PonoskiLD")
        txtPonoskiRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("PonoskiRD")
        txtPonosmikiLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("PonosmikiLD")
        txtPonosmikiRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("PonosmikiRD")
        txtDermatikiparamorfosiLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiLD")
        txtDermatikiparamorfosiRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("DermatikiparamorfosiRD")
        txtAnomaliathilisLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AnomaliathilisLD")
        txtAnomaliathilisRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AnomaliathilisRD")
        txtDenanaferetaiLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("DenanaferetaiLD")
        txtDenanaferetaiRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("DenanaferetaiRD")
        txtMiaimatiroLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("MiaimatiroLD")
        txtMiaimatiroRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("MiaimatiroRD")
        txtAimatiroLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AimatiroLD")
        txtAimatiroRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AimatiroRD")
        txtOgidioLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("OgidioLD")
        txtOgidioRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("OgidioRD")
        txtSymptomatikiLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("SymptomatikiLD")
        txtSymptomatikiRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("SymptomatikiRD")
        txtAsymptomatikiLD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AsymptomatikiLD")
        txtAsymptomatikiRD.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AsymptomatikiRD")
        txtMetaggisiT.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("MetaggisiT")

```

```

        txtAllergiesT.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AllergiesT")
        txtLipsifarmakwnT.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("LipsifarmakwnT")
        txtAllopathologikoT.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AllopathologikoT")
        txtKakoithinosos.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("kakoithinososT")
        txtKaloithinosos.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("kaloithinososT")
        cmbAllergiesYN.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("allergiesCYN")
        cmbMetaggisiYN.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("MetaggisiCYN")
        cmbLipsifarmakwnYN.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("LipsifarmakwnCYN")
        cmbAllopathologiko.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AllopathologikoC")
        cmbAllopathologikoYN.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AllopathologikoCYN")
        cmbAlleskakoiitheies.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AlleskakoiitheiesC")
        cmbAlleskakoiitheiesYN.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("AlleskakoiitheiesCYN")
        cmbKakoithinososYN.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("kakoithinososCYN")
        cmbKakoithinososRL.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("kakoithinososCRL")
        cmbKaloithinososYN.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("kaloithinososCYN")
        cmbKaloithinososRL.Text =
dsDeseasesData.Tables(0).Rows.Find(rowID).Item("kaloithinososCRL")
    Else
        NoRecords = "Παρούσα Νόσος / Ατομικό Αναμνηστικό" & vbCrLf
    End If
    con.Close()
End Sub

Private Sub RetrieveDangersData()
    Dim sql = "SELECT * FROM DangersData where id=" & rowID & ""
    Dim dsDangersData As New DataSet
    con.Open()
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.MissingSchemaAction = MissingSchemaAction.AddWithKey
    da.Fill(dsDangersData)
    If dsDangersData.Tables(0).Rows.Count > 0 Then
        txtEidosKarkinouVathmosSiggenias.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("EidosKarkinouVathmosSiggenia
s")
        txtHlikiaOlokliromenisKuisis.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("HlikiaOlokliromenisKuisis")
        txtArithmosKuisseon.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ArithmosKuisseon")
        txtApovoles.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("Apovoles")
        txtHlikiaaminopausis.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("Hlikiaaminopausis")
        txtHlikiaaminarxis.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("Hlikiaaminarxis")
    End If
End Sub

```



```

        txtProsfatiLipsiOrmononPillDur.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononPillDur"
).ToString
        txtProsfatiLipsiOrmononPill.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononPill")
        txtProsfatiLipsiOrmononOtherDur.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononOtherDur
")
        txtProsfatiLipsiOrmononOther.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononOther")
        txtProsfatiLipsiOrmononEidos.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononEidos")
        txtProigoumeniLipsiOrmononEidos.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononEidos
")
        txtProigoumeniLipsiOrmononPill.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononPill"
)
        txtProigoumeniLipsiOrmononOtherDur.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononOther
Dur")
        txtProigoumeniLipsiOrmononOther.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononOther
")
        txtProigoumeniLipsiOrmononPillDur.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononPillD
ur")
        txtMitera.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("Mitera")
        txtPeriptoseisTheies.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("PeriptoseisTheies")
        txtAlloi.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("Alloi")
        txtAdelfi.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("Adelfi")
        txtPeriptoseisAdelfes.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("PeriptoseisAdelfes")
        txtTheiaApoPatera.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("TheiaApoPatera")
        txtTheiaDeuterouVathmou.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("TheiaDeuterouVathmou")
        txtGiagiaApoMitera.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("GiagiaApoMitera")
        txtGiagiaApoPatera.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("GiagiaApoPatera")
        txtProgiagia.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("Progiagia")
        txtTheiaApoMitera.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("TheiaApoMitera")
        cmbOikogeniakoIstorikoAllonKarkinonYN.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("OikogeniakoIstorikoAllonKark
inonYN")
        cmbDiakopiThilasmouYN.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("DiakopiThilasmouYN")
        cmbThilasateYN.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ThilasateYN")
        cmbProigoumeniLipsiOrmononYN.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProigoumeniLipsiOrmononYN")
        cmbProsfatiLipsiOrmononYN.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("ProsfatiLipsiOrmononYN")

```

```

        cmbOikogeniakoIstorikoKarkinouMastouYN.Text =
dsDangersData.Tables(0).Rows.Find(rowID).Item("OikogeniakoIstorikoKarkinouM
astouYN")
    Else
        NoRecords = NoRecords & "Παράγοντες κινδύνου" & vbNewLine
    End If
    con.Close()
End Sub

Private Sub RetrieveStadioData()
    Dim sql = "SELECT * FROM StadioData where id=" & rowID & ""
    Dim dsStadioData As New DataSet
    con.Open()
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.MissingSchemaAction = MissingSchemaAction.AddWithKey
    da.Fill(dsStadioData)

    If dsStadioData.Tables(0).Rows.Count > 0 Then
        txtTNMT.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("TNMT")
        txtTNMN.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("TNMN")
        txtTNMM.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("TNMM")
        cmbStadio.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("stadio")
        txtPorogenes.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("porogenes")
        txtLoviakos.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("loviakos")
        txtThilis.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("thilis")
        txtIstoallo.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("istoallo")
        txtApoieiavarous.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("apoleiavarous")
        txtDispnoia.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("dispnoia")
        txtVixas.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("vixas")
        txtAlgos.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("algos")
        txtMetastallo.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("metastallo")
        chkAnapneustikoF.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("anapneustikoF")
        chkAnapneustikoO.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("anapneustikoO")
        chkAnapneustikoM.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("anapneustikoM")
        chkOstaF.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("ostaF")
        chkOstaO.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("ostaO")
        chkOstaM.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("ostaM")
        chkKoiliaF.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("koiliaF")
        chkKoiliaO.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("koiliaO")
    End If
End Sub

```

```

        chkKoiliaM.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("koiliaM")
        chkKNSF.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("KNSF")
        chkKNSO.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("KNSO")
        chkKNSM.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("KNSM")
        chkAllaF.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("allaF")
        chkAllaO.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("allaO")
        chkAllaM.Checked =
dsStadioData.Tables(0).Rows.Find(rowID).Item("allaM")

        cmbXeirourgiki.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("xeirourgikiC")
        cmbSumpliromatikitherapeia.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("sumpliromatikitherapeiaC")
        cmbKatastasi.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("katastasiC")
        txtSimeioypotropis.Text =
dsStadioData.Tables(0).Rows.Find(rowID).Item("simeioypotropis")
    Else
        NoRecords = NoRecords & "Σταδιοποίηση Καρκίνου" & vbCrLf
    End If
    con.Close()
End Sub

Private Sub RetrieveClinicalData(ByVal clinicalID As Integer)
    Dim sql = "SELECT * FROM ClinicalData where clinicalID=" &
clinicalID & " '
    Dim dsClinicalData As New DataSet
    con.Open()
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.MissingSchemaAction = MissingSchemaAction.AddWithKey
    da.Fill(dsClinicalData)
    cmbProigoumenesEpemvaseisYN.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("ProigoumenesEpemvaseis
YN").ToString
    txtYperklidioiMegethos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("YperklidioiMegethos").
ToString
    txtYpoklidioiMegethos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("YpoklidioiMegethos").T
oString
    txtMasxaliaioiAkinitoiMegethos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiAkinitoiMeg
ethos").ToString
    txtMasxaliaioiKinitoiMegethos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiKinitoiMege
thos").ToString
    txtAnomaliesThilaiasEidos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("AnomaliesThilaiasEidos
").ToString
    txtTopikaOksidiaMegethos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("TopikaOksidiaMegethos"
).ToString
    cmbKlinikaEurimataYN.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("KlinikaEurimataYN").To
String

```

```

        txtSumpagesOgidioMegethos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("SumpagesOgidioMegethos
").ToString
        txtKistikoOgidioMegethos.Text =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("KistikoOgidioMegethos"
).ToString
        chkSimfietaiPeritoniaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("SimfietaiPeritoniaL")
        chkSimfietaiPeritoniaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("SimfietaiPeritoniaR")
        chkSimfietaiThorakikoTixomaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("SimfietaiThorakikoTixo
maL")
        chkSimfietaiThorakikoTixomaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("SimfietaiThorakikoTixo
maR")
        chkEisolikiDermatosL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EisolikiDermatosL")
        chkPeauDOrangeL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("PeauDOrangeL")
        chkPeauDOrangeR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("PeauDOrangeR")
        chkEisolikiDermatosR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EisolikiDermatosR")
        chkDiithisiDermatosL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("DiithisiDermatosL")
        chkDiithisiDermatosR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("DiithisiDermatosR")
        chkEisolikiThilisR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EisolikiThilisR")
        chkMasxaliaioiKinitoiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiKinitoiL")
        chkMasxaliaioiKinitoiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiKinitoiR")
        chkMasxaliaioiAkinitoiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiAkinitoiL")
        chkMasxaliaioiAkinitoiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaioiAkinitoiR")
        chkYperklidioiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("YperklidioiL")
        chkYperklidioiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("YperklidioiR")
        chkYpoklidioiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("YpoklidioiL")
        chkYpoklidioiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("YpoklidioiR")
        chkEsxaraThilisL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EsxaraThilisL")
        chkEsxaraThilisR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EsxaraThilisR")
        chkErithrotitaThilisL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("ErithrotitaThilisL")
        chkErithrotitaThilisR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("ErithrotitaThilisR")
        chkAimatiroEkrimaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("AimatiroEkrimaL")
        chkAnomaliesThiliasL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("AnomaliesThiliasL")
        chkAnomaliesThiliasR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("AnomaliesThiliasR")

```

```

        chkAimatiroEkrimaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("AimatiroEkrimaR")
        chkMiAimatiroEkrimaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MiAimatiroEkrimaL")
        chkMiAimatiroEkrimaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MiAimatiroEkrimaR")
        chkEisolikiThilisL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EisolikiThilisL")
        chkSumpagesOgidioL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("SumpagesOgidioL")
        chkSumpagesOgidioR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("SumpagesOgidioR")
        chkDiaxitaOksidiaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("DiaxitaOksidiaL")
        chkDiaxitaOksidiaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("DiaxitaOksidiaR")
        chkTopikaOksidiaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("TopikaOksidiaL")
        chkTopikiEuesthisiaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("TopikiEuesthisiaL")
        chkTopikiEuesthisiaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("TopikiEuesthisiaR")
        chkTopikaOksidiaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("TopikaOksidiaR")
        chkDiaxitiEuesthisiaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("DiaxitiEuesthisiaL")
        chkDiaxitiEuesthisiaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("DiaxitiEuesthisiaR")
        chkKistikoOgidioL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("KistikoOgidioL")
        chkKistikoOgidioR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("KistikoOgidioR")
        chkEpanepemvasiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EpanepemvasiL")
        chkEpanepemvasiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("EpanepemvasiR")
        chkMasxaliaiosKatharismosL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaiosKatharismos
L")
        chkMasxaliaiosKatharismosR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("MasxaliaiosKatharismos
R")
        chkViopsiaL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("ViopsiaL")
        chkViopsiaR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("ViopsiaR")
        chkOgkektomiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("OgkektomiL")
        chkOlikiMastektomiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("OlikiMastektomiR")
        chkOgkektomiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("OgkektomiR")
        chkOlikiMastektomiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("OlikiMastektomiL")
        chkTmimatikiMastektomiL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("TmimatikiMastektomiL")
        chkTmimatikiMastektomiR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("TmimatikiMastektomiR")
        chkOuliR.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("OuliR")

```

```

        chkOuliL.Checked =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("OuliL")
        If Not
IsDBNull(dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("IkonidiaC"))
Then
            cmbIkonidia.SelectedIndex =
dsClinicalData.Tables(0).Rows.Find(clinicalID).Item("IkonidiaC")
            cmbIkonidia.Update()
        Else
            cmbIkonidia.SelectedIndex = -1
        End If
        con.Close()
    End Sub

    Private Sub RetrieveDiagnosisData(ByVal diagnosisID As Integer)
        Dim sql = "SELECT * FROM DiagnosisData WHERE diagnosisID=" &
diagnosisID & ""
        Dim dsDiagnosisData As New DataSet
        con.Open()
        Dim da = New OleDb.OleDbDataAdapter(sql, con)
        da.MissingSchemaAction = MissingSchemaAction.AddWithKey
        da.Fill(dsDiagnosisData)
        chkViopsiaAneparkisL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaAneparkisL")
        chkViopsiaThesiL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaThesiL")
        chkViopsiaKakoithiR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKakoithiR")
        chkViopsiaKaloithiR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKaloithiR")
        chkViopsiaKakoithiL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKakoithiL")
        chkViopsiaAneparkisR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaAneparkisR")
        chkViopsiaKaloithiL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaKaloithiL")
        chkViopsiaThesiR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ViopsiaThesiR")
        chkYperixosFisiologikiR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosFisiologikiR
")
        chkYperixosFisiologikiL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosFisiologikiL
")
        chkYperixosDisplasiaL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosDisplasiaL")
        chkYperixosDisplasiaR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosDisplasiaR")
        chkYperixosApotitanoseisL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosApotitanosei
sL")
        chkYperixosApotitanoseisR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosApotitanosei
sR")
        chkYperixosKakoithiaR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosKakoithiaR")
        chkYperixosKakoithiaL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosKakoithiaL")
        chkYperixosYpoptoR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosYpoptoR")

```

```

        chkYperixosKaloithiaR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosKaloithiaR")
        chkYperixosYpoptoL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosYpoptoL")
        chkYperixosKaloithiaL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosKaloithiaL")
        chkApotitanoseisEntosR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEntosR"
)
        chkApotitanoseisEntosL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEntosL"
)
        chkApotitanoseisEktosR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEktosR"
)
        chkApotitanoseisEktosL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("ApotitanoseisEktosL"
)
        chkMastografiaFisiologikiR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaFisiologi
kiR")
        chkMastografiaFisiologikiL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaFisiologi
kiL")
        chkMastografiaDisplasiaL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDisplasia
L")
        chkMastografiaDisplasiaR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDisplasia
R")
        chkMastografiaApotitanoseisL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaApotitano
seisL")
        chkMastografiaApotitanoseisR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaApotitano
seisR")
        chkMastografiaKakoithiaR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKakoithia
R")
        chkMastografiaKakoithiaL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKakoithia
L")
        chkMastografiaYpoptoR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaYpoptoR")
        chkMastografiaKaloithiaR.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKaloithia
R")
        chkMastografiaYpoptoL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaYpoptoL")
        chkMastografiaKaloithiaL.Checked =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaKaloithia
L")
        txtFNAAristeraKitarologikiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNAAristeraKitarolog
ikiL").ToString
        txtFNAAristeraThesiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNAAristeraThesiL")
.ToString
        txtFNAAristeraThesiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNAAristeraThesiR")
.ToString

```

```

        txtFNADexiaKitarologikiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaKitarologiki
R").ToString
        txtFNADexiaKitarologikiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaKitarologiki
L").ToString
        txtFNADexiaThesiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaThesiL").ToS
tring
        txtFNADexiaThesiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaThesiR").ToS
tring
        txtYperixosAristeraMegethosR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraMege
thosR").ToString
        txtYperixosAristeraMegethosL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraMege
thosL").ToString
        txtYperixosAristeraThesiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraThes
iL").ToString
        txtYperixosAristeraThesiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosAristeraThes
iR").ToString
        txtYperixosDexiaMegethosR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaMegetho
sR").ToString
        txtYperixosDexiaMegethosL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaMegetho
sL").ToString
        txtYperixosDexiaThesiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaThesiL"
)'.ToString
        txtYperixosDexiaThesiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("YperixosDexiaThesiR"
).ToString
        txtMastografiaAristeraMegethosR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraM
egethosR").ToString
        txtMastografiaAristeraMegethosL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraM
egethosL").ToString
        txtMastografiaAristeraThesiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraT
hesiL").ToString
        txtMastografiaAristeraThesiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaAristeraT
hesiR").ToString
        txtMastografiaDexiaMegethosR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaMege
thosR").ToString
        txtMastografiaDexiaMegethosL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaMege
thosL").ToString
        txtMastografiaDexiaThesiL.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaThes
iL").ToString
        txtMastografiaDexiaThesiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaDexiaThes
iR").ToString

```



```

        cmbMastografiaApotitanoseisYN.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("MastografiaApotitano
seisYN").ToString
        txtFNADexiaKitarologikiR.Text =
dsDiagnosisData.Tables(0).Rows.Find(diagnosisID).Item("FNADexiaKitarologiki
R").ToString
        con.Close()
    End Sub

    Private Sub RetrieveObserveData(ByVal observeID As Integer)
        Dim sql = "SELECT * FROM ObserveData where observeID=" & observeID
& ""
        Dim dsObserveData As New DataSet
        con.Open()
        Dim da = New OleDb.OleDbDataAdapter(sql, con)
        da.MissingSchemaAction = MissingSchemaAction.AddWithKey
        da.Fill(dsObserveData)
        cmbMinesmeta.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("minesmetaC")
        txtTrauma.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("trauma")
        txtMasxali.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("masxali")
        txtSustoixo.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("sustoixo")
        txtPlastiki.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("plastiki")
        txtAisthanetai.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("aisthanetai")
        txtHct.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("Hct")
        txtHb.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("Hb")
        txtWBC.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("WBC")
        txtNEUTR.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("NEUTR")
        txtLYMPH.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("LYMPH")
        txtPLT.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("PLT")
        txtSGOT.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("SGOT")
        txtGGT.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("GGT")
        txtALP.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("ALP")
        txtLDH.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("LDH")
        txtCA125.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("CA125")
        txtCA153.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("CA153")
        txtCA199.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("CA199")
        txtCEA.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("CEA")
        txtMastografia.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("mastografia")
        txtYpermastwn.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("ypermastwn")

```

```

        txtYperanokatokoilias.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("yperanokatokoilias")
        txtAktinothorakos.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("aktinothorakos")
        txtAksonikithorakos.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("aksonikithorakos")
        txtAksonikikoilias.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("aksonikikoilias")
        txtSpinthirografima.Text =
dsObserveData.Tables(0).Rows.Find(observeID).Item("spinthirografima")
        con.Close()
    End Sub

    Public Sub SubmitCoordinatesData(ByVal diagID As String)
        Dim dsNewRow As DataRow
        Dim dsCoordinatesData As New DataSet
        Dim sql = "SELECT * FROM CoordinatesData "
        Dim da = New OleDb.OleDbDataAdapter(sql, con)
        da.Fill(dsCoordinatesData, "CoordinatesData")
        con.Open()
        Dim cb As New OleDb.OleDbCommandBuilder(da)
        For i As Integer = 0 To 9
            dsNewRow = dsCoordinatesData.Tables("CoordinatesData").NewRow()
            dsNewRow.Item("diagnosisID") = diagID
            dsNewRow.Item("mpx") = coordlist(1, i)
            dsNewRow.Item("mpy") = coordlist(0, i)
            dsNewRow.Item("paxos") = pinelo(i)
            dsCoordinatesData.Tables("CoordinatesData").Rows.Add(dsNewRow)
            da.Update(dsCoordinatesData, "CoordinatesData")
        Next
        con.Close()
    End Sub

    Private Sub RetrieveCoordinatesData(ByVal diagID As Integer)
        Dim sql = "SELECT * FROM CoordinatesData WHERE diagnosisID=" &
diagID & ""
        Dim dsCoordinatesData As New DataSet
        con.Open()
        Dim da = New OleDb.OleDbDataAdapter(sql, con)
        da.MissingSchemaAction = MissingSchemaAction.AddWithKey
        da.Fill(dsCoordinatesData)
        i = 0
        For i As Integer = 0 To
dsCoordinatesData.Tables(0).DefaultView.Count - 1
            coordlist(1, i) =
dsCoordinatesData.Tables(0).Rows(i).Item("mpx")
            coordlist(0, i) =
dsCoordinatesData.Tables(0).Rows(i).Item("mpy")
            pinelo(i) = dsCoordinatesData.Tables(0).Rows(i).Item("paxos")
            PictureBox2.Invalidate()
        Next
        'Application.DoEvents()
        'PictureBox2.Refresh()
        con.Close()
    End Sub

    Private Sub resetCoordinates()
        For count As Integer = 0 To 9
            coordlist(1, count) = 0
            coordlist(0, count) = 0
            pinelo(count) = 0
        Next
    End Sub

```

```

        PictureBox2.Invalidate()
        i = 0
    Next
End Sub
Private Sub TxtName_ModifiedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TxtName_NeosAsthenis.ModifiedChanged
    If TxtName_NeosAsthenis.Text = "" Then
        txtAge_NeosAsthenis.Enabled = False
        txtAddress_NeosAsthenis.Enabled = False
        txtPhone_NeosAsthenis.Enabled = False
        btnNewRecord.Enabled = False
        btnReset.Enabled = False
    End If
    txtAge_NeosAsthenis.Enabled = True
    txtAddress_NeosAsthenis.Enabled = True
    txtPhone_NeosAsthenis.Enabled = True
    btnNewRecord.Enabled = True
    btnReset.Enabled = True
End Sub

Private Sub TabControlBasicData_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
TabControlBasicData.SelectedIndexChanged
    If TabControlBasicData.SelectedIndex = 0 Then
        PanelSearch.Visible = False
        TabControlKartel.Visible = False
        btnReturn_Click(sender, New System.EventArgs()) ' etsi
caloume leitourgia tou button
        TabControlKartel.SelectedIndex = 0
    ElseIf TabControlBasicData.SelectedIndex = 1 Then
        Cursor = Cursors.WaitCursor
        PanelSearch.Visible = True
        TabControlKartel.Visible = False
        btnSaveAll.Visible = False
        btnReturn2.Visible = False
        cmbYparxonOnoma.Visible = True
        txtYparxonOnoma.Visible = False
        btnReset_Click(sender, New System.EventArgs())
        TabControlKartel.SelectTab(TabDisease)
        loadComboData()
        Cursor = Cursors.Default()
    End If
End Sub

Private Sub TabControlKartel_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles TabControlKartel.SelectedIndexChanged
    If TabControlKartel.SelectedIndex = 2 And PanelSearch.Visible =
False Then
        If TabControlBasicData.SelectedIndex = 1 Then
            GroupBoxClinicDate.Visible = False
            GroupBoxDiagnosis.Visible = False
            gbxObserve.Visible = False
        End If
        GroupBoxClinicDate.Visible = True
        GroupBoxDiagnosis.Visible = False
        gbxObserve.Visible = False

        ElseIf TabControlKartel.SelectedIndex = 3 And PanelSearch.Visible =
False Then
            If TabControlBasicData.SelectedIndex = 1 Then
                GroupBoxClinicDate.Visible = False
            End If
        End If
    End Sub

```

```

        GroupBoxDiagnosis.Visible = False
        gbxObserve.Visible = False
    End If
    GroupBoxClinicDate.Visible = False
    GroupBoxDiagnosis.Visible = True
    gbxObserve.Visible = False
    ElseIf TabControlKartel.SelectedIndex = 5 And PanelSearch.Visible =
False Then
        If TabControlBasicData.SelectedIndex = 1 Then
            GroupBoxClinicDate.Visible = False
            GroupBoxDiagnosis.Visible = False
            gbxObserve.Visible = False
        End If
        GroupBoxClinicDate.Visible = False
        GroupBoxDiagnosis.Visible = False
        gbxObserve.Visible = True
    Else
        GroupBoxClinicDate.Visible = False
        GroupBoxDiagnosis.Visible = False
        gbxObserve.Visible = False
    End If
End Sub

'Oi methodoi Reset
Private Sub ClearTab(ByVal TabToClear As TabPage)
    Dim ctl As Object
    For Each ctl In TabToClear.Controls 'στοιχείαNeouAsthenous.Controls
        If ctl.GetType.Name = "GroupBox" Then
            Dim gbxCtls As Object
            For Each gbxCtls In ctl.controls
                If TypeOf gbxCtls Is TextBox Then
                    gbxCtls.Text = ""
                ElseIf TypeOf gbxCtls Is CheckBox Then
                    gbxCtls.checked = False
                ElseIf TypeOf gbxCtls Is ComboBox Then
                    gbxCtls.SelectedIndex = -1
                End If
            Next
        End If
        If TypeOf ctl Is TextBox Then
            ctl.Text = ""
        ElseIf TypeOf ctl Is CheckBox Then
            ctl.checked = False
        ElseIf TypeOf ctl Is ComboBox Then
            ctl.SelectedIndex = -1
        End If
    Next
End Sub

'Leitourgia interface
Private Sub cmbKakoithinososYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbKakoithinososYN.SelectedIndexChanged
    If cmbKakoithinososYN.SelectedItem Is "Ναί" Then
        cmbKakoithinososRL.Enabled = True : txtKakoithinosos.Enabled = True Else
        cmbKakoithinososRL.Enabled = False : cmbKakoithinososRL.SelectedIndex = -
1 : txtKakoithinosos.Enabled = False : txtKakoithinosos.Text = ""
    End Sub

```

```

Private Sub cmbKaloithinososYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbKaloithinososYN.SelectedIndexChanged
    If cmbKaloithinososYN.SelectedItem Is "ΝαΙ" Then
cmbKaloithinososRL.Enabled = True : txtKaloithinosos.Enabled = True Else
cmbKaloithinososRL.Enabled = False : cmbKaloithinososRL.SelectedIndex = -
1 : txtKaloithinosos.Enabled = False : txtKaloithinosos.Text = ""
    End Sub

Private Sub cmbAlleskakoiitheiesYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbAlleskakoiitheiesYN.SelectedIndexChanged
    If cmbAlleskakoiitheiesYN.SelectedItem Is "ΝαΙ" Then
cmbAlleskakoiitheies.Enabled = True Else cmbAlleskakoiitheies.Enabled =
False : cmbAlleskakoiitheies.SelectedIndex = -1 : txtAlleskakoiitheiesT.Text
= "" : txtAlleskakoiitheiesT.Enabled = False
    End Sub

Private Sub cmbAlleskakoiitheies_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbAlleskakoiitheies.SelectedIndexChanged
    If cmbAlleskakoiitheies.SelectedItem Is "'Αλλοζ Τύποζ" Then
txtAlleskakoiitheiesT.Enabled = True Else txtAlleskakoiitheiesT.Enabled =
False : txtAlleskakoiitheiesT.Text = ""
    End Sub

Private Sub cmbAllopathologikoYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbAllopathologikoYN.SelectedIndexChanged
    If cmbAllopathologikoYN.SelectedItem Is "ΝαΙ" Then
cmbAllopathologiko.Enabled = True Else cmbAllopathologiko.Enabled = False :
cmbAllopathologiko.SelectedIndex = -1 : txtAllopathologikoT.Text = ""
    End Sub
    Private Sub cmbAllopathologiko_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbAllopathologiko.SelectedIndexChanged
        If cmbAllopathologiko.SelectedItem Is "Αλλο ζ" Then
txtAllopathologikoT.Enabled = True Else txtAllopathologikoT.Enabled =
False : txtAllopathologikoT.Text = ""
        End Sub

Private Sub cmbLipsifarmakwnYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbLipsifarmakwnYN.SelectedIndexChanged
    If cmbLipsifarmakwnYN.SelectedItem Is "ΝαΙ" Then
txtLipsifarmakwnT.Enabled = True Else txtLipsifarmakwnT.Enabled = False :
txtLipsifarmakwnT.Text = ""
    End Sub

Private Sub cmbAllergiesYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbAllergiesYN.SelectedIndexChanged
    If cmbAllergiesYN.SelectedItem Is "ΝαΙ" Then txtAllergiesT.Enabled
= True Else txtAllergiesT.Enabled = False : txtAllergiesT.Text = ""
    End Sub

Private Sub cmbMetaggisiYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbMetaggisiYN.SelectedIndexChanged
    If cmbMetaggisiYN.SelectedItem Is "ΝαΙ" Then txtMetaggisiT.Enabled
= True Else txtMetaggisiT.Enabled = False : txtMetaggisiT.Text = ""

```

```

End Sub

Private Sub cmbProsfatiLipsiOrmononYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbProsfatiLipsiOrmononYN.SelectedIndexChanged
    If cmbProsfatiLipsiOrmononYN.SelectedItem Is "Να1" Then
gbxProsfatiLipsiOrmonon.Enabled = True Else gbxProsfatiLipsiOrmonon.Enabled
= False : resetgbx(gbxProsfatiLipsiOrmonon)
    End Sub

Private Sub cmbProigoumeniLipsiOrmononYN_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
cmbProigoumeniLipsiOrmononYN.SelectedIndexChanged
    If cmbProigoumeniLipsiOrmononYN.SelectedItem Is "Να1" Then
gbxProigoumeniLipsiOrmonon.Enabled = True Else
gbxProigoumeniLipsiOrmonon.Enabled = False :
resetgbx(gbxProigoumeniLipsiOrmonon)
    End Sub

Private Sub
cmbOikogeniakoIstorikoKarkinouMastouYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbOikogeniakoIstorikoKarkinouMastouYN.SelectedIndexChanged
    If cmbOikogeniakoIstorikoKarkinouMastouYN.SelectedItem Is "Να1"
Then gbxOikogeneiakoIstorikoKarkinouMastouYN.Enabled = True Else
gbxOikogeneiakoIstorikoKarkinouMastouYN.Enabled = False :
resetgbx(gbxOikogeneiakoIstorikoKarkinouMastouYN)
    End Sub

Private Sub
cmbOikogeniakoIstorikoAllonKarkinonYN_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbOikogeniakoIstorikoAllonKarkinonYN.SelectedIndexChanged
    If cmbOikogeniakoIstorikoAllonKarkinonYN.SelectedItem Is "Να1" Then
txtEidosKarkinouVathmosSiggenias.Enabled = True Else
txtEidosKarkinouVathmosSiggenias.Enabled = False :
txtEidosKarkinouVathmosSiggenias.Text = ""
    End Sub
    Private Sub cmbProigoumenesEpemvaseisYN_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
cmbProigoumenesEpemvaseisYN.SelectedIndexChanged
    If cmbProigoumenesEpemvaseisYN.SelectedItem Is "Να1" Then
gbxProigoumenesEpemvaseis.Enabled = True Else
gbxProigoumenesEpemvaseis.Enabled = False :
resetgbx(gbxProigoumenesEpemvaseis)
    End Sub

'koubia prev next kartelwn    Private Sub Button16_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles Button16.Click
    TabControlKartel.SelectedIndex += 1
    End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
    TabControlKartel.SelectedIndex += 1
    End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TabControlBasicData.SelectTab(TabPage2)

```

```

End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
    If TabControlKartel.SelectedIndex > 0 Then
        TabControlKartel.SelectedIndex -= 1
    End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If TabControlKartel.SelectedIndex > 0 Then
        TabControlKartel.SelectedIndex -= 1
    End If
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click
    If TabControlKartel.SelectedIndex > 0 Then
        TabControlKartel.SelectedIndex -= 1
    End If
End Sub

Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
    If TabControlKartel.SelectedIndex > 0 Then
        TabControlKartel.SelectedIndex -= 1
    End If
End Sub

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button13.Click
    If TabControlKartel.SelectedIndex > 0 Then
        TabControlKartel.SelectedIndex -= 1
    End If
End Sub

Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
    TabControlKartel.SelectedIndex += 1
End Sub

Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button18.Click
    TabControlKartel.SelectedIndex += 1
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
    TabControlKartel.SelectedIndex += 1
End Sub

'Menustrip kai koubia toolbox
Private Sub btnReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReset.Click
    ClearTab(TabDesease)
    ClearTab(TabDanger)
    ClearTab(TabClinical)
    ClearTab(TabDiagnosis)
    ClearTab(TabStadio)
    ClearTab(TabObserve)

```

```

        ClearTab(TabPage1)
        resetCoordinates()
    End Sub

    Private Sub ΘέματαΤηςΒοήθειαςToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ΘέματαΤηςΒοήθειαςToolStripMenuItem.Click
        btnHelp_Click(sender, New System.EventArgs())
    End Sub

    Private Sub ΠροβολήΣτατιστικώνToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ΠροβολήΣτατιστικώνToolStripMenuItem.Click
        btnStats_Click(sender, New System.EventArgs())
    End Sub

    Private Sub ΈξοδοςToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ΈξοδοςToolStripMenuItem.Click
        Me.Close()
    End Sub

    Private Sub ΑναζήτησηΑσθενούςToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ΑναζήτησηΑσθενούςToolStripMenuItem.Click
        BtnSearch_Click(sender, New System.EventArgs())
    End Sub

    Private Sub ΕισαγωγήΝεουΑσθενουςToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ΕισαγωγήΝεουΑσθενουςToolStripMenuItem.Click
        TabControlBasicData.SelectTab(TabPage1)
    End Sub

    Private Sub ΕκτύπωσηToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ΕκτύπωσηToolStripMenuItem.Click
        BtnSearch_Click(sender, New System.EventArgs())
        PrintForm.Show()
    End Sub

    Private Sub BtnSearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSearch.Click
        TabControlBasicData.SelectTab(TabPage2)
        TabControlKartel.Visible = False
        TabControlKartel.SelectTab(TabDesease)
        GroupBoxClinicDate.Visible = False
        GroupBoxDiagnosis.Visible = False
        cmbYparxonOnoma.Visible = True
        txtYparxonOnoma.Visible = False
        gbxObserve.Visible = False
        btnSaveAll.Visible = False
        btnReturn2.Visible = False
        ClearTab(TabDesease)
        ClearTab(TabDanger)
        ClearTab(TabClinical)
        ClearTab(TabDiagnosis)
        ClearTab(TabStadio)
        ClearTab(TabObserve)
        ClearTab(TabPage1)
        PanelSearch.Visible = True
    End Sub

```



```

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
    Me.Close()
End Sub

Private Sub ButtonPlus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPlus.Click
    btnReturn_Click(sender, New System.EventArgs())
    TabControlBasicData.SelectTab(TabPage1)
    ClearTab(TabDesease)
    ClearTab(TabDanger)
    ClearTab(TabClinical)
    ClearTab(TabDiagnosis)
    ClearTab(TabStadio)
    ClearTab(TabObserve)
    ClearTab(TabPage1)
End Sub

Private Sub btnReturn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReturn.Click
    btnNewRecord.Text = "Καταχώρηση"
    btnReset_Click(sender, New System.EventArgs())
    gbxStoixeiaNeouAsthenous.Visible = True
    gbxStoixeiaAsthenous.Visible = False
    TabControlKartel.Visible = False
    TabControlKartel.SelectTab(TabDesease)
    btnReturn.Enabled = False
End Sub

Private Sub btnHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHelp.Click
    Dim p As New Process()
    Dim psi As New
ProcessStartInfo(System.AppDomain.CurrentDomain.BaseDirectory() &
"Helpchm.chm")
    psi.Verb = "open"
    p.StartInfo = psi
    p.Start()
End Sub

Private Sub btnViewAndProccess_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnViewAndProccess.Click
    TabControlKartel.Visible = True
    PanelSearch.Visible = False
    TabControlKartel.SelectTab(TabDesease)
    btnReturn2.Visible = True
    btnSaveAll.Visible = True
    cmbYparxonOnoma.Visible = False
    txtYparxonOnoma.Visible = True
    If rowID = Nothing Then
        rowID = DT.Rows(0).Cells(0).Value.ToString
    End If
    loadDataOnDoubleClick()
End Sub

'Λειτουργίες ζωγραφικής σε εικόνα
Private Sub trckPinelo_Scroll(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles trckPinelo.Scroll
    lblPineloValue.Text = trckPinelo.Value.ToString
End Sub

```

```

Private Sub PictureBox2_MouseLeave(ByVal sender As Object, ByVal e As
System.EventArgs) Handles PictureBox2.MouseLeave
    Me.Cursor = Cursors.Default
End Sub

Private Sub PictureBox2_MouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventHandler) Handles PictureBox2.MouseMove
    Dim mycursor As New Icon(Application.StartupPath + "\brush.ico")
    Me.Cursor = New Cursor(mycursor.Handle)
End Sub

Private Sub PictureBox2_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventHandler) Handles PictureBox2.MouseDown
    coord.X = e.X
    coord.Y = e.Y
    If i <= 9 Then
        coordlist(1, i) = coord.X
        coordlist(0, i) = coord.Y
        pinelo(i) = trckPinelo.Value
        PictureBox2.Invalidate()
        'Me.Text = "Mouse position inside the picture box is: x:" &
coordlist(1, i) & " y:" & coordlist(0, i)
        i = i + 1
    Else
        MsgBox("Δεν μπορείτε να τοποθετήσετε πάνω από 10 σημεία")
        i = 0
        PictureBox2.Invalidate()
    End If
End Sub

Private Sub PictureBox2_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PictureBox2.Paint
    Dim pen1 As New System.Drawing.Pen(Color.Red, 100)
    For count As Integer = 0 To 9
        e.Graphics.DrawEllipse(Pens.Red, coordlist(1, count) - 15,
coordlist(0, count) + 5, pinelo(count), pinelo(count))
        e.Graphics.FillEllipse(Brushes.Red, coordlist(1, count) - 15,
coordlist(0, count) + 5, pinelo(count), pinelo(count))
    Next
End Sub

Private Sub btnClearCoordinates_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnClearCoordinates.Click
    resetCoordinates()
End Sub
'koubia diagrafis kai neas kataxorisiss  dgv eksetasewn apo uparxon
astheni

Private Sub btnDeleteClinical_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnDeleteClinical.Click
    DeleteFromDGV(dgvClinical, "ClinicalData", "Query1")
End Sub

Private Sub btnDeleteObserve_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnDeleteObserve.Click
    DeleteFromDGV(dgvObserve, "ObserveData", "Query3")
End Sub

Private Sub btnDeleteDiagnosis_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnDeleteDiagnosis.Click

```

```

        DeleteFromDGV(dgvDiagnosis, "DiagnosisData", "Query2")
    End Sub

Private Sub btnNewClinical_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNewClinical.Click
    SubmitClinicalData(rowID)
    FillDGVs(dgvClinical, "Query1", rowID)
    MsgBox("Η κλινική εξέταση καταχωρήθηκε με επιτυχία")
End Sub

Private Sub btnNewDiagnosis_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNewDiagnosis.Click
    SubmitDiagnosisData(rowID)
    Dim ds As New DataSet
    Dim sql = "SELECT * FROM DiagnosisData WHERE id=" & rowID & ""
    Dim da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "DiagnosisData")
    Dim numofrows As Integer
    numofrows = ds.Tables("DiagnosisData").Rows.Count - 1
    Dim lastid As String
    lastid =
ds.Tables("DiagnosisData").Rows(numofrows).Item("diagnosisID")
    SubmitCoordinatesData(lastid)
    FillDGVs(dgvDiagnosis, "Query2", rowID)
    MsgBox("Ο διαγνωστικός έλεγχος καταχωρήθηκε με επιτυχία")
End Sub

Private Sub btnNewObserve_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNewObserve.Click
    SubmitObserveData(rowID)
    FillDGVs(dgvObserve, "Query3", rowID)
    MsgBox("Η παρακολούθηση ασθενών καταχωρήθηκε με επιτυχία")
End Sub

    Private Sub txtAge_NeosAsthenis_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles txtAge_NeosAsthenis.Click
        checkTxtName_NeosAsthenis(True)
    End Sub
    Private Sub txtAddress_NeosAsthenis_Click(ByVal sender As Object, ByVal
e As System.EventArgs) Handles txtAddress_NeosAsthenis.Click
        checkTxtName_NeosAsthenis(True)
    End Sub
    Private Sub txtPhone_NeosAsthenis_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles txtPhone_NeosAsthenis.Click
        checkTxtName_NeosAsthenis(True)
    End Sub
    Private Sub TxtName_NeosAsthenis_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles TxtName_NeosAsthenis.Click
        TxtName_NeosAsthenis.Modified = False
        checkTxtName_NeosAsthenis(False)
    End Sub

    Private Sub checkTxtName_NeosAsthenis(ByVal flag As Boolean)
        If TxtName_NeosAsthenis.Text = "" Then
            If flag = True Then
                MsgBox("Το πεδίο 'Όνοματεπώνυμο' πρέπει να είναι
συμπληρωμένο")
            End If
            txtAge_NeosAsthenis.Enabled = False
            txtAddress_NeosAsthenis.Enabled = False
            txtPhone_NeosAsthenis.Enabled = False
        End Sub

```

```

        btnNewRecord.Enabled = False
        btnReset.Enabled = False
        txtAge_NeosAsthenis.Text = ""
        txtAddress_NeosAsthenis.Text = ""
        txtPhone_NeosAsthenis.Text = ""
    End If
    flag = False
End Sub

Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    If e.CloseReason = CloseReason.UserClosing Then
        If MessageBox.Show("Είστε σίγουρος ότι θέλετε να τερματίσετε το
πρόγραμμα;", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) =
DialogResult.No Then
            e.Cancel = True
        Else
            Application.Exit()
        End If
    End If
End Sub

Private Sub btnReturn2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReturn2.Click
    BtnSearch_Click(sender, New System.EventArgs())
    ClearTab(TabDesease)
    ClearTab(TabDanger)
    ClearTab(TabClinical)
    ClearTab(TabDiagnosis)
    ClearTab(TabStadio)
    ClearTab(TabObserve)
    ClearTab(TabPage1)
    resetCoordinates()
End Sub

Private Sub DT_CellMouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
DT.CellMouseDown
    If e.RowIndex >= 0 Then 'if για να μην κρασάρει όταν το κλικ
γίνεται σε column header.
        rowIndex = e.RowIndex
        rowID = DT.Rows(e.RowIndex).Cells(0).Value.ToString
        'datarow einai grammi dedomenon to 0 seimainei oti einai o
protos pinakas tou dataset . kai i find pernei mono to proteuon kleidi tou
pinaka
        Dim foundRow As DataRow = dsDT.Tables(0).Rows.Find(rowID)
        ' gia na paro to ID kathe fora me to click oste na to
xrisimopoiiso gia kataxoriseis idonclick eksoteriki metavliti
        rowID = foundRow.Item(0).ToString
        lblID.Text = foundRow.Item(0).ToString
        cmbYparxonOnoma.Text = foundRow.Item("Ονοματεπώνυμο").ToString
        'txtPhoneSearch.Text = foundRow.Item("Τηλέφωνο").ToString
        'txtAddressSearch.Text = foundRow.Item("Διεύθυνση").ToString
        'txtAgeSearch.Text = foundRow.Item("Ημερομηνία
γέννησης").ToString
    End If
End Sub

Private Sub btnStats_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStats.Click
    FormStats.Show()

```

```

End Sub

Private Sub btnPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrint.Click
    PrintForm.Show()
End Sub

Public Sub resetgbx(ByVal gbx As GroupBox)
    Dim cb As CheckBox
    For Each ctrl As Control In gbx.Controls
        If TypeOf ctrl Is CheckBox Then
            cb = CType(ctrl, CheckBox)
            cb.Checked = False
        End If
    Next ctrl
    For Each ctrl As Control In gbx.Controls
        If TypeOf ctrl Is TextBox Then
            CType(ctrl, TextBox).Text = String.Empty
        End If
    Next ctrl
End Sub

Private Sub btnToolPrint_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnToolPrint.Click
    BtnSearch_Click(sender, New System.EventArgs())
    PrintForm.Show()
End Sub

Public Function IsValidAMKA(ByVal AMKANumber) As Boolean '-----
-----function pou tsekarei to amka ean einai egguro
    Dim IsEven As Boolean, i%, iDigit%, iSum%, sTMP$, sChr$
    For i = 1 To Len(AMKANumber)
        sChr = Mid(AMKANumber, i, 1)
        If IsNumeric(sChr) Then sTMP = sTMP & sChr
    Next
    If Len(sTMP) <> 11 Then Exit Function
    For i = 1 To Len(sTMP)
        iDigit = Mid(sTMP, i, 1)
        If IsEven Then
            iDigit = iDigit * 2
            If iDigit > 9 Then iDigit = iDigit - 9
        End If
        iSum = iSum + iDigit
        IsEven = Not IsEven
    Next
    IsValidAMKA = iSum Mod 10 = 0
End Function

Private Function searchString(ByVal strS As String, ByVal colname As
String) As Integer '-----function pou psaxnei to string sto
column tou DT
    Dim intcount As Integer = 0
    For Each Row As DataGridViewRow In DT.Rows
        If DT.Rows(intcount).Cells(colname).Value.ToString = strS Then
            DT.Rows(DT.Rows(intcount).Cells(0).Value).Selected = True
            'rowID = DT.Rows(intcount).Cells(0).Value
            Return DT.Rows(intcount).Cells(0).Value
        End If
        intcount += 1
    Next
    MsgBox("Η αναζήτηση δενεπέστρεψε αποτέλεσμα")
    Return rowID
End Function

```

```

Private Sub txtAMKASearch_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles txtAMKASearch.KeyDown
    If e.KeyCode = Keys.Return Then
        rowID = searchString(txtAMKASearch.Text, "AMKA").ToString
        Dim foundRow As DataRow = dsDT.Tables(0).Rows.Find(rowID)
        rowID = foundRow.Item(0).ToString
        cmbYparxonOnoma.Text = foundRow.Item("Όνοματεπώνυμο").ToString
        DT.FirstDisplayedScrollingRowIndex = DT.SelectedRows(0).Index
    End If
End Sub

Private Sub txtPhoneSearch_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles txtPhoneSearch.KeyDown
    If e.KeyCode = Keys.Return Then
        rowID = searchString(txtPhoneSearch.Text, "Τηλέφωνο").ToString
        Dim foundRow As DataRow = dsDT.Tables(0).Rows.Find(rowID)
        rowID = foundRow.Item(0).ToString
        cmbYparxonOnoma.Text = foundRow.Item("Όνοματεπώνυμο").ToString
        DT.FirstDisplayedScrollingRowIndex = DT.SelectedRows(0).Index
    End If
End Sub

Private Function getDatefromAMKA(ByVal strAMKA As String) As String
    Dim day As Integer = Cint(strAMKA.Substring(0, 2))
    Dim month As Integer = Cint(strAMKA.Substring(2, 2))
    Dim year As Integer = Cint(strAMKA.Substring(4, 2))
    If day <= 31 And month <= 12 Then
        Dim dateAMKA As String = day & "/" & month & "/19" & year
        Return dateAMKA
    ElseIf day <= 31 And month <= 12 And year <= 15 Then
        Dim dateAMKA As String = day & "/" & month & "/20" & year
        Return dateAMKA
    Else
        Return ""
    End If
End Function

Private Sub btnBck_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBck.Click
    FormBackUp.Show()
End Sub
End Class

```

FormStats

```

Imports System.Windows.Forms.DataVisualization.Charting

Public Class FormStats
    Private Sub FormStats_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        End Sub

    Private Sub ReturnResultsByAgeTraffic(ByVal query As String, ByVal
Pinakas As Array, ByVal strTitle As String)
        Dim dsStats As DataSet = New DataSet()
        Dim cmd As New OleDb.OleDbCommand
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Connection = Form1.con
    End Sub

```

```

cmd.CommandText = query
Dim da As OleDb.OleDbDataAdapter = New OleDb.OleDbDataAdapter(cmd)
da.Fill(dsStats)
Dim Yval(Pinakas.Length - 1) As Integer
For count As Integer = 0 To (Pinakas.Length - 1)
    Yval(count) = dsStats.Tables(0).Rows(count).Item(0)
Next
mscTrafficByAge.ChartAreas.Clear()
mscTrafficByAge.Series.Clear()
Dim area As New ChartArea("AREA")
mscTrafficByAge.ChartAreas.Add(area)
Dim series As Series = New Series(strTitle)
If cmbViewType.Text = "Πίνακας" Then
    series.ChartType = SeriesChartType.Pie
End If
If cmbViewType.Text = "Στήλες" Then
    series.ChartType = SeriesChartType.Column
End If
series.ChartArea = "AREA"
mscTrafficByAge.Series.Add(series)
For i As Integer = 0 To (Pinakas.Length - 1)
    series.Points.AddXY(Pinakas(i), Yval(i))
Next
End Sub

Private Sub cmbStats_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles cmbStats.SelectedIndexChanged
    Select Case cmbStats.SelectedIndex
        Case 0
            Dim PinStatLabelsAge() As String = {"<25", "25 με 35", "35
με 45", "45 με 55", "55 με 65", "65 με 75", ">75"}
            Dim strTitle1 As String = "Συχνότητα προσέλευσης ανα
ηλικιακή ομάδα"
            ReturnResultsByAgeTraffic("queryStatsAge1",
PinStatLabelsAge, strTitle1)
        Case 1
            Dim PinStatLabelsFamilyhis() As String = {"καρκίνος μαστού
", "άλλων καρκίνων"}
            Dim strTitle2 As String = "Συχνότητα εμφάνισης
οικογενειακού ιστορικού καρκίνου του μαστού και άλλων καρκίνων"
            ReturnResultsByAgeTraffic("queryStatsFamily",
PinStatLabelsFamilyhis, strTitle2)
        Case 2
            Dim PinStatLabelsNosos() As String = {"καλοήθη", "κακοήθη"}
            Dim strTitle3 As String = "Αναλογία ευρημάτων
κακοήθη/καλοήθη "
            ReturnResultsByAgeTraffic("queryStatsNosos",
PinStatLabelsNosos, strTitle3)
    End Select
End Sub
End Class

```

FormBackUp

```
Public Class FormBackUp
```

```

Private Sub btnBackUp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBackUp.Click
    Dim FileToCopy As String
    Dim NewCopy As String
    FileToCopy = System.AppDomain.CurrentDomain.BaseDirectory() &
"BCP.mdb"
    If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then
        NewCopy = FolderBrowserDialog1.SelectedPath & "\" &
String.Format("{0:dd_MM_yyyy}", DateTime.Now) & "\BCP.mdb"
        If Not
System.IO.Directory.Exists(FolderBrowserDialog1.SelectedPath & "\" &
String.Format("{0:dd_MM_yyyy}", DateTime.Now)) Then
System.IO.Directory.CreateDirectory(FolderBrowserDialog1.SelectedPath & "\"
& String.Format("{0:dd_MM_yyyy}", DateTime.Now))
            End If
            If System.IO.File.Exists(FileToCopy) = True Then
                System.IO.File.Copy(FileToCopy, NewCopy)
                MsgBox("Το αρχείο αντιγράφηκε")
            End If
        End If
    End Sub

Private Sub btnRestore_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRestore.Click
    Dim FileToCopy As String
    Dim NewCopy As String
    If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then
        FileToCopy = FolderBrowserDialog1.SelectedPath & "\BCP.mdb"
        NewCopy = System.AppDomain.CurrentDomain.BaseDirectory() &
"BCP.mdb"
        If System.IO.File.Exists(FileToCopy) = True Then
            System.IO.File.Copy(FileToCopy, NewCopy, True)
            MsgBox("Η επαναφορά ολοκληρώθηκε")
        End If
    End If
End Sub
End Class

```

PrintForm

```

Public Class PrintForm

    Private Sub PrintForm_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        Dim rowCount As Integer
        CrystalReportViewer1.Zoom(85)
        PrintReport1.Subreports.Item(2).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & ""
        PrintReport1.Subreports.Item(1).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & ""
        PrintReport1.Subreports.Item(0).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & ""
        PrintReport1.Subreports.Item(3).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & ""
        PrintReport1.Subreports.Item(4).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & ""
        PrintReport1.Subreports.Item(5).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & ""
        Dim test = PrintReport1.Subreports.Item(0).Name.ToString
        test = PrintReport1.Subreports.Item(1).Name.ToString
    End Sub
End Class

```



```

test = PrintReport1.Subreports.Item(2).Name.ToString
test = PrintReport1.Subreports.Item(3).Name.ToString
test = PrintReport1.Subreports.Item(4).Name.ToString
test = PrintReport1.Subreports.Item(5).Name.ToString
reportLoad("DiseasesData", 2)
reportLoad("DangersData", 1)
reportLoad("ClinicalData", 0)
reportLoad("DiagnosisData", 3)
reportLoad("StadioData", 5)
reportLoad("ObserveData", 4)
Form1.FillDGVs(Form1.dgvClinical, "Query1", Form1.rowID)
Form1.FillDGVs(Form1.dgvDiagnosis, "Query2", Form1.rowID)
Form1.FillDGVs(Form1.dgvObserve, "Query3", Form1.rowID)
rowCount = Form1.dgvClinical.RowCount
If rowCount = 0 Then
    cmbClinical.Text = "Κενή"
Else
    cmbClinical.Items.Add("Καμία")
    For i = 0 To rowCount - 1
        cmbClinical.Items.Add(getDGVDates(Form1.dgvClinical, i))
    Next
End If
rowCount = Form1.dgvDiagnosis.RowCount
If rowCount = 0 Then
    cmbDiagnosis.Text = "Κενή"
Else
    cmbDiagnosis.Items.Add("Καμία")
    For i = 0 To rowCount - 1
        cmbDiagnosis.Items.Add(getDGVDates(Form1.dgvDiagnosis, i))
    Next
End If
rowCount = Form1.dgvObserve.RowCount
If rowCount = 0 Then
    cmbObserve.Text = "Κενή"
Else
    cmbObserve.Items.Add("Καμία")
    For i = 0 To rowCount - 1
        cmbObserve.Items.Add(getDGVDates(Form1.dgvObserve, i))
    Next
End If
End Sub

Private Function getDGVDates(ByVal DGV As DataGridView, ByVal rowCount
As Integer) As String
    getDGVDates = DGV.Rows(rowCount).Cells(0).Value.ToString
    getDGVDates = Microsoft.VisualBasic.Left(getDGVDates, 9)
End Function

Private Sub chkOles_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles chkOles.CheckedChanged
    'Dim test = PrintReport1.Subreports.Item(0).Database
    If chkOles.Checked = True Then
        chkEpilogiKartelon.Checked = False
        chkDiseases.Enabled = False
        chkDiseases.Checked = True
        chkDangers.Enabled = False
        chkDangers.Checked = True
        lblClinical.Enabled = False
        cmbClinical.Enabled = False
        cmbClinical.Text = "Ολες"
        lblDiagnosis.Enabled = False

```

```

        cmbDiagnosis.Enabled = False
        cmbDiagnosis.Text = "Όλες"
        chkStadio.Enabled = False
        chkStadio.Checked = True
        lblObserve.Enabled = False
        cmbObserve.Enabled = False
        cmbObserve.Text = "Όλες"
    End If
End Sub

Private Sub chkEpilogiKartelon_CheckedChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles chkEpilogiKartelon.CheckedChanged
    If chkEpilogiKartelon.Checked = True Then
        chkOles.Checked = False
        chkDeseases.Enabled = True
        chkDeseases.Checked = False
        chkDangers.Enabled = True
        chkDangers.Checked = False
        lblClinical.Enabled = True
        cmbClinical.Enabled = True
        cmbClinical.Text = "Καμία"
        lblDiagnosis.Enabled = True
        cmbDiagnosis.Enabled = True
        cmbDiagnosis.Text = "Καμία"
        chkStadio.Enabled = True
        chkStadio.Checked = False
        lblObserve.Enabled = True
        cmbObserve.Enabled = True
        cmbObserve.Text = "Καμία"
    End If
End Sub

Private Sub chkDeseases_CheckedChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles chkDeseases.CheckedChanged
    If chkDeseases.Checked = False Then
        PrintReport1.Section3.SectionFormat.EnableSuppress = True
        CrystalReportViewer1.RefreshReport()
    Else
        PrintReport1.Section3.SectionFormat.EnableSuppress = False
        CrystalReportViewer1.RefreshReport()
    End If
End Sub

Private Sub chkDangers_CheckedChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles chkDangers.CheckedChanged
    If chkDangers.Checked = False Then
        PrintReport1.DetailSection2.SectionFormat.EnableSuppress = True
        CrystalReportViewer1.RefreshReport()
    Else
        PrintReport1.DetailSection2.SectionFormat.EnableSuppress =
False
        CrystalReportViewer1.RefreshReport()
    End If
End Sub

Private Sub cmbClinical_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles cmbClinical.SelectedIndexChanged
    If cmbClinical.Text = "Καμία" Then
        PrintReport1.DetailSection3.SectionFormat.EnableSuppress = True
        CrystalReportViewer1.RefreshReport()
    Else

```

```

        If PrintReport1.DetailSection3.SectionFormat.EnableSuppress =
True Then
            PrintReport1.DetailSection3.SectionFormat.EnableSuppress =
False
        End If
        Dim ClinicalID =
Form1.dgvClinical.Rows (cmbClinical.SelectedIndex -
1).Cells(1).Value.ToString()
        PrintReport1.Subreports.Item(0).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & " AND {ClinicalData.ClinicalID}=" &
ClinicalID & ""
        CrystalReportViewer1.RefreshReport()
    End If
End Sub

Private Sub cmbDiagnosis_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles cmbDiagnosis.SelectedIndexChanged
    If cmbDiagnosis.Text = "Καμία" Then
        PrintReport1.DetailSection4.SectionFormat.EnableSuppress = True
        CrystalReportViewer1.RefreshReport()
    Else
        If PrintReport1.DetailSection4.SectionFormat.EnableSuppress =
True Then
            PrintReport1.DetailSection4.SectionFormat.EnableSuppress =
False
        End If
        Dim DiagnosisID =
Form1.dgvDiagnosis.Rows (cmbDiagnosis.SelectedIndex -
1).Cells(1).Value.ToString()
        PrintReport1.Subreports.Item(3).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & " AND {DiagnosisData.DiagnosisID}=" &
DiagnosisID & ""
        CrystalReportViewer1.RefreshReport()
    End If
End Sub

Private Sub chkStadio_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles chkStadio.CheckedChanged
    If chkStadio.Checked = False Then
        PrintReport1.DetailSection5.SectionFormat.EnableSuppress = True
        CrystalReportViewer1.RefreshReport()
    Else
        PrintReport1.DetailSection5.SectionFormat.EnableSuppress =
False
        CrystalReportViewer1.RefreshReport()
    End If
End Sub

Private Sub cmbObserve_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles cmbObserve.SelectedIndexChanged
    If cmbObserve.Text = "Καμία" Then
        PrintReport1.DetailSection6.SectionFormat.EnableSuppress = True
        CrystalReportViewer1.RefreshReport()
    Else
        If PrintReport1.DetailSection6.SectionFormat.EnableSuppress =
True Then
            PrintReport1.DetailSection6.SectionFormat.EnableSuppress =
False
        End If
        Dim ObserveID = Form1.dgvObserve.Rows (cmbObserve.SelectedIndex
- 1).Cells(1).Value.ToString()

```

```

        PrintReport1.Subreports.Item(4).RecordSelectionFormula =
"{BasicData.id}=" & Form1.rowID & " AND {ObserveData.observeID}=" &
ObserveID & ""
        CrystalReportViewer1.RefreshReport()
    End If
End Sub

Private Sub reportLoad(ByVal pinakas As String, ByVal x As Integer)
    Dim dsReport As New BCPDataSet
    Dim Sql = "SELECT * FROM BasicData"
    Dim daBasicData = New OleDb.OleDbDataAdapter(Sql, Form1.con)
    daBasicData.MissingSchemaAction = MissingSchemaAction.AddWithKey
    daBasicData.Fill(dsReport, "BasicData")
    Sql = "SELECT * FROM " & pinakas & ""
    Dim daKartela = New OleDb.OleDbDataAdapter(Sql, Form1.con)
    daKartela.MissingSchemaAction = MissingSchemaAction.AddWithKey
    daKartela.Fill(dsReport, pinakas)
    PrintReport1.Subreports.Item(x).SetDataSource(dsReport)
End Sub

```

Script Inno Setup

```

#define MyAppName "BCP"
#define MyAppVersion "1.5"
#define MyAppPublisher "My Company, Inc."
#define MyAppExeName "BreastCancerProject.exe"

```

[Setup]

```

AppId={{D7B757CA-5C03-408F-B9C3-18D09D76ED76}}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
;AppVerName={#MyAppName} {#MyAppVersion}
AppPublisher={#MyAppPublisher}
DefaultDirName={pf}\{#MyAppName}
DisableDirPage=yes
DefaultGroupName={#MyAppName}
OutputDir=C:\Documents and Settings\vostrix\Desktop\ino
OutputBaseFilename=Setup
SetupIconFile=C:\Documents and Settings\vostrix\My Documents\Visual Studio
2008\Projects\BreastCancerProject\bin\Debug\mediiicon.ico
Compression=lzma
SolidCompression=yes

```

[Languages]

```

Name: "english"; MessagesFile: "compiler:Default.isl"
Name: "greek"; MessagesFile: "compiler:Languages\Greek.isl"

```

[Tasks]

```

Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDescription: "{cm:AdditionalIcons}";
Flags: unchecked

```

[Files]

```

Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio
2008\Projects\BreastCancerProject\bin\Debug\BreastCancerProject.exe"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio
2008\Projects\BreastCancerProject\bin\Debug\BCP.mdb"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio
2008\Projects\BreastCancerProject\bin\Debug\brush.ico"; DestDir: "{app}"; Flags: ignoreversion

```

Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\Helpchm.chm"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\mediicon.ico"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\statue.gif"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Windows.Forms.DataVisualization.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Core.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Data.DataSetExtensions.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Data.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Drawing.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Windows.Forms.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Xml.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\System.Xml.Linq.dll"; DestDir: "{app}"; Flags: ignoreversion
Source: "C:\Documents and Settings\vostrix\My Documents\Visual Studio 2008\Projects\BreastCancerProject\bin\Debug\CRRedist2008_x86.msi"; DestDir: "{tmp}"; Flags: deleteafterinstall

[Icons]

Name: "{group}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"
Name: "{commondesktop}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"; Tasks: desktopicon;
IconFilename: "{app}\mediicon.ico" ;

[Code]

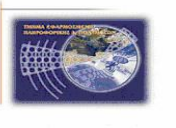
```
var
IsRestartNeeded: boolean;
function Is86(): boolean;
begin
Result:= not IsWin64;
end;
function ShouldInstallCR86(): boolean;
begin
result:=false;
if ((is86) and (not IsRestartNeeded)) then
result:=true;
end;
```

[Run]

Filename: "msiexec.exe"; Parameters: "/i ""{tmp}\CRRedist2008_x86.msi"""; Check: ShouldInstallCR86;
Filename: "{app}\{#MyAppExeName}"; Description: "{cm:LaunchProgram,{#StringChange(MyAppName, '&', '&&')}}"; Flags: nowait postinstall skipifsilent

Παράρτημα Β – Παρουσίαση

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Κατασκευή εφαρμογής για καταχώρηση και παρακολούθηση ασθενών του ογκολογικού τμήματος καρκίνου του μαστού.

Μπούκλας Ιωάννης (ΑΜ: 225)

Στυλιανού Σταύρος (ΑΜ: 301)

Επιβλέπων καθηγητής : Παπαδουράκης Γεώργιος

Έντυπη μορφή δεδομένων

ΚΑΙΝΙΚΗ ΕΞΕΤΑΣΗ				ΙΑΤΡΕΙΟ ΜΑΣΤΟΥ				
Κλινικά ευρήματα: <i>Ναι Όχι</i>				ΕΝΗΜΕΡΩΤΙΚΟ ΕΝΤΥΠΟ ΑΣΘΕΝΟΥΣ				
Μαστός				ΣΤΟΙΧΕΙΑ ΑΣΘΕΝΟΥΣ				
Κυστικό στήθος	Δεξιά	Αριστερά	Μέγεθος(cm):.....	Όνοματεπώνυμο:.....	Ηλικία:.....	Φύλο:.....		
Συμπαγές στήθος			Μέγεθος(cm):.....	Διεύθυνση:.....				
Διάχυτα οζίδια			Μέγεθος(cm):.....	Τηλέφωνο:.....				
Διάχυτη ενσυσθραία				ΠΑΡΟΥΣΑ ΝΟΣΟΣ				
Τοπική ενσυσθραία				Ασυμπτωματική				
Οξείη				Συμπτωματική				
Επισποική δερματός								
Διήθηση δέρματος								
Ρετινίτ'orange								
Σημειώνεται στη περιτονία								
Σημειώνεται στο θωρακικό τρίγωνο								
				Διάρκεια σε εβδομάδες				
				Μικρή ψηλαφητή μάζα	Δεξιά	Αριστερά	Δεξιά	Αριστερά
				Όγκος				
				Έκκριμα θηλής				
				• Αμικτρού				
				• Μη συμμετρικό				
				• Δεν αναφέρεται				
				Επισποική θηλής				
				Άλλη ανωμαλία της θηλής				
				Δερματική παραμόρφωση				
				Πόνος (μη κυκλικός)				
				Πόνος (κυκλικός)				
				Μισοκύκλιες ψηλαφητές μάζες				
				Άλλα				
				ΑΤΟΜΙΚΟ ΑΝΑΜΝΗΣΤΙΚΟ				
				Μαστός: Καλοήγη νόσος	<i>Ναι Όχι</i>			
				• <i>Ναι</i>	Δεξιά	Αριστερά	Τύπος:	
				Κακοήγη νόσος	<i>Ναι Όχι</i>			
				• <i>Ναι</i>	Δεξιά	Αριστερά	Τύπος:	
				Άλλες κακοήθειες:	<i>Ναι Όχι</i>			
				• <i>Ναι</i>	Μήτρα	Ωοθήκη	Παχό έντερο	Άλλος τύπος:
				Άλλο παθολογικό/χειρουργικό ιστορικό:	<i>Ναι Όχι</i>			
				• <i>Ναι</i>	Ενδοκρινείς ΚΝΣ	Αναπνευστικό	Οστά	Γαστροεντερικό
				Ουρογεννητικό	Άλλο:			
				Διηθητηριακός:	<i>Ναι Όχι</i>			
				Είδος:				
				• <i>Ναι Όχι</i>	Είδος:			
				Μετάγγιση:	<i>Ναι Όχι</i>	Είδος:		

Σκοπός και Στόχοι Εργασίας

- Desktop εφαρμογή
- Καλύτερη δυνατή αντιστοιχία καρτέλας-ηλεκτρονικής φόρμας.
- Παροχή extra λειτουργιών
- Να βοηθήσει να καλυφθούν κενά στη μηχανογράφηση
- Άμεσα διαθέσιμη στο εμπόριο

Εργαλεία υλοποίησης εφαρμογής

- **Visual Studio**
 - › .Net
 - › Υλοποίηση διεπαφής
 - › Προβολή στατιστικών με MS Charts
 - › Αναφορές με Crystal Reports
- **Microsoft ACCESS.**
 - › Υλοποίηση Βάσης δεδομένων
 - › Υλοποίηση ερωτημάτων SQL
- **Επιπλέον εργαλεία**
 - › Υλοποίηση αρχείου Help με Microsoft Help Workshop
 - › Δημιουργία εικονιδίων με Adobe Photoshop
 - › Υλοποίηση αρχείου εγκατάστασης με Inno Setup

Υλοποίηση Εφαρμογής

•Σχεδιασμός Διεπαφής

- Χρηστικότητα
- Χρώμα
- Μηνύματα

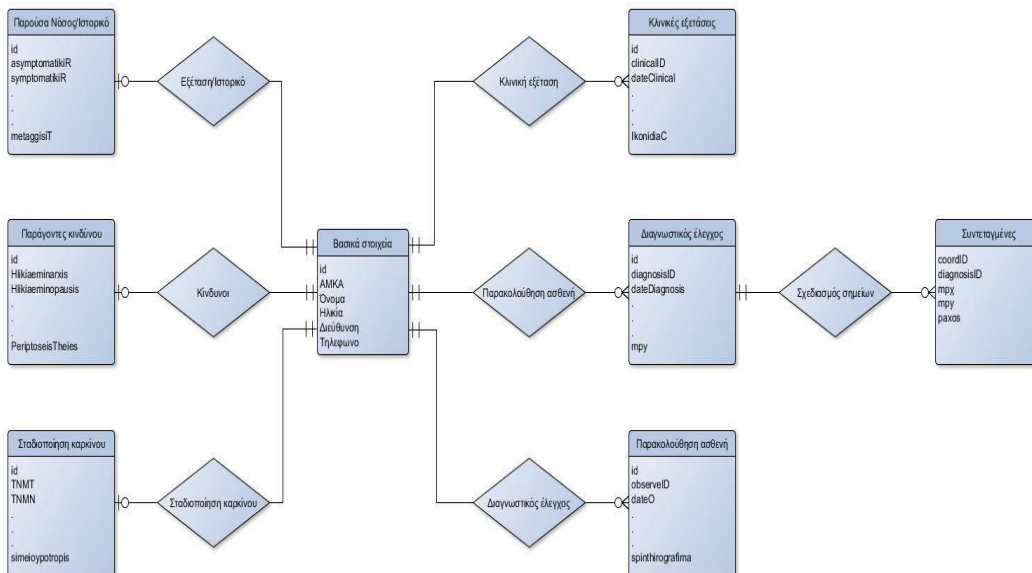
•Σχεδιασμός βάσης δεδομένων

- Πίνακες
- Συσχετίσεις πινάκων
- Ερωτήματα

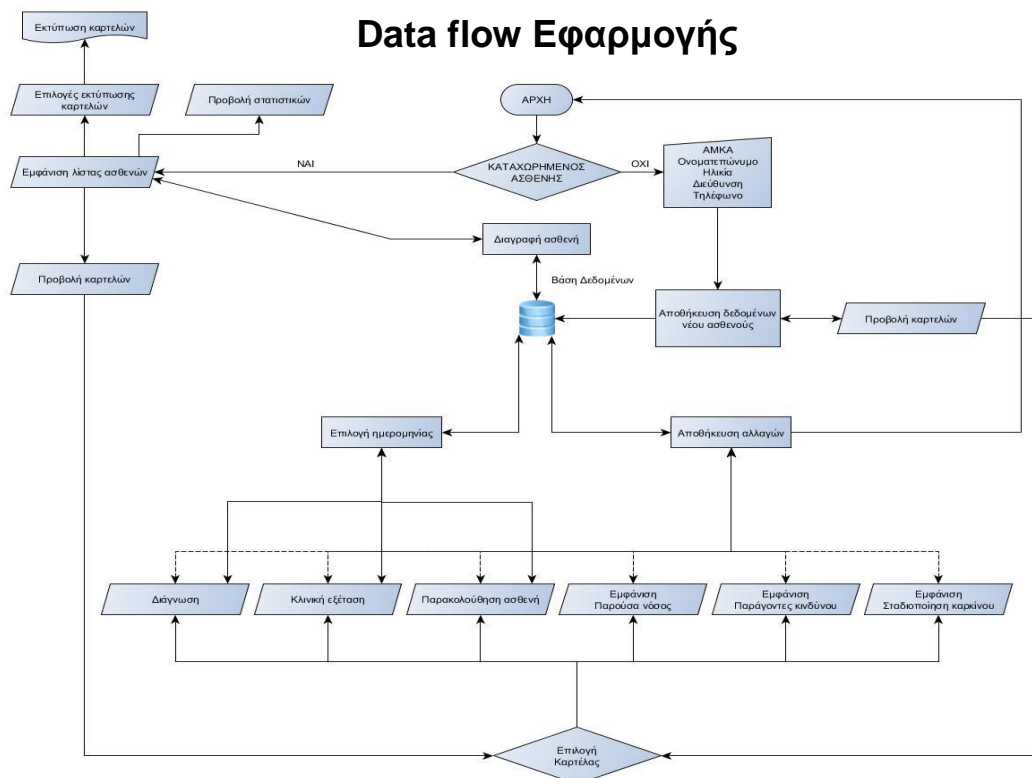
Setup

- Εγκατάσταση με οδηγό
- Έλεγχος συμβατότητας
- Βοηθητικά αρχεία περιέχονται στην διανομή

Βάση Δεδομένων



Data flow Εφαρμογής



Ανάλυση του προγράμματος

- Εισαγωγή νέου ασθενούς
- Εύρεση υπάρχοντος ασθενούς
 - Διαγραφή
 - Προβολή και επεξεργασία
 - Αποθήκευση αλλαγών
 - Καταχώρηση εξέτασης
 - Διαγραφή εξέτασης
 - Προεπισκόπηση εκτύπωσης
- Προβολή στατιστικών
- Προβολή αρχείου βοήθειας
- Back up

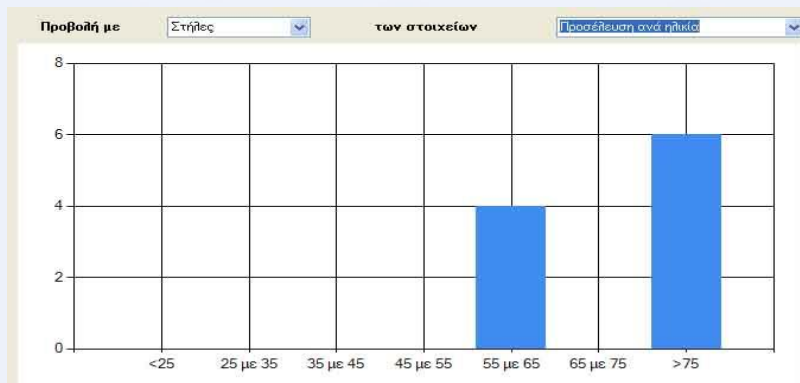
Εκτύπωση

- Προβολή αναφορών μέσω Crystal Reports
- Επιλογή δεδομένων εκτύπωσης
- Προεπισκόπηση

The screenshot shows a web-based clinical examination form. At the top, there are navigation and search options. The main form area is titled 'ΚΛΙΝΙΚΗ ΕΞΕΤΑΣΗ' (Clinical Examination). It contains several sections for recording patient data and examination findings. The 'Μαστός' (Breast) section includes checkboxes for 'Μέγεθος (cm)' (Size) and 'Εισαγωγή θηλής' (Nipple insertion) for both 'Δεξιό' (Right) and 'Αριστερό' (Left) breasts. The 'Ουλή' (Axilla) section includes checkboxes for 'Εισαγωγή θηλής' (Nipple insertion) and 'Ανωμαλίες θηλάδας' (Abnormalities of the areola). The 'Προηγούμενες επιμέτρησεις' (Previous measurements) section includes checkboxes for 'Θωρακικό' (Chest) and 'Αριστερό' (Left). The form is currently in a preview or editing state, with various input fields and checkboxes visible.

Στατιστικά

- Επιλογή δεδομένων
- Επιλογή μορφής εμφάνισης



Back Up

- Δημιουργία back up της βάσης δεδομένων
- Επαναφορά της βάσης από το αντίγραφο ασφαλείας



Αρχείο βοήθειας

- Χρήση αρχείων html.
- Προβολή απλών και κατανοητών οδηγιών
- Προβολή κατατοπιστικών εικόνων

Συμπεράσματα

- Χαμηλές απαιτήσεις συστήματος
- Εύκολη εγκατάσταση \ απεγκατάσταση
- Γνώριμο περιβάλλον λειτουργίας
- Ταχύτερη προσπέλαση δεδομένων
- Αξιοποίηση των δεδομένων

ΕΠΕΚΤΑΣΕΙΣ

- Καταχώρηση με scanner
- Πολυγλωσσικό
- Προσαρμογή αναλύσεων
- Λειτουργία on line
- Λειτουργία προσθήκης SQL query
- 3D απεικόνιση

Παράρτημα Γ – Περίληψη πτυχιακής σε στυλ δημοσίευσης

Σχολή Τεχνολογικών Εφαρμογών Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή Εργασία

Κατασκευή εφαρμογής για καταχώρηση και παρακολούθηση ασθενών του ογκολογικού τμήματος καρκίνου του μαστού.

Μπούκλας Ιωάννης (AM: 225)

Στυλιανού Σταύρος (AM: 301)

1. Εισαγωγή

Η πτυχιακή αυτή ασχολείται με όλες τις παραμέτρους σχεδιασμού και υλοποίησης μιας εφαρμογής η οποία καταγράφει και αρχειοθετεί τα συμπεράσματα και ευρήματα εξετάσεων ενός ιατρείου καρκίνου του μαστού. Τα δεδομένα αποθηκεύονται και ανακαλούνται από μια βάση δεδομένων μέσω της εφαρμογής, παρέχοντας έτσι όλα τα πλεονεκτήματα της ψηφιακής διαχείρισης δεδομένων αντικαθιστώντας την χρήση χάρτου. Κατά την υλοποίησης της εφαρμογής, είχαμε την ευκαιρία να εντρυφήσουμε στον προγραμματισμό στην πιο ουσιαστική του μορφή που είναι η παραγωγή μιας ολοκληρωμένης και λειτουργικής εφαρμογής, καθώς και σε πολλά εργαλεία και αρχές που περιβάλλουν την ανάπτυξη μιας τέτοιας εφαρμογής.

2. Στόχοι Εργασίας

Ο καρκίνος του μαστού αποτελεί μία από τις συχνότερα εμφανιζόμενες μορφές καρκίνου παγκοσμίως και είναι η πρώτη σε αριθμό κρουσμάτων στο γυναικείο πληθυσμό. Σύμφωνα με τη Διεθνή Έκθεση για τις καρκινικές νόσους που παρουσιάστηκε στη Γενεύη της Ελβετίας τον Απρίλιο του 2003 ο καρκίνος του μαστού αποτελεί την πιο συχνή μορφή καρκίνου μεταξύ των γυναικών, με περίπου 1.000.000 νέα κρούσματα παγκοσμίως.

Η ακρίβεια που απαιτείται για την υλοποίηση ενός συστήματος διαχείρισης αυτών των δεδομένων καθώς και η προσοχή για την αποφυγή λαθών, αποτρέπει την ενασχόληση με την δημιουργία προγραμμάτων ιατρικής παρακολούθησης. Η ανάπτυξη της εφαρμογής βασίστηκε σε έξι καρτέλες που χρησιμοποιούν ιατροί και ασθενείς για την συμπλήρωση των στοιχείων των ασθενών και των εξετάσεων. Στόχος της εφαρμογής είναι αρχικά η ηλεκτρονική αρχειοθέτηση των ασθενών και των εξετάσεων τους και ίσως η οριστική αντικατάσταση των καρτελών. Η εφαρμογή καθίσταται συμφέρουσα προς το περιβάλλον και το ογκολογικό τμήμα μαστού καθώς συντελεί στην εξοικονόμηση αποθηκευτικού χώρου και επειδή καθιστά την κατανάλωση αποθεμάτων χάρτου μη αναγκαία. Η ανάκτηση των δεδομένων γίνεται κατά πολύ ευκολότερη, παρέχοντας παράλληλα υπηρεσίες που η έντυπη αρχειοθέτηση καθιστά σχεδόν αδύνατες όπως είναι η εξαγωγή στατιστικών και η σύγκριση κλινικών ευρημάτων. Παράλληλα διευκολύνεται η φορητότητα των δεδομένων μέσω αποθηκευτικών μονάδων.

3. Ανάπτυξη της εφαρμογής

Κατά την ανάπτυξη της εφαρμογής χρησιμοποιήσαμε διάφορα προγράμματα για την υλοποίηση του κάθε τμήματος της. Το IDE

πάνω στο οποίο στηρίξαμε την ανάπτυξη της εφαρμογής είναι το **Visual Studio** το οποίο υποστηρίζει την ανάπτυξη προγραμμάτων βοηθώντας τον προγραμματιστή να αναπτύξει τα προγράμματά του με σχετική ευκολία, καθώς η τεχνολογία Microsoft IntelliSense οδηγεί τον προγραμματιστή να κατανοήσει με ευκολία πιθανά λάθη του κώδικά του, υπογραμμίζοντας τα με κόκκινη γραμμή δυναμικά κατά το χρόνο συγγραφής του προγράμματος. Περιλαμβάνει «στοιχεία ελέγχου» πάνω στα οποία βασίζεται το «χτίσιμο» ενός project και υποστηρίζει διαφορετικές γλώσσες προγραμματισμού καθώς και τη μεταφορά προγραμμάτων από την μία γλώσσα στην άλλη. Η γραμμή εργαλείων περιλαμβάνει τα βασικά στοιχεία ελέγχου. ωστόσο, ο χρήστης μπορεί εύκολα να προσθέσει επιπλέον όσα προαιρετικά εργαλεία. Τέτοια στοιχεία ελέγχου είναι τα **MS charts** και **Crystal viewer**.

Το **MS chart control** επιτρέπει τη δημιουργία διαγραμμάτων για Windows Forms εφαρμογές και το χρησιμοποιήσαμε για την λειτουργία προβολής στατιστικών της εφαρμογής μας. Μια εικόνα chart αποτελείται από διάφορα στοιχεία, όπως άξονες, σειρές και άλλα στοιχεία γραφήματος και παρέχεται η δυνατότητα πλήρους προσαρμογής τους με χρήση των ιδιοτήτων εμφάνισης τους.

Τα **Crystal Reports** είναι πλέον το καθιερωμένο πρότυπο για την υποβολή εκθέσεων στο Visual Studio.NET και τα χρησιμοποιήσαμε για την υλοποίηση της λειτουργίας της εκτύπωσης. Με τα Crystal Reports γίνεται εφικτή η δημιουργία διαδραστικών περιεχόμενων σε ποιότητα παρουσίασης για πλατφόρμα .NET. Παρέχεται το περιβάλλον εργασίας Designer Crystal Report για την απλούστευση της δημιουργίας και μορφοποίησης της αναφοράς με βάση τις αναπτυξιακές μας ανάγκες.

Για ορισμένα από τα γραφικά στοιχεία της εφαρμογής όπως τα εικονίδια χρησιμοποιήσαμε το **Adobe Photoshop** το οποίο είναι το ιδανικό πρόγραμμα για δημιουργία και επεξεργασία εικόνων. Το αρχείο βοήθειας που εμπεριέχεται στην εφαρμογή δημιουργήθηκε με το εργαλείο **Microsoft Help Workshop** που είναι το πρότυπο σύστημα βοήθειας για την πλατφόρμα των Windows. Μιας και δημιουργεί ένα σύστημα παροχής πληροφοριών μπορεί να χρησιμοποιηθεί για μια ευρεία γκάμα εφαρμογών συμπεριλαμβανομένων οδηγών

εκμάθησης, διαδραστικών βιβλίων και κυρίως για δημιουργία βοήθειας χρήσης εφαρμογών. Επίσης καθιστά δυνατή τη συμπίεση HTML, γραφικών και άλλων αρχείων σε ένα σχετικά μικρό σε όγκο αρχείο βοήθειας του οποίου η κατάληξη είναι .chm. Για την υλοποίηση του αρχείου εγκατάστασης, χρησιμοποιήσαμε το πρόγραμμα **Inno Setup** προκειμένου να δημιουργήσουμε ένα πρόγραμμα εγκατάστασης της εφαρμογής και όλων των εξαρτήσεων της που να αποτελείται από ένα και μόνο εκτελέσιμο αρχείο. Προσφέρεται η δυνατότητα για disk spanning, κατά την οποία δημιουργούνται προγράμματα εγκατάστασης από περισσότερα αρχεία συγκεκριμένου μεγέθους για την τοποθέτηση σε αντίστοιχα μέσα

Η ανάπτυξη της βάσης δεδομένων μας έγινε μέσω της **Microsoft Access** και η διασύνδεση της με την εφαρμογή, με τεχνολογία **Ole db**. Η Microsoft Access είναι ένα ευέλικτο και εύκολο στη χρήση σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) καθώς και ένα πλήρες εργαλείο ανάπτυξης εφαρμογών για βάσεις δεδομένων. Ανάμεσα στα ισχυρότερα χαρακτηριστικά της Access είναι και οι Οδηγοί (Wizards), τους οποίους μπορούμε να χρησιμοποιούμε για την κατασκευή πινάκων και ερωτημάτων και για τον ορισμό μιας μεγάλης ποικιλίας φορμών και αναφορών. Οι Οδηγοί αυτοί είναι έτοιμες σχεδιάσεις που έχει ενσωματωμένες η Access για να μας βοηθήσει στη σχεδίαση της βάσης δεδομένων μέσω κατανοητών βημάτων.

Το **OLE DB** (Object Linking and Embedding Database) είναι ένα API (application programming interface) σχεδιασμένο από την Microsoft που επιτρέπει την πρόσβαση δεδομένων από ποικίλες πηγές με έναν ενοποιημένο τρόπο. Το OLE DB διαχωρίζει την αποθήκευση των υπό προσπέλαση δεδομένων από την εφαρμογή, μέσω εννοιών όπως είναι η πηγή των δεδομένων, οι εντολές και οι εγγραφές. Το OLE DB μπορεί να παίζει τον ρόλο του παρόχου δεδομένων στις εφαρμογές που ζητούν δεδομένα για την λειτουργία τους ή εκείνου που ζητάει δεδομένα από εφαρμογές με τις οποίες συνδέεται καθώς επίσης υποστηρίζει απευθείας σύνδεση με SQL βάσεις δεδομένων.

4. Σχεδιασμός διεπαφής

Για την δημιουργία της διεπαφής λάβαμε υπόψη μας τις αρχές του σχεδιασμού

διεπαφής που αποτελούν τη βάση για τη σχεδίαση κάθε διεπαφής παρότι δεν εφαρμόζονται όλες τους σε κάθε σχεδίαση.

- **Εξοικείωση χρηστών.** Η διασύνδεση πρέπει να χρησιμοποιεί όρους και έννοιες που προέρχονται από τις εμπειρίες των ανθρώπων που θα χρησιμοποιούν περισσότερο το σύστημα και όχι ορολογία υπολογιστών.
- **Συνέπεια.** Η διασύνδεση πρέπει να είναι συνεπής, δηλαδή, όπου είναι δυνατό, παρόμοιες λειτουργίες θα πρέπει να ενεργοποιούνται με τον ίδιο τρόπο.
- **Ελάχιστες εκπλήξεις.** Οι χρήστες δεν πρέπει να εκπλήσσονται ποτέ από τη συμπεριφορά του συστήματος. Αν μια διαταγή λειτουργεί με ένα συγκεκριμένο τρόπο, ο χρήστης πρέπει να είναι σε θέση να προβλέπει τη λειτουργία παρόμοιων διαταγών.
- **Ανακαμψιμότητα.** Η διασύνδεση πρέπει να περιλαμβάνει μηχανισμούς που θα επιτρέπουν στους χρήστες να ανακάμπτουν από σφάλματα. Το σύστημα πρέπει να είναι ανθεκτικό σε σφάλματα των χρηστών και να τους επιτρέπει να ανακάμπτουν από τα λάθη τους.
- **Καθοδήγηση χρηστών.** Η διασύνδεση πρέπει να παρέχει κατανοητές αποκρίσεις όταν συμβαίνουν λάθη, και να παρέχει στο χρήστη συναφή βοήθεια.
- **Διαφορετικότητα χρηστών.** Η διασύνδεση πρέπει να παρέχει κατάλληλες λειτουργίες αλληλεπίδρασης για τους διάφορους τύπους χρηστών του συστήματος.
- **Χρώμα.** Το χρώμα χρησιμοποιείται στην παρουσίαση για την επίτευξη τριών βασικών στόχων όπως προσδιορισμός, αντίθεση και επισήμανση και μπορεί να βοηθήσει το χρήστη να αντιλαμβάνεται πολύπλοκες δομές πληροφοριών
- **Μηνύματα.** Τα μηνύματα πρέπει να χαρακτηρίζονται από ευγένεια, ακρίβεια, συνέπεια και δημιουργική κριτική και πρέπει το γνωστικό υπόβαθρο και η πείρα του χρήστη να αποτελούν καθοριστικό παράγοντα στο σχεδιασμό των μηνυμάτων

5. Σχεδίαση Βάσεων δεδομένων

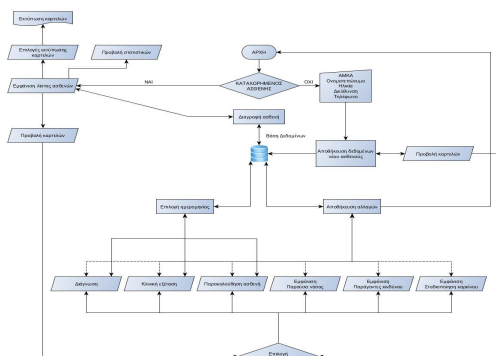
Η σωστή σχεδίαση είναι ουσιαστικής σημασίας για την επίτευξη των στόχων κατά την δημιουργία μιας βάση δεδομένων. Στόχος μας είναι μια βάση δεδομένων που να

καλύπτει τις ανάγκες μας και μπορεί να εξυπηρετήσει με ευκολία τις αλλαγές. Για αυτό τον σκοπό υπάρχουν ορισμένες αρχές που καθοδηγούν τη διαδικασία της σχεδίασης βάσης δεδομένων. Η πρώτη αρχή είναι ότι οι διπλότυπες πληροφορίες (πλεονάζοντα δεδομένα) είναι ανεπιθύμητες, διότι σπαταλούν χώρο και αυξάνουν την πιθανότητα σφαλμάτων και ασυνεπειών. Η δεύτερη αρχή είναι ότι η ορθότητα και η πληρότητα των πληροφοριών είναι σημαντικές. Εάν η βάση δεδομένων περιέχει λανθασμένες πληροφορίες, οι εκθέσεις που αντλούν πληροφορίες από τη βάση δεδομένων επίσης θα περιέχουν λανθασμένες πληροφορίες. Επομένως μια καλή σχεδίαση βάσης δεδομένων είναι αυτή που:

- Χωρίζει τις πληροφορίες σε πίνακες που βασίζονται σε θέματα, για να ελαττωθούν τα πλεονάζοντα δεδομένα.
- Παρέχει στην Access τις πληροφορίες που απαιτούνται για να συνδεθούν οι πληροφορίες στους πίνακες, σύμφωνα με τις ανάγκες.
- Βοηθά την υποστήριξη και εξασφαλίζει την ακρίβεια και την ακεραιότητα των πληροφοριών.
- Εξυπηρετεί την επεξεργασία των δεδομένων και των αναγκών έκθεσης.

6. Ανάλυση της εφαρμογής

Οι επιλογές για τις λειτουργίες της εφαρμογής βρίσκονται στο μενού επιλογών. Οι βασικότερες και περισσότερο χρησιμοποιούμενες από αυτές βρίσκονται και στην επιφάνεια εργασίας της εφαρμογής με την μορφή κουμπιών συντόμευσης. Τα κουμπιά αυτά είναι έξι, ένα για κάθε παράθυρο και είναι ορατά καθ' όλη τη διάρκεια χρήσης του προγράμματος προκειμένου ο χρήστης να είναι σε θέση να προβάλει με ευκολία τις διάφορες λειτουργίες της εφαρμογής. Οι λειτουργίες των κουμπιών και μενού είναι οι εισαγωγή νέου ασθενούς, υπάρχων ασθενής, εγχειρίδιο χρήσης, προβολή στατιστικών, εκτύπωση, back up και η έξοδος του προγράμματος. Η κύρια φόρμα του προγράμματος μας παρέχει την επιλογή εισαγωγής ενός νέου ασθενή στην βάση μας ή την πλοήγηση σε αποτελέσματα και συμπεράσματα που έχουν προκύψει από ασθενείς που έχουν ήδη καταχωρηθεί.



Εικόνα 1 . Data flow Εφαρμογής

- Επιλέγοντας να εισάγουμε νέο ασθενή πρέπει πρώτα να συμπληρώσουμε στοιχεία του όπως όνομα, διεύθυνση Α.Μ.Κ.Α κτλ. Αμέσως μετά ανοίγει ένα tab page με την πρώτη καρτέλα που αντιστοιχεί στην «Παρούσα νόσο/Ατομικό Αναμνηστικό». Μέσω της πλοήγησης με την μορφή tabs μπορούν να συμπληρωθούν και οι άλλες καρτέλες «Παράγοντες κινδύνου», «Κλινική εξέταση», «Διαγνωστικός έλεγχος», «Σταδιοποίηση καρκίνου», «Παρακολούθηση ασθενών με καρκίνο του μαστού». Μετά την συμπλήρωση όσων δεδομένων είναι διαθέσιμα και πατώντας το κουμπί «Αποθήκευση» τα δεδομένα αποθηκεύονται στην βάση δεδομένων μας.
- Επιλέγοντας «Υπάρχων ασθενής» μπορούμε να ανατρέξουμε στο ιστορικό προηγούμενων εξετάσεων, να διορθώσουμε ή και να συμπληρώσουμε νέα δεδομένα για έναν ασθενή που έχει ήδη καταχωρηθεί.
- Ανά πάσα στιγμή μπορούμε να εκτυπώσουμε όποια καρτέλα θέλουμε εφόσον αυτή έχει αποθηκευτεί στην βάση μας. Η επιλογή της εκτύπωσης, προβάλλει την φόρμα εκτύπωσης η οποία μέσω των Crystal Reports, μας δίνει την δυνατότητα να επιλέξουμε ποιες από τις καρτέλες θέλουμε να εκτυπώσουμε.
- Παρέχεται επίσης η επιλογή της εμφάνισης στατιστικών που μέσω του control MSCharts. Αν επιλέξουμε την εμφάνιση των στατιστικών εμφανίζεται η φόρμα στατιστικών που μας δίνει την επιλογή ορισμένων στατιστικών και της μορφής με την οποία θέλουμε να εμφανίζονται. Η άντλησή τους γίνεται μέσω SQL queries που είναι αποθηκευμένα στην βάση μας

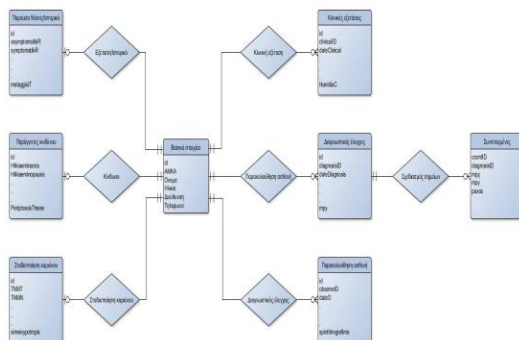
7. Βάση Δεδομένων

Κατά την ανάπτυξη της βάσης δεδομένων για κάθε καρτέλα με στοιχεία ασθενών και εξετάσεων δημιουργήθηκε ξεχωριστός πίνακας με στήλες τα δεδομένα των καρτελών. Επίσης δημιουργήθηκαν ορισμένα SQL ερωτήματα για ορισμένα αποτελέσματα που απαιτούνταν για συγκεκριμένες λειτουργίες του προγράμματος. Στην βάση μας χρησιμοποιούμε δύο ειδών ερωτημάτων (queries). Ερωτήματα επιλογής (Select queries) και ερωτήματα ενοποίησης (Union queries). Οι πίνακες που περιέχει και που αντιστοιχούν στις καρτέλες με τα στοιχεία των εξετάσεων είναι οι εξής.

- **BasicData** - Ο πίνακας που περιέχει τα βασικά στοιχεία του ασθενούς (ονοματεπώνυμο, τηλέφωνο, διεύθυνση, ηλικία). Είναι ο βασικός πίνακας για την συσχέτιση με τους υπόλοιπους πίνακες της βάσης. Έχει σαν primary key το **id** ασθενούς που είναι αύξων αριθμός, αυξάνει με κάθε καινούρια καταχώρηση.
- **DeseasesData** – Περιέχει τα στοιχεία της καρτέλας παρούσα νόσος/οικογενειακό ιστορικό. Έχει σαν πρωτεύων κλειδί το **id** ασθενούς που είναι αριθμός. Κάθε ασθενής έχει το πολύ μία καταχώρηση σε αυτόν τον πίνακα.
- **DangersData** – Περιέχει τα στοιχεία της καρτέλας παράγοντες κινδύνου. Έχει σαν πρωτεύων κλειδί το **id** ασθενούς που είναι αριθμός. Κάθε ασθενής έχει το πολύ μία καταχώρηση σε αυτόν τον πίνακα.
- **StadioData** – Περιέχει τα στοιχεία της καρτέλας σταδιοποίηση καρκίνου. Έχει σαν πρωτεύων κλειδί το **ID** ασθενούς που είναι αριθμός. Κάθε ασθενής έχει το πολύ μία καταχώρηση σε αυτόν τον πίνακα.
- **ClinicalData** – Περιέχει τα στοιχεία της καρτέλας κλινική εξέταση. Έχει σαν πρωτεύων κλειδί το **clinicalID** που είναι αύξων αριθμός, αυξάνει με κάθε καταχώρηση στον πίνακα. Περιέχει το **id** ασθενούς σαν ξένο κλειδί προκειμένου να συσχετιστεί με τον πίνακα BasicData. Κάθε ασθενής μπορεί να έχει παραπάνω από μία καταχωρήσεις σε αυτόν τον πίνακα.
- **DiagnosisData** – Περιέχει τα στοιχεία της καρτέλας διαγνωστικός έλεγχος. Έχει σαν πρωτεύων κλειδί το **diagnosisID** που είναι αύξων αριθμός, αυξάνει με κάθε

καταχώρηση στον πίνακα. Περιέχει το **id** ασθενούς σαν ξένο κλειδί προκειμένου να συσχετιστεί με τον πίνακα BasicData. Κάθε ασθενής μπορεί να έχει παραπάνω από μία καταχωρήσεις σε αυτόν τον πίνακα.

- **CoordinatesData** – Στην εφαρμογή ο γιατρός έχει την δυνατότητα να τοποθετήσει κουκίδες σε μια εικόνα στην καρτέλα του διαγνωστικού ελέγχου προκειμένου να έχει γραφική απεικόνιση των ευρημάτων στους μαστούς. Ο πίνακας περιέχει τις συντεταγμένες και το πάχος των κουκίδων. Έχει σαν πρωτεύων κλειδί το **coordID** που είναι αύξων αριθμός, αυξάνει με κάθε καταχώρηση στον πίνακα.
- Περιέχει το **diagnosisID** σαν ξένο κλειδί προκειμένου να συσχετιστεί με τον πίνακα DiagnosisData. Κάθε διαγνωστικός έλεγχος μπορεί να έχει παραπάνω από μία καταχωρήσεις σε αυτόν τον πίνακα
- **ObserveData** – Περιέχει τα στοιχεία της καρτέλας παρακολούθησης ασθενών. Έχει σαν πρωτεύων κλειδί το **observeID** που είναι αύξων αριθμός, αυξάνει με κάθε καταχώρηση στον πίνακα. Περιέχει το **id** ασθενούς σαν ξένο κλειδί προκειμένου να συσχετιστεί με τον πίνακα BasicData. Κάθε ασθενής μπορεί να έχει παραπάνω από μία καταχωρήσεις σε αυτόν τον πίνακα.



Εικόνα 2. Διάγραμμα σχέσεων οντοτήτων

Ο πίνακας **BasicData** συνδέεται με τους πίνακες **DeseasesData**, **DangersData** και **StadioData** με σχέση ένα προς ένα μέσω του **id** ως κοινού πεδίου. Ο πίνακας BasicData συνδέεται με τους πίνακες **ClinicalData**, **DiagnosisData** και **ObserveData** με σχέση ένα προς πολλά μέσω του **id** ως κοινού πεδίου. Ο πίνακας **DiagnosisData** συνδέεται με τον πίνακα **CoordinatesData** με σχέση ένα προς πολλά μέσω του **diagnosisID** ως κοινού πεδίου.

8. Επεκτάσεις

Η πτυχιακή εργασία αυτή θα μπορούσε μελλοντικά να έχει τις ακόλουθες επεκτάσεις:

- 1) Να γίνει χρήση των σύγχρονων τεχνολογιών ώστε μέσω scanner να καταχωρούνται απευθείας τα στοιχεία και απλά να επαληθεύονται από τον χρήστη.
- 2) Επανασχεδίαση του προγράμματος με τεχνικές multi-threading ώστε να εκμεταλλεύεται στο έπακρο την τεχνολογία των σύγχρονων επεξεργαστών.
- 3) Να γίνει η μετάφραση του στα Αγγλικά και σε άλλες γλώσσες.
- 8) Προσαρμογή του προγράμματος σε διάφορες αναλύσεις κατά την εγκατάσταση του προγράμματος και λειτουργία επιλογών απεικόνισης όπως μέγεθος και χρώμα γραμματοσειράς.
- 9) Επανασχεδίαση του προγράμματος για λειτουργία σε server με ταυτόχρονη πολλαπλή πρόσβαση από απομακρυσμένες θέσεις καθώς και χρήση πρωτοκόλλων αυξημένης ασφαλείας για την μεταφορά των ευαίσθητων ιατρικών δεδομένων. Προϋποθέτει δημιουργία λογαριασμών με αντίστοιχα δικαιώματα πρόσβασης ανάλογα με την ειδικότητα του κάθε χρήστη.
- 10) Επιπλέον λειτουργία για απλοποιημένη δημιουργία SQL queries μέσω γραφικού περιβάλλοντος για την επιστροφή των δεδομένων κατά την εξαγωγή των στατιστικών.
- 11) Επιπλέον δυνατότητα απεικόνισης των ευρημάτων του διαγνωστικού ελέγχου σε 3 διαστάσεις.

Βιβλιογραφία

- [1] Μανιάτης Πέτρος, Δημιουργία ιατρικής βάσης δεδομένων με την βοήθεια της Microsoft Access (2006).
- [2] Dan Appleman, Visual Basic Programmer's Guide to the Win32 API (2009)
- [3] Paul Deitel, H.M Deitel, and G. J. Ayer, Simply Visual Basic 2008 (2010)
- [4] Dan Appleman, Visual Basic Programmer's Guide to the Win32 API (2009)
- [5] Joyce Cox, Joan Lambert, Microsoft Access 2010 Βήμα βήμα (2011)

[6] Αθανάσιος Σπυριδάκος,
Αντικειμενοστραφής προγραμματισμός σε
περιβάλλον Visual Basic.NET (2008)
[7] Εμμανουήλ Σκορδαλάκης. Εισαγωγή στην
Τεχνολογία Λογισμικού. Συμμετρία (1991)

Πηγές από το διαδίκτυο

[1] <http://www.office.microsoft.com>
[2] <http://www.databases.about.com/>
[3] <http://www.vbtutor.net>
[4] <http://www.vb6.us>

[5] <http://www.crystalreports.com>
[6] <http://www.businessobjects.com>
[7] <http://www.ceid.upatras.gr/>
[8] <http://www.w3counter.com/>
[9] <http://office.microsoft.com>
[10] <http://www.dmst.aueb.gr/>
[11] <http://www.wikipedia.org>
[12] <http://www.dotnetperls.com>
[13] <http://www.codeproject.com>
[14] <http://www.processing.org>
[15] <http://www.jrsoftware.org>