



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
«ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ»

ΦΟΙΤΗΤΕΣ

ΚΙΛΙΚΗΣ ΔΗΜΗΤΡΙΟΣ ΑΜ :3653

ΜΟΥΜΟΥΛΙΔΗΣ ΘΕΜΙΣΤΟΚΛΗΣ ΑΜ: 3503

ΠΑΠΑΓΕΩΡΓΟΥΣΗΣ ΑΡΗΣ ΑΜ: 3368

ΗΡΑΚΛΕΙΟ ΚΡΗΤΗΣ 2018

Table of Contents

ΕΥΧΑΡΙΣΤΙΕΣ.....	7
ΠΕΡΙΛΗΨΗ.....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
ABSTRACT.....	9
ΚΕΦΑΛΑΙΟ 1: ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ	10
ΕΙΣΑΓΩΓΗ ΣΤΟ ΕΜΠΟΡΙΟ	10
ΤΟ ΠΡΟΙΟΝ ΚΑΙ ΟΙ ΥΠΗΡΕΣΙΕΣ	10
Η ΑΓΟΡΑ	11
Η ΘΕΩΡΙΑ ΤΟΥ ΤΕΛΕΙΟΥ ΑΝΤΑΓΩΝΙΣΜΟΥ(PERFECT COMPETITION).....	13
ΟΙ ΣΥΝΘΗΚΕΣ ΤΟΥ ΤΕΛΕΙΟΥ ΑΝΤΑΓΩΝΙΣΜΟΥ	14
ΠΑΡΑΔΕΙΓΜΑ ΜΗ ΤΕΛΕΙΟΥ ΑΝΤΑΓΩΝΙΣΜΟΥ - ΜΟΝΟΠΩΛΙΟ	15
ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ ΜΟΝΟΠΩΛΙΟΥ	16
ΤΑ ΕΙΔΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ	17
BUSINESS TO BUSINESS (B-2B)	18
BUSINESS TO CONSUMER (B2C)	18
CONSUMER TO CUSTOMER (C2C)	18
CONSUMER TO BUSINESS (C2B)	18
BUSINESS TO ADMINISTRATOR (B2A)	19
CONSUMER TO ADMINISTRATOR (C2A).....	19
ΠΛΕΟΝΕΚΤΗΜΑΤΑ & ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ	19
ΚΕΦΑΛΑΙΟ 2: ΕΓΚΑΤΑΤΑΣΗ ΤΟΥ ΒΑΣΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ	20
ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA	20
ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA SE DEVELOPMENT KIT 9	21
ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS 8.2 IDE	22
ΚΕΦΑΛΑΙΟ 3: Η ΓΛΩΣΣΑ XML.....	26
ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ XML.....	26
ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΤΗΣ ΓΛΩΣΣΑΣ XML	26
ΠΕΡΙΟΡΙΣΜΟΙ ΣΤΑ XML ΑΡΧΕΙΑ (DTD's & XML Schemas)	27
ΑΝΑΖΗΤΗΣΗ ΚΑΙ ΜΟΡΦΟΠΟΙΗΣΗ(ΧΡΑΤΗ – XSLT)	33
ΚΕΦΑΛΑΙΟ 4: ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	35

ΟΙ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ΠΕΛΑΤΗ	35
ΕΓΓΡΑΦΗ ΠΕΛΑΤΗ ΣΤΟ ΣΥΣΤΗΜΑ	35
ΕΙΣΟΔΟΣ ΠΕΛΑΤΗ ΣΤΟ ΣΥΣΤΗΜΑ	37
ΠΡΟΣΘΗΚΗ ΣΤΟ ΚΑΛΑΘΙ ΠΡΟΙΟΝΤΩΝ	39
ΔΙΑΓΡΑΦΗ ΠΑΡΑΓΓΕΛΙΑΣ	43
ΕΚΤΥΠΩΣΗ ΠΡΟΙΟΝΤΩΝ	45
ΟΙ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ADMIN	49
ADD COMPANY	49
ADD SUBSTORE	52
ADD/REMOVE PRODUCTS	57
ADD/REMOVE LAPTOPS.....	57
UPDATE ORDER	66
PRINT ORDERS	70
ΣΥΜΠΕΡΑΣΜΑΤΑ	79
ΒΙΒΛΙΟΓΡΑΦΙΑ	81

Table of Figures

Εικόνα 1: Ο τέλειος ανταγωνισμός [Retrieved from Here]	13
Εικόνα 2: Ο ανταγωνισμός του μονοπωλίου[Retrieved from Here]	16
Εικόνα 3: Η εγκατάσταση της Java[Retrieved from Here]	20
Εικόνα 4: Η εγκατάσταση της Java (b)	21
Εικόνα 5: Εγκατάσταση της JAVA SE Kit 9.0.1 [Retrieved from Here]	22
Εικόνα 6: Εγκατάσταση NetBeans8.2 και JDK[Retrieved from Here]	23
Εικόνα 7: Το interface του NetBeans IDE 8.2.....	23
Εικόνα 8: Η επιλογή Edit του NetBeans.....	24
Εικόνα 9: Η επιλογή File του NetBeans.....	24
Εικόνα 10: Παράδειγμα XML (1).....	27
Εικόνα 11: Παράδειγμα XML (2).....	27
Εικόνα 12: Παράδειγμα DTD (1)	29
Εικόνα 13: Παράδειγμα DTD'S (2a) [Retrieved from Here]	31
Εικόνα 14: Παράδειγμα XML Schema [Retrieved from Here]	32
Εικόνα 15: Παράδειγμα με Complex Type[Retrieved from Here]	33
Εικόνα 16: Αναζήτηση σε XML αρχείο [Retrieved from Here]	33
Εικόνα 17: Μορφοποίηση XML με χρήση XSLT[Retrieved from Here].....	34
Εικόνα 18: Εγγραφή πελάτη στο σύστημα	35
Εικόνα 19: Εγγραφή πελάτη στο σύστημα(2).....	35
Εικόνα 20: Το XML αρχείο customers_XML.xml.....	36
Εικόνα 21: Εγγραφή ενός πελάτη	36
Εικόνα 22: Εγγραφή του χρήστη στο σύστημα – XML.....	37
Εικόνα 23: Είσοδος του πελάτη στο σύστημα (1).....	38
Εικόνα 24: Είσοδος πελάτη στο σύστημα (2)	38
Εικόνα 25: Το κεντρικό JFrame του πελάτη.....	39
Εικόνα 26: Προσθήκη προϊόντων στο καλάθι αγορών.....	39
Εικόνα 27: Το αρχείο unfinished_order_laptops_XML.xml.....	40
Εικόνα 28: Update του αρχείου unfinished_order_laptops_XML.xml.....	40
Εικόνα 29: Κανοντας append τρία νέα πεδία.....	41
Εικόνα 30: Ανοίγουμε τα unfinished_order αρχεία.....	41
Εικόνα 31: Παίρνουμε τα elements των αρχείων unfinished_order.....	42
Εικόνα 32: Κάνουμε appending στα αρχεία unfinished_order	42
Εικόνα 33: Πώς γράφουμε σε ένα αρχείο XML με χρήση DOM API.....	43
Εικόνα 34: delete_order button	43
Εικόνα 35: Διαγραφή mobiles από το καλάθι (1).....	44
Εικόνα 36: Κλήση της συνάρτησης removeEmptyNodes και update του XML.....	45
Εικόνα 37: JFrame για εκτύπωση προϊόντων	45
Εικόνα 38: JTextArea για εκτύπωση τηλεοράσεων	46

Εικόνα 39: Το JFrame για εκτύπωση προϊόντων ανα εταιρεία	46
Εικόνα 40: Τα mobiles της εταιρείας PUBLIC	47
Εικόνα 41: Τα televisions της εταιρείας PUBLIC	47
Εικόνα 42: Τα laptops της εταιρείας PUBLIC	48
Εικόνα 43: Κώδικας για εκτύπωση laptops	48
Εικόνα 44: Το κεντρικό JFrame του admin	49
Εικόνα 45: Το JFrame add company	50
Εικόνα 46: Το αρχείο companies_XML.xml	50
Εικόνα 47: Εισαγωγή εταιρείας σε XML αρχείο	50
Εικόνα 48: Η εταιρεία ZAFEIRIADIS έχει προστεθεί στο XML αρχείο	51
Εικόνα 49: Εισαγωγή JFrame για την company	51
Εικόνα 50: Έλεγχος για διπλότυπα ονόματα και TIN εταιριών	52
Εικόνα 51: appending εταιριών στο companies_XML.xml	52
Εικόνα 52: Το JFrame για εισαγωγή substore	53
Εικόνα 53: Το αρχείο substores_XML.xml	53
Εικόνα 54: Εισαγωγή υποκαταστήματος στο σύστημα.....	54
Εικόνα 55: Εισαγωγή του υποκαταστήματος Public Chios.....	54
Εικόνα 56: JFrame και έλεγχος διπλότυπων χαρακτηριστικών για το substores_XML.xml.....	55
Εικόνα 57:Κλήση κατάλληλης συνάρτησης για εισαγωγή ενός υποκαταστήματος	55
Εικόνα 58: appending υποκαταστήματα στο substores_XML.xml.....	56
Εικόνα 59: Κεντρικό JFrame για add/remove προϊόντων	57
Εικόνα 60: Add/remove laptops απο το σύστημα.....	57
Εικόνα 61: Εισαγωγή laptop στο σύστημα	58
Εικόνα 62: Το αρχείο laptops_XML.xml.....	58
Εικόνα 63: Εισαγωγή laptop στο laptops_XML.xml.....	59
Εικόνα 64: Το μοντέλο SAMSUNG HT-82 έχει εγγραφεί στο αρχείο.....	59
Εικόνα 65: Κώδικα για την εισαγωγή ενός laptop.....	60
Εικόνα 66:Κώδικα για την εισαγωγή ενός laptop (2)	61
Εικόνα 67: Κλήση της συνάρτησης AddLaptop.....	61
Εικόνα 68: JFrame για διαγραφή προϊόντος απο το mobiles_XML.xml.....	62
Εικόνα 69: Βασικό JFrame για διαγραφή mobile	63
Εικόνα 70: Το αρχείο mobiles_XML.xml πριν τη διαγραφή mobiles.....	63
Εικόνα 71: Διαγραφή mobile απο το αρχείο mobiles_XML.xml	64
Εικόνα 72: Διαγραφή 11ης γραμμής απο το αρχείο mobiles_XML.xml.....	64
Εικόνα 73 : Διαγραφή 36ης γραμμής απο το αρχείο mobiles_XML.xml.....	64
Εικόνα 74: Διαγραφή 54ης γραμμής απο το αρχείο mobiles_XML.xml.....	65
Εικόνα 75: Διαγραφή κινητού τηλεφώνου απο το αρχείο mobiles_XML.xml	66
Εικόνα 76: Συναρτήσεις για θεώρηση μιας παραγγελίας ως ολοκληρωμένη	67
Εικόνα 77: Εισαγωγή των laptops απο το unfinished_order στο finished_order αρχείο	68
Εικόνα 78: Αφαίρεση laptops απο απο το αρχείο unfinished_order_laptops_XML.xml.....	69
Εικόνα 79: JFrame για finish order	69
Εικόνα 80: Διαγραφή προϊόντων απο το καλάθι αγορών.....	70
Εικόνα 81: Βασικό JFrame για εκτύπωση παραγγελιών και κέρδη υποκαταστημάτων	70
Εικόνα 82: Εύρεση παραγγελίας ανα username	71
Εικόνα 83: Εκτύπωση παραγγελίας (laptops).....	71

Εικόνα 84: Εκτύπωση παραγγελίας (mobiles)	72
Εικόνα 85: Εκτύπωση παραγγελίας (televisions).....	72
Εικόνα 86: Εκτύπωση καλαθιού αγοράς πελάτη.....	72
Εικόνα 87: Εκτύπωση ανα είδος προϊόντων.....	73
Εικόνα 88: Εκτύπωση κινητών τηλεφώνων - ολοκληρωμένη παραγγελία	74
Εικόνα 89: Εκτύπωση σε σχέση με το χρόνο ολοκλήρωσης της παραγγελίας.....	74
Εικόνα 90: Εκτύπωση σύμφωνα με το χρόνο ολοκλήρωσης παραγγελίας.....	75
Εικόνα 91: Κλήση της συνάρτησης printAllProfits()	75
Εικόνα 92: Κλήση της συνάρτησης getProfitsWithGivenSubstoreName	76
Εικόνα 93: Υπολογισμός τελικού κέρδους για τα laptops	76
Εικόνα 94:Υπολογισμός τελικού κέρδους για τα mobiles	77
Εικόνα 95: Υπολογισμός τελικού κέρδους για τις televisions	78
Εικόνα 96:Κέρδη υποκαταστημάτων.....	78
Εικόνα 97: Εκτύπωση κερδών υποκαταστημάτων.....	79

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας Παπαδάκη Νικόλαο για την παροχή πολύτιμων συμβουλών κατά την διάρκεια της εκπόνησης της. Κατόπιν τους γονείς μας που με τόσο κόπο μας στήριξαν αυτά τα χρόνια με την αγάπη και την συμπαράσταση τους.

Τέλος θα θέλαμε να ευχαριστήσουμε λοιπούς συγγενείς και φίλους που στάθηκαν δίπλα μας όλο αυτόν τον καιρό και ειδικά τον κοινό μας φίλο Όγλου Γεώργιο για την πολύτιμη βοήθεια στην κατανόηση της XML δίνοντας μας την δύναμη να συνεχίζουμε το έργο μας.

ΠΕΡΙΛΗΨΗ

Σε πρώτη φάση αναφερόμαστε στα είδη του ηλεκτρονικού εμπορίου όπου αναλύουμε εκτενέστερα τις δυνατότητες και την χρησιμότητα του στις μέρες μας. Κατόπιν στα είδη της αγοράς των προϊόντων και υπηρεσιών. Έπειτα αναφερόμαστε στη θεωρία του τέλειου ανταγωνισμού, στα χαρακτηριστικά γνωρίσματα της, και έπειτα στη θεωρία του μονοπωλίου.

Πιο μετά, αναλύουμε επιγραμματικά τους κανόνες, και τον τρόπο δημιουργίας, αποθήκευσης, μορφοποίησης αλλά και αναζήτησης μέσα σε ένα αρχείο XML. Παρουσιάζουμε τις μεθόδους XSLT, XPATH καθώς και όλα τα απαραίτητα δομικά στοιχεία ενός αρχείου XML.

Στη συνέχεια θα προχωρήσουμε στην εγκατάσταση του απαραίτητου λογισμικού για να μπορέσουμε να αναπτύξουμε τον κώδικα μας που βασίστηκε σε Java και τη χρήση XML αρχείων ως μεθόδους αποθήκευσης αλλά και ανάκτησης δεδομένων. Σε επόμενο κεφάλαιο θα παρουσιάσουμε εκτενώς τον κώδικα μας και θα τρέξουμε διάφορα σενάρια αυτού.

ABSTRACT

First of all we discuss about e-Commerce types. Then we explain the theory of perfect and monopoly competition. After that we talk about the XML language. The way of how to store, retrieve and present XML files using XPATH and XSLT methods.

Then, we try to install the appropriate software in order to write our code. Explain in detail how to install Java, JAVA SE and NetBeans 8.2 platform. Finally we present our code step by step and test its functioning.

ΚΕΦΑΛΑΙΟ 1: ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ

ΕΙΣΑΓΩΓΗ ΣΤΟ ΕΜΠΟΡΙΟ

Το εμπόριο δεν είναι τίποτε άλλο από την ικανότητα αγοράς και πώλησης αγαθών. Στις μέρες μας είναι το πιο σημαντικό πλαίσιο μιας επιχείρησης καθώς μέσω αυτού μπορεί να βγάλει κέρδος. Το εμπόριο περιλαμβάνει την αγοραπωλησία αγαθών αλλά και υπηρεσιών σε απλούς καταναλωτές αλλά και σε εταιρείες (Chisholm, 1911)

ΤΟ ΠΡΟΙΟΝ ΚΑΙ ΟΙ ΥΠΗΡΕΣΙΕΣ

Πιο αναλυτικά το **προϊόν** είναι το οτιδήποτε μπορεί κάποιος να διαθέσει στην αγορά το οποίο να αποτελεί ανάγκη για κάποιον άλλο άνθρωπο ή και εταιρεία (Kotler, Armstrong, Brown, Adam, 2006)

Από την άλλη πλευρά ένα προϊόν μπορεί να χαρακτηριστεί με τους δύο παρακάτω τρόπους ανάλογα με το είδος του (Klein, Robinson, 2011):

- **Άυλο περιουσιακό στοιχείο.** Δεν διαθέτει υλική υπόσταση και είναι συνήθως αρκετά δύσκολο να εκτιμηθεί η τιμή αγοραπωλησίας του. Συνήθως μπορεί να είναι ένα από τα παρακάτω:
 - Δικαιώματα
 - Πατέντες
 - Εμπορικά σήματα
 - Εμπορικά ονόματα
 - Λογισμικό
- **Υλικό περιουσιακό στοιχείο.** Οτιδήποτε μπορεί να αγοραστεί ή να πωληθεί ενώ διαθέτει φυσική υπόσταση (ύλη).

Δηλαδή, ένα προϊόν δεν είναι τίποτε άλλο από έναν οικονομικό πόρο που μπορεί να αξιοποιηθεί κατάλληλα για να παράγει αξία (Donaldson, Federal, 2007)

Συνήθως για την αξιολόγηση ενός προϊόντος χρησιμοποιείται το σύστημα Aspinwall το οποίο **αξιολογεί** ένα προϊόν σύμφωνα με τις παρακάτω προδιαγραφές (Kotler, Armstrong, 1989):

- Πόσο συχνά ένα προϊόν επαναγοράζεται.
- Πόσο κέρδος μπορούμε να έχουμε από την πώληση ενός προϊόντος.

- Πόσο ευέλικτοι είναι οι αγοραστές σε σχέση με την αγορά του προϊόντος.
- Για πόσο καιρό το προϊόν θα παρέχει ικανοποίηση στον πελάτη.
- Πόσο συχνά οι πελάτες αγοράζουν το προϊόν.

Απο την άλλη πλευρά εκτός από προϊόντα διαθέτουμε και **υπηρεσίες**. Υπηρεσίες μπορεί να είναι πόροι, ικανότητες αλλά και η εμπειρία. Οι υπηρεσίες μπορούν να χωριστούν στις παρακάτω βασικές κατηγορίες(Kotler, Armstrong, 1989):

- **Ανεπίδεκτες.** Δηλαδή δεν διαθέτουν φυσική υπόσταση
- **Φθαρτότητα.** Για παράδειγμα επειδή οι υπηρεσίες είναι άυλες δεν μπορούν να αποθηκευτούν κάπου, επομένως παράγονται και καταναλώνονται ταυτόχρονα
- **Αδιαχώριστες.** Δεν γίνεται δηλαδή να επαναλάβεις στην ίδια τοποθεσία, στις ίδιες συνθήκες και καταστάσεις ακριβώς την ίδια υπηρεσία.
- **Συμμετοχή.** Πρέπει και ο πάροχος της υπηρεσίας αλλά και ο καταναλωτής αυτής να συμμετέχουν ενεργά.

Τέλος η μεταφορά μιας υπηρεσίας αποτελείται από τους παρακάτω έξι παράγοντες(Kotler, Armstrong, 1989) :

- Τους παρόχους υπηρεσίας.
- Ο εξοπλισμός που χρησιμοποιείται για την παροχή της υπηρεσίας
- Το άτομο που θα καταναλώσει τη συγκεκριμένη υπηρεσία
- Τοποθεσία
- Λοιποί πελάτες που θα καταναλώσουν την υπηρεσία κατά την διαδικασία της μεταφοράς της.
- Ο τρόπος επικοινωνίας με τον καταναλωτή.

Η ΑΓΟΡΑ

Η αγορά είναι μια από τις πολλές μορφές των συστημάτων, των θεσμών των κοινωνικών σχέσεων αλλά και των υποδομών που συναλλάσσονται(Marshall, 1890)

Βασίζεται συνήθως όχι τόσο στην επικοινωνία αυτών των μορφών αλλά στους πωλητές που προσφέρουν την υπηρεσία τους στους αγοραστές. Χάρη στην αγορά διαθέτουμε τιμή για κάθε προϊόν και υπηρεσία.

Επιπρόσθετα είναι υπεύθυνη για την διανομή και κατανομή των αγαθών στο δίκτυο εμπόρων, αγοραστών, πωλητών αλλά και εργαζομένων καθώς αυτοί με τον τρόπο τους κινούν την αγορά(O'Sullivan, Sheffrin, Steven, 2004).

Δεν είναι όλες οι αγορές ίδιες. Κάθε αγορά απευθύνεται σε διαφορετικό αγοραστικό κοινό ενώ το εκάστοτε κράτος φορολογεί τα αγαθά και τις υπηρεσίες(Marshall, 1890).

Η τιμή μιας υπηρεσίας και ενός αγαθού συνήθως δε μένει σταθερή καθώς εξαρτάται αποκλειστικά από την ικανότητα εύρεσης ενός αγαθού(πόσο σπάνιο είναι). Η αγορά μπορεί να χωριστεί στις παρακάτω πέντε κατηγορίες(Heyne, Boettke, 2014) :

- **Αγορά των φυσικών επιχειρήσεων.** Εδώ ανήκει η χονδρική αγορά και πώληση, η αγορά των ενδιάμεσων προϊόντων, η αγορά εργασίας και οι αγορές που αφορούν περιπτώσεις δημοπρασίας.
- **Φυσικές καταναλωτικές αγορές.** Εδώ ανήκουν οι αγορές λιανικής πώλησης των προϊόντων, τα καταστήματα retail αλλά και οι προσωρινές αγορές.
- **Μη φυσικές αγορές.** Εδώ βρίσκουμε την αγορά των πολυμέσων όπως είναι για παράδειγμα η παροχή τηλεοπτικών σταθμών, διαδικτυακές εφημερίδες κ.α, το ηλεκτρονικό εμπόριο αλλά και οι τεχνητές αγορές που έχουν σκοπό τη βελτίωση των εξωτερικών παραγόντων
- **Παράνομες αγορές.** Εδώ μπορούμε να βρούμε πώληση παράνομων υπηρεσιών και αγαθών όπως μπορεί να είναι για παράδειγμα η πώληση ναρκωτικών ουσιών και εμπόριο όπλων.

Η αγορά είναι ένα σύστημα άρα διαθέτει και δομή. Ένα τέτοιο σύστημα ορίζεται από την **θεωρία του τέλειου ανταγωνισμού** που θα εξετάσουμε σε επόμενη ενότητα. Καμία αγορά δεν θεωρείται τέλεια αλλά όμως διαθέτει τα παρακάτω χαρακτηριστικά: (Heyne, Boettke, 2014)

- Πολλούς μικρούς αγοραστές και καταναλωτές.
- Οι καταναλωτές και οι αγοραστές έχουν πρόσβαση στην ίδια πληροφορία.
- Τα προϊόντα έχουν την δυνατότητα να μπορούν να συγκρίνονται με κάποια άλλα.

Η ΘΕΩΡΙΑ ΤΟΥ ΤΕΛΕΙΟΥ ΑΝΤΑΓΩΝΙΣΜΟΥ(PERFECT COMPETITION)

Η θεωρία αυτή προσπαθεί να ορίσει τα θεμέλια μιας ιδανικής αγοράς που συνήθως ονομάζεται αγορά του τέλειου ανταγωνισμού.

Προσπαθούμε μέσω αυτής της θεωρίας να πειραματιζόμαστε και να εξομαλύνουμε τις αγορές, όσο το δυνατόν πιο αποδοτικά(Gerard, 1927)

Παρακάτω φαίνονται τα χαρακτηριστικά του τέλειου ανταγωνισμού(*Εικόνα : 1*).



Εικόνα 1: Ο τέλειος ανταγωνισμός [Retrieved from [Here](#)]

Για να θεωρηθεί μια αγορά ότι είναι αρκετά κοντά στην αγορά του τέλειου ανταγωνισμού απαιτείται να ακολουθεί τις **παρακάτω συνθήκες**, που αρχικά θα παρουσιάσουμε επιγραμματικά ενώ σε επόμενες ενότητες όσο το δυνατόν γίνεται πιο αναλυτικά(Roberts 1987):

- Απαιτείται ένας μεγάλος αριθμός από αγοραστές και καταναλωτές
- Τέλεια πληροφόρηση και στα δύο άκρα επικοινωνίας που είναι οι καταναλωτές με τους αγοραστές και αντίστροφα
- Ομοιγένεια στα προϊόντα. Δηλαδή το ίδιο προϊόν που πληρεί κάποια συγκεκριμένα χαρακτηριστικά να μην εξαρτάται από τον κατασκευαστή ή παροχό του προϊόντος
- Σωστός καθορισμός των δικαιωμάτων ιδιοκτησίας.

- Κανένας απο τους συμμετέχοντες στην αγορά να μη διαθέτει την μεγαλύτερη δύναμη στον καθορισμό των τιμών των προϊόντων.
- Οι παράγοντες παραγωγής ενός προϊόντος να είναι ευέλικτοι σε περίπτωση που οι συνθήκες τις αγορές θα είναι μεταβαλλόμενες.
- Μέγιστη τιμη που μπορεί να φτάσει ενα προϊόν.
- Να μην υπάρχουν εξωτερικοί παράγοντες. Τα κόστη και τα ωφέλη απο την αγοραπωλησία αγαθών και υπηρεσιών να μην επηρεάζουν τρίτους.
- Να μην υπάρχουν κόστη μεταφοράς των αγαθών στους αγοραστές και καταναλωτές.
- Να υπάρχει προστασία σε περίπτωση που η αγορά ξεφύγει απο έλεγχο και οι λίγοι διαθέτουν την μεγαλύτερη ισχύ. Θέλουμε ομοιομορφία.

ΟΙ ΣΥΝΘΗΚΕΣ ΤΟΥ ΤΕΛΕΙΟΥ ΑΝΤΑΓΩΝΙΣΜΟΥ

Μια απο τις συνθήκες του τς θεωρίας του τέλειου ανταγωνισμού είναι η «τέλεια πληροφορία». Έχοντας τέλεια πληροφόρηση και οι καταναλωτές αλλά και οι αγοραστές όσο αφορά την τιμή, ποιότητα, και την μέθοδο παραγωγής των προϊόντων ευθύς αμέσως η αγορά γίνεται όλο και πιο αποδοτική(Roberts 1987).

Για την αξιολόγηση της τέλειας πληροφορίας εφαρμόζεται η **θεωρία των παιγνίων**(game theory). Ένα αλλο χαρακτηριστικό είναι η εμπορευματοποίηση ενός προϊόντος. Προσπαθούμε δηλαδή να εμπορευματοποιήσουμε όσο το δυνατόν περισσότερα πράγματα ανάλογα με τις εκάστοτε ανάγκες που έχουν οι καταναλωτές. Το προϊόν για να μπορέσει να συμβεί αυτό πρέπει να είναι ανταλλάξιμο(Roberts 1987).

Απο την άλλη πλευρά, τα δικαιώματα ιδιοκτησίας είναι απαραίτητα για την χρήση ή πώληση ενός αγαθού. Παρακάτω αναφέρουμε επιγραμματικά τις συνιστώσες αυτού του χαρακτηριστικού:

- Το **δικαίωμα** χρήσης ενός προϊόντος
- Το **δικαίωμα** να μπορεί κάποιος να βγάζει εισόδημα απο το προϊόν
- Το **δικαίωμα** να μπορεί να μεταφέρει αυτό το προϊόν σε άλλα άτομα
- Το **δικαίωμα** να μπορεί να εφαρμόζει σε αυτό το προϊόν, δικαιώματα ιδιοκτησίας.

Αλλα χαρακτηριστικά είναι το **φράγμα** που ορίζει την ελάχιστη τιμή και την μέγιστη τιμή ενος προϊόντος. Αυτό γινεται εύκολα αντιληπτό διότι εαν ενα προϊόν βγει στην αγορά εργασίας με πολύ μικρή τιμή αμέσως οι καταναλωτές σπευδουν σε αυτό και έτσι υπάρχει ανατάραξη της αγοράς(Roberts 1987).

Επιπρόσθετα ένα άλλο χαρακτηριστικό είναι το **πόσο δύναμη μπορεί να αποκτήσει** η αγορά ανάλογα με το πόσο αυξάνει ή μειώνει την τιμή ενός αγαθού. Πρέπει δηλαδή όσο γίνεται να προστατευτούμε από το μονοπώλιο (Gerard, 1927).

Εάν μια ομάδα επιχειρήσεων συμφωνήσουν να επηρεάσουν την τιμή της αγοράς ή την παραγωγή ονομάζεται **καρτέλ** (cartel). Επίσης, ένα ακόμη χαρακτηριστικό που πρέπει να προσπαθήσουμε να ελέγξουμε, εάν φυσικά θέλουμε να έχουμε τέλει ανταγωνισμό είναι τον παράγοντα της **κίνησης των συντελεστών**.

Παράλληλα, η μετανάστευση δημιουργεί προβλήματα σε ένα τέτοιο σύστημα διότι γίνονται αλληπάληλες διαθέσεις δανείων, χωρίς έλεγχο και φρένο με αποτέλεσμα να φτάσουν σε σημείο οι κυβερνήσεις να απαγορεύουν την μετανάστευση, την διαρροή κεφαλαίων και την άμεση επένδυση (Gerard, 1927).

Τέλος, σημαντικό είναι ο παράγοντας της δημιουργίας υπηρεσιών που βλάπτουν το περιβάλλον. Τέτοιες υπηρεσίες δεν θα πρέπει να υπάρχει στο σύστημα μας καθώς όχι μόνο καταστρέφει την υγεία των πολιτών, αλλά προκαλεί πρόσθετα κόστη στο κράτος για την δημιουργία μέτρων προστασίας.

ΠΑΡΑΔΕΙΓΜΑ ΜΗ ΤΕΛΕΙΟΥ ΑΝΤΑΓΩΝΙΣΜΟΥ - ΜΟΝΟΠΩΛΙΟ

Το μονοπώλιο είναι ένα παράδειγμα μη τελείου ανταγωνισμού καθώς ο κάθε πωλητής που βρίσκεται στην αγορά διαθέτει τα αγαθά του σε όποια τιμή αυτός θέλει με αποτέλεσμα να μην υπάρχει ισορροπία στο σύστημα (Krugman, Obstfeld, 2008).

Τα χαρακτηριστικά του μονοπωλίου είναι τα κάτωθι (Goodwin, Nelson, Ackerman, Weisskopf, 2009):

- Πολύ μεγάλα κέρδη
- Ορίζεται η τιμή αλλά όχι η διαθεσιμότητα του προϊόντος
- Υψηλά φράγματα τα οποία δεν επιτρέπουν σε μικρούς πωλητές να εισέλθουν σε αυτό του είδους την αγορά
- Μονοπώληση.

- Η τιμή αλλάζει σύμφωνα με τα άτομα που ορίζουν την αγορά του μονοπωλίου.

Πολλές φορές όμως και η ίδια η κυβέρνηση εκθειάζει το μονοπώλειο. Για παράδειγμα η παροχή ευρεσιτεχνιών, δικαιώματα, εμπορικά σήματα κτλ. Στην παρακάτω εικόνα συνοψίζονται τα χαρακτηριστικά του μονοπωλίου (**Εικόνα : 2**).



Εικόνα 2: Ο ανταγωνισμός του μονοπωλίου [Retrieved from [Here](#)]

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ ΜΟΝΟΠΩΛΙΟΥ

Το μονοπώλιο διαθέτει τα παρακάτω έξι χαρακτηριστικά (Krugman, Obstfeld, 2008):

- Διαφοροποίηση των προϊόντων
- Πολλές επιχειρήσεις
- Ανεξάρτητος τρόπος λήψης αποφάσεων
- Ελευθερία στην είσοδο και στην έξοδο από την αγορά
- Αρκετή δύναμη στην αγορά
- Οι καταναλωτές και οι αγοραστές δεν διαθέτουν πλήρη πληροφόρηση.

Παρόλα αυτά όμως, το μονοπώλιο έχει και προβλήματα (Blinder, Baumol, 2001). Συνήθως οι εταιρείες που βρίσκονται σε ανταγωνισμό, είναι αναποτελεσματικές και αυτό συμβαίνει διότι το κόστος ρύθμισης των τιμών για τα προϊόντα που πωλούνται χρησιμοποιώντας μονοπωλιακό ανταγωνισμό, υπερβαίνει τα σημαντικά ωφέλη μιας τέτοιας ρύθμισης.

ΤΑ ΕΙΔΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

Το ηλεκτρονικό εμπόριο(e-Commerce) είναι η αγορά και η πώληση προϊόντων και υπηρεσιών διαμέσου του διαδικτύου. Στην εποχή μας, ένα πολύ μεγάλο ποσοστό ανθρώπων αγοράζουν προϊόντα από το διαδίκτυο, ευκολα , γρήγορα και συνήθως αξιόπιστα γλιτώνοντας χρόνο από την επίσκεψη τους σε ένα μέρος πώλησης προϊόντων(Bloomidia, 2014).

Υπερέχει από το κλασικό εμπόριο γιατί έγκειται στην ταχύτητα στην ευκολία όπου μπορεί κάποιος να αγοράσει ή να πουλήσει ένα προϊόν. Το σύστημα των εμπορικών ανταλλαγών περιλαμβάνει διάφορους τομείς της τεχνολογίας(Bloomidia, 2014):

- Διαχείριση εταιρικού περιεχομένου
- Συστήματα ανταλλαγής προσωπικών μηνυμάτων
- Ψηφιακές αγορές
- Διεθνή και εθνικά ηλεκτρονικά συστήματα συναλλαγών
- Email και newsgroups

Όσο αφορά τον τρόπο λειτουργίας ενός τέτοιου συστήματος, συνήθως υπάρχει μια πλατφόρμα που παρέχει αυτόν τον τρόπο αγοραπωλησίας προϊόντων. Δύο μεγάλα παραδείγματα είναι η Amazon και το e-Bay.

Ένα τέτοιο σύστημα διαθέτει ένα καλάθι αγορών μέσω του οποίου ένας πελάτης μπορεί να εισάγει προϊόντα που έχει ήδη επιλέξει μέσα από την πλατφόρμα. Όταν αποφασίσει ότι θέλει να αγοράσει αυτά τα προϊόντα τότε το καλάθι αγορών αυτομάτως μετατρέπεται σε παραγγελία και έτσι στη συνέχεια του ζητούνται κάποια βασικά στοιχεία όπως η διεύθυνση παραλαβής, τηλέφωνο.

Όσο αφορά όμως την πληρωμή συνήθως γίνεται με κάποια πιστωτική ή χρεωστική κάρτα, αν και ακόμη αν και προϊόντα παραγγέλνονται από το διαδίκτυο, υπάρχει η επιλογή της αντικαταβολής στην μεταφορική υπηρεσία που θα φέρει τα προϊόντα στο χώρο όπου ο πελάτης ζήτησε. Στις παρακάτω ενότητες θα αναφερθούμε στα είδη του ηλεκτρονικού εμπορίου ξεχωριστά.

BUSINESS TO BUSINESS (B-2B)

Εδώ είναι το μοντέλο όπου μια εταιρεία κάνει συναλλαγές με μια άλλη εταιρεία. Ένα παράδειγμα τέτοιου μοντέλου είναι όταν μια εταιρεία προσλαμβάνει οικονομολόγους και λογιστές για την διαχείριση των οικονομικών της (Huczynski, 2011).

BUSINESS TO CONSUMER (B2C)

Είναι ένα είδος ηλεκτρονικού εμπορίου όπου διαφέρει από την κλασσική επικοινωνία με τη χρήση διαδικτύου. Είναι κατά έναν τρόπο παραδοσιακό εμπόριο με τη διαφορά ότι μπορεί οι συναλλαγές να γίνονται μέσω διαδικτύου αλλά πολλές φορές χρειάζεται ο πελάτης να έχει και παρουσία στην εταιρεία. Ένα παράδειγμα είναι η αγορά αυτοκινήτου, παπουτσιών ή υπολογιστών (Bloomidia, 2014).

CONSUMER TO CUSTOMER (C2C)

Υπάρχει έμφαση στο διαδίκτυο. Συνήθως ένας τρίτος διαθέτει την πλατφόρμα για την αγοραπωλησία προϊόντων, όπου δύο μέρη επικοινωνούν μεταξύ τους καθαρά ηλεκτρονικά. Για παράδειγμα η πλατφόρμα e-Bay παρέχει μια πλατφόρμα ηλεκτρονικού εμπορίου όπου οι πωλητές και οι πελάτες συναλλάσσονται. Η e-Bay παίρνει ένα μικρό κέρδος από την πώληση προϊόντων και με αυτόν τον τρόπο βγάζει το κέρδος της (Bloomidia, 2014).

Η μέθοδος χρήσης του ηλεκτρονικού εμπορίου με αυτόν τον τρόπο, έχει και πλεονεκτήματα αλλά και μειονεκτήματα. Ένα από τα θετικά είναι ότι λόγω της χρήσης διαδικτύου καταφέραμε να εξαλείψουμε τον «μεσάζοντα». Από την άλλη πλευρά, είναι πάρα πολύ οικονομικό και εύκολο να ξεκινήσει κάποιος την διαδικτυακή του επιχείρηση από αύριο κιόλας (Bloomidia, 2014).

Θα μπορούσε για παράδειγμα να δημιουργήσει μια ιστοσελίδα όπου θα έχει πάνω όλα τα προϊόντα του και στη συνέχεια μέσω χρήσης PayPal να λαμβάνει τις πληρωμές του.

CONSUMER TO BUSINESS (C2B)

Αυτό διαφέρει από τα σύνηθη μοντέλα αγοραπωλησίας. Ένας αγοραστής μπορεί να ζητήσει να δημιουργηθεί ένα συγκεκριμένο προϊόν ή υπηρεσία και ένα μεγάλο πλήθος ατόμων να προσπαθήσουν να το κατασκευάσουν. Το άτομο που θα κινήσει την περιέργεια του αγοραστή, καθώς θα έχει δημιουργήσει κατά εκείνον, το καλύτερο προϊόν σε σχέση με τους άλλους ανταγωνιζόμενους, θα βγάλει κέρδος (Bloomidia, 2014).

Παράδειγμα αυτού είναι η δημιουργία εμπορικού σήματος(logo) απο μερικούς σχεδιαστές ιστοσελίδων. Ένας αγοραστής παραθέτει τις προδιαγραφές που πρέπει να διαθέτει το logo της επιχείρησης και ευθύς αμέσως, οι σχεδιαστές ιστοσελίδων ανταγωνίζονται για το ποιός θα δημιουργήσει το καλύτερο logo.

BUSINESS TO ADMINISTRATOR (B2A)

Στο συγκεκριμένο μοντέλο αναφερόμαστε στις συναλλαγές μεταξύ εταιρειών και πολιτείας.Αγοραπωλησία εμπιστευτικών εγγράφων, νομικά έγγραφα και οικονομικά έγγραφα αποτελούν ένα παράδειγμα χρήσης τέτοιου μοντέλου(Bloomidia, 2014).

CONSUMER TO ADMINISTRATOR (C2A)

Έμφαση δίνεται στις ηλεκτρονικές συναλλαγές μεταξύ ατόμων και τη δημόσια διοίκηση.Μερικές απο αυτές μπορεί να είναι η εκπαίδευση πχ εξαποστάσεως, η φορολόγηση του πληθυσμού, ραντεβού στα ιατρεία κ.α

ΠΛΕΟΝΕΚΤΗΜΑΤΑ & ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

Το **βασικό πλεονέκτημα** είναι η ικανότητα να μπορέσουμε να μπορέσουμε να εισέλθουμε στην παγκόσμια αγορά χωρίς να χρειαστεί να ξοδέψουμε ένα πολύ μεγάλο κεφάλαιο(Bloomidia, 2014).

Επομένως με λίγα χρήματα μπορούμε να προωθήσουμε το προϊόν ή την υπηρεσία μας σε εκατομμύρια πελάτες ανα τον κόσμο. Απο την άλλη πλευρά **επιτρέπει** να έχει ο πελάτης και ο πάροχος των προϊόντων μια πιο στενή επαφή για οποιαδήποτε απορία και αν διαθέτει(Krugman, Wells, 2009).

Στις ημέρες μας τα ηλεκτρονικά καταστήματα **είναι ανοιχτά** εικοσιτέσσερις ώρες το εικοσιτετράωρο, καθώς διαθέτει άτομα για την επικοινωνία με τους πελάτες όλο αυτό το χρονικό διάστημα. Για τυχόν απορίες, παραλείψεις το κατάστημα είναι εκεί να εξυπηρετήσει ανα πάσα στιγμή(Krugman, Wells, 2009)..

Οσο αφορά τα μειονεκτήματα μπορούμε να πούμε οτι οι πελάτες δεν μπορούν να δοκιμάσουν τα προϊόντα πριν τα παραλάβουν και η έλλειψη εμπιστοσύνης οσο αφορά την παραχώρηση εμπιστευτικών στοιχείων στο κατάστημα, διότι καθώς αυτό βρίσκεται στο διαδίκτυο, γίνεται αμέσως ευάλωτο σε κυβερνοεπιθέσεις.

ΚΕΦΑΛΑΙΟ 2: ΕΓΚΑΤΑΤΑΣΗ ΤΟΥ ΒΑΣΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ

Σε αυτό το κεφάλαιο θα αναφερθούμε διεξοδικά στο πώς μπορούμε να εγκαταστήσουμε την Java αλλά και το NetBeans 8.2 στον υπολογιστή μας. Θα μπορούσαμε να χρησιμοποιήσουμε το Eclipse IDE, αλλά προτιμήσαμε την παραπάνω πλατφόρμα.

ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA

Για να εγκαταστήσουμε την Java στον υπολογιστή μας ακολουθούμε τον παρακάτω σύνδεσμο :

<https://java.com/en/download/>

Στη συνέχεια μας εμφανίζεται η παρακάτω εικόνα(*Εικόνα : 3*)

Free Java Download

Download Java for your desktop computer now!

Version 8 Update 151

Release date October 17, 2017

Free Java Download

» [What is Java?](#) » [Do I have Java?](#) » [Need Help?](#)

Εικόνα 3: Η εγκατάσταση της Java[Retrieved from [Here](#)]

Μόλις κατεβάσουμε την εφαρμογή την ανοίγουμε και εμφανίζεται το παρακάτω(**Εικόνα: 4**)



Εικόνα 4: Η εγκατάσταση της Java (b)

Συμφωνούμε με τους όρους χρήσης και πατάμε συνεχώς next.

ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA SE DEVELOPMENT KIT 9

Στη συνέχεια κατεβάζουμε το JDK9 απο τον παρακάτω σύνδεσμο :

<http://www.oracle.com/technetwork/java/javase/downloads/jdk9-downloads-3848520.html>

Αφου ακολουθήσουμε τον παραπάνω σύνδεσμο βλέπουμε τις εξής επιλογές(**Εικόνα: 5**)

Java SE Development Kit 9 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

JDK 9.0.1 checksum

Java SE Development Kit 9.0.1

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux	304.99 MB	jdk-9.0.1_linux-x64_bin.rpm
Linux	338.11 MB	jdk-9.0.1_linux-x64_bin.tar.gz
macOS	382.11 MB	jdk-9.0.1_osx-x64_bin.dmg
Windows	375.51 MB	jdk-9.0.1_windows-x64_bin.exe
Solaris SPARC	206.85 MB	jdk-9.0.1_solaris-sparcv9_bin.tar.gz

Εικόνα 5: Εγκατάσταση της JAVA SE Kit 9.0.1 [Retrieved from [Here](#)]

Αφού κατεβάσουμε το αρχείο, μένει να το εγκαταστήσουμε πατώντας σε όλα οκ.

ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS 8.2 IDE

Για να κατεβάσουμε το NetBeans8.2 που έρχεται μαζί με το JDK ακολουθούμε τον παρακάτω σύνδεσμο :

<http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html>

Μολις μπουμε στη σελίδα βλέπουμε τις εξής επιλογές(Εικόνα: 6)

JDK 8u151 with NetBeans 8.2

This distribution of the JDK includes the Java SE bundle of [NetBeans IDE](#), which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

You must accept the [JDK 8u151 and NetBeans 8.2 Cobundle License Agreement](#) to download this software.

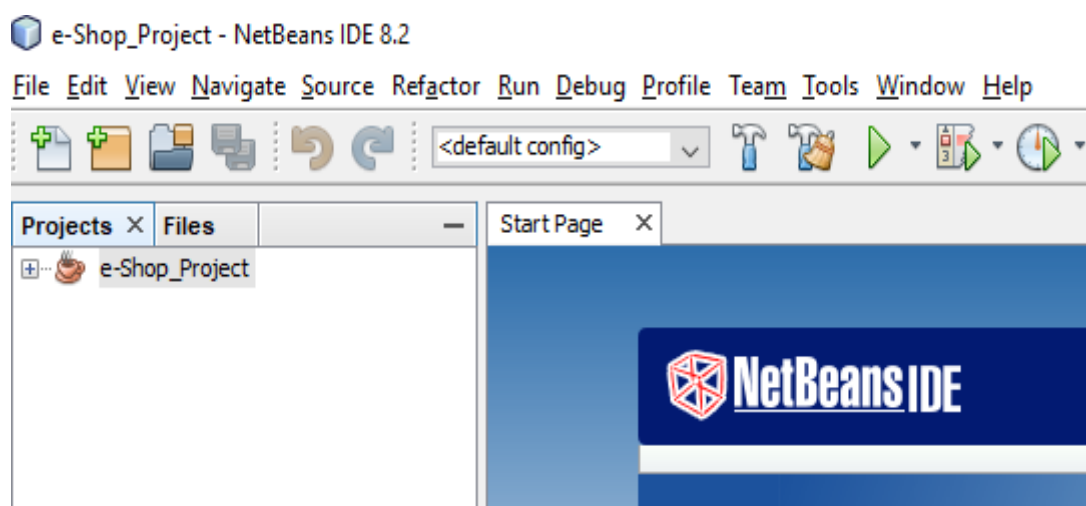
Accept License Agreement Decline License Agreement

Java SE and NetBeans Cobundle (JDK 8u151 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	295.39 MB	jdk-8u151-nb-8_2-linux-i586.sh
Linux x64	290.35 MB	jdk-8u151-nb-8_2-linux-x64.sh
Mac OS X x64	362.67 MB	jdk-8u151-nb-8_2-macosx-x64.dmg
Windows x86	326.03 MB	jdk-8u151-nb-8_2-windows-i586.exe
Windows x64	337.35 MB	jdk-8u151-nb-8_2-windows-x64.exe

- [License](#)

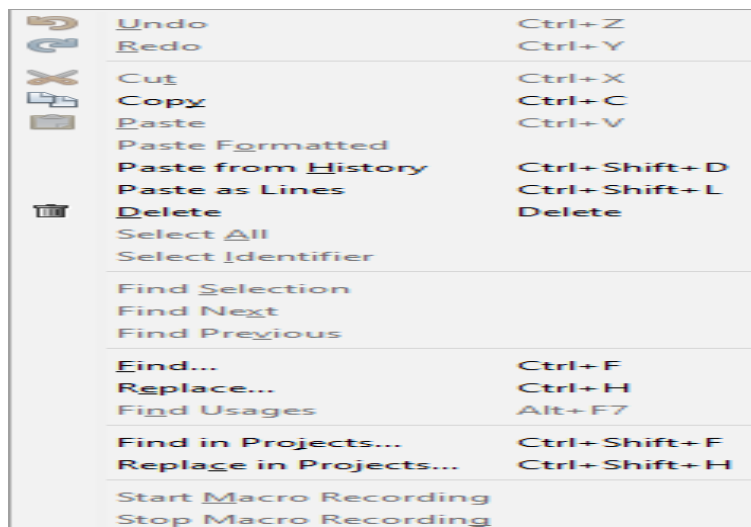
Εικόνα 6: Εγκατάσταση NetBeans8.2 και JDK [Retrieved from [Here](#)]

Επιλέγουμε την αρχιτεκτονική του συστήματος μας και στη συνέχεια κατεβάζουμε το αρχείο. Μόλις το κατεβάσουμε κάνουμε κανονική εγκατάσταση πατώντας σε όλα ok, αφού φυσικά δεχθούμε τους όρους αδείας που έρχονται με το λογισμικό. Ανοίγοντας το λογισμικό μας εμφανίζει το παρακάτω interface (*Εικόνα: 7*)



Εικόνα 7: Το interface του NetBeans IDE 8.2

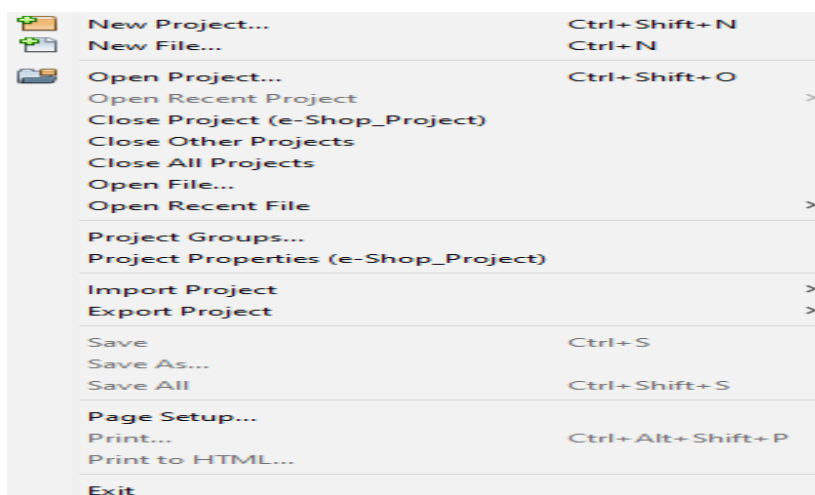
Εαν επιλέξουμε την επιλογή **Edit** βλέπουμε τις εξής επιλογές(**Εικόνα: 8**)



Εικόνα 8: Η επιλογή Edit του NetBeans

Μπορούμε δηλαδή να κάνουμε cut, copy, paste κομμάτια κώδικα στα αρχεία μας , find ή replace σε περίπτωση που κάνουμε αναζήτηση στον κώδικα. Τα κλασσικά undo, redo και delete.

Απο την άλλη πλευρά πατώντας στην επιλογή File(**Εικόνα: 9**) έχουμε τις επιλογές New Project, New File, Close Project, Close File και άλλες, όπως import και export.



Εικόνα 9: Η επιλογή File του NetBeans

Οι επιλογές **navigate**, **resource** είναι πιο εξειδικευμένες και χρειάζονται σε πρότζεκτ που είναι πολύ μεγάλα σε μέγεθος. Μας βοηθούν να επεξεργαζόμαστε ταχύτερα κάποια κομμάτια κώδικα.

Η επιλογή **refractor** μας δίνει τη δυνατότητα να μετονομάσουμε ένα αρχείο χωρίς να χαλάσουμε τη λογική του κώδικα που έχουμε δημιουργήσει ενώ η βασικότερη εντολή είναι η **run** η οποία τρέχει το πρόγραμμα μας και στη συνέχεια ανοίγει το **console** με σκοπό να μας παρουσιάσει λάθη που υπάρχουν.

Οι υπόλοιπες εντολές είναι και αυτές εξειδικευμένες πχ πως θα κάνουμε **debugging** σε ένα πρόγραμμα αλλά και πως θα κάνουμε εγκατάσταση ή αναβάθμιση των **plugin** μας, και με το **tools** μπορούμε να φτιάξουμε το **interface** της πλατφόρμας όπως εμάς μας αρέσει πχ γραμματοσειρές κτλ.

ΚΕΦΑΛΑΙΟ 3: Η ΓΛΩΣΣΑ XML

ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ XML

Η **γλώσσα XML** ουσιαστικά δεν είναι γλώσσα προγραμματισμού αλλά μια markup γλώσσα που ορίζει κανόνες για κωδικοποίηση των εγγράφων με σκοπό να είναι εύκολα αναγνώσιμα από τους ανθρώπους και τις μηχανές. Ιδιαίτερη έμφαση δίνεται στην απλότητα σύνταξης της (World Wide Web Consortium, 1996).

Στην παρούσα πτυχιακή μας ζητήθηκε να χρησιμοποιήσουμε τη γλώσσα XML ως μέσο αποθήκευσης των δεδομένων σε συνεργασία με τη γλώσσα προγραμματισμού Java που θα μας παρέχει τους μηχανισμούς λογικής και επέμβασης πάνω στα αρχεία αυτά.

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΤΗΣ ΓΛΩΣΣΑΣ XML

Αρχικά οποιοδήποτε χαρακτήρας που είναι Unicode μπορεί να εμφανιστεί σε ένα XML αρχείο. Η δόμηση των αρχείων XML βασίζεται στις **ετικέτες** (tags). Ένα tag πρέπει να ξεκινά και να κλείνει από μέσα προς τα έξω και να διαθέτει το ίδιο όνομα. Αρχίζει με τον χαρακτήρα «<» και τελειώνει με τον χαρακτήρα «>».

Επίσης διαθέτει στοιχεία (elements) τα οποία ξεκινούν με μια ετικέτα ανοίγματος και κλείνουν με μια ετικέτα κλεισίματος. Το περιεχόμενο που βρίσκεται ανάμεσα στις ετικέτες μπορεί να φανεί στον φυλλομετρητή μας (browser) αν επέμβουμε με τις κατάλληλες ενέργειες(World Wide Web Consortium, 1996).

Αν μια ετικέτα ανοίγματος και μια ετικέτα κλεισίματος περιέχει και άλλες παρόμοιες ετικέτες προς τα μέσα τότε ονομάζονται παιδιά αυτής της αρχικής ετικέτας ανοίγματος και κλεισίματος.

Στην παρακάτω εικόνα μπορούμε να δούμε ένα παράδειγμα εγγράφου XML(*Εικόνα: 10*)

```

1 <customers>
2   <customer>
3     <fname>Ioannis</fname>
4     <lname>Dimakis</lname>
5   </customer>
6   <customer>
7     </customer>
8     <fname>Euaggelos</fname>
9     <lname>Leontiadis</lname>
10 </customers>

```

Εικόνα 10: Παράδειγμα XML (1)

Επιπρόσθετα εκτός από στοιχεία διαθέτουμε και ιδιότητες που ονομάζονται attributes και χαρακτηρίζουν τα **elements**. Στην παρακάτω εικόνα βλέπουμε ότι προσθέσαμε στο παραπάνω παράδειγμα ιδιότητες στους πελάτες (*Εικόνα : 11*)

```

<customers>
  <customer>
    <fname id="0">Ioannis</fname>
    <lname>Dimakis</lname>
  </customer>
  <customer id="1">
    <fname>Euaggelos</fname>
    <lname>Leontiadis</lname>
  </customer>
</customers>

```

Εικόνα 11: Παράδειγμα XML (2)

Βλέπουμε ότι δώσαμε στον κάθε πελάτη ένα id με σκοπό να ξεχωρίζουν στο αρχείο. Επομένως με χρήση διάφορων γλωσσών προγραμματισμού μπορούμε να γράψουμε, να διαβάσουμε και να μορφοποιήσουμε τέτοιους είδους αρχεία.

ΠΕΡΙΟΡΙΣΜΟΙ ΣΤΑ XML ΑΡΧΕΙΑ (DTD's & XML Schemas)

Για να είμαστε σίγουροι ότι τα αρχεία γράφονται σωστά χωρίς λάθη πρέπει να εφαρμόσουμε κατάλληλους περιορισμούς που εμποδίζουν την λανθασμένη εγγραφή σε ένα τέτοιο σύστημα. Για να γίνει αυτό υπάρχουν δύο μέθοδοι (World Wide Web Consortium, 1996):

- **DTD's.** Είναι μια παλιά τεχνολογία εφαρμογής περιορισμών που χρησιμοποιείται μέχρι και σήμερα
- **XML Schema.** Νέα εχνολογία στην εφαρμογη περιορισμών στα XML αρχεία

Περιορισμοί με χρήση DTD's

Το DTD(Document Type Definition) ορίζει ποιά elements, element types, attributes lists κ.α επιτρέπονται σε ενα συγκεκριμένο αρχείο XML. Με αυτό τον τρόπο δε μπορούμε να βλάψουμε το αρχείο με λανθασμένη εισαγωγή δεδομένων. Υπάρχουν οι παρακάτω λέξεις κλειδιά πο είναι δεσμευμένες απο το σύστημα οι οποίες χρησιμοποιούνται για την κατασκευή DTD's αρχείων :

- **EMPTY**
- **ANY**
- **#PC DATA**
- **sequence list**
- **ATTLIST**
- **CDATA**
- **ID**
- **REQUIRED**
- **IMPLIED**
- **choice list**
- **IDREF** κ.α

Για την καλύτερη κατανόηση θα προσπαθήσουμε επιγραμματικά να αναλύσουμε το DTD με κατάλληλη χρήση παραδειγμάτων. Ας δούμε το παρακάτω DTD(*Εικόνα :12*)

```

<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Ioannis</to>
<from>Euaggelos</from>
<heading>Hello my friend</heading>
<body>Please buy a gift to our friend Nikos. Do no forget it again!</body>
</note>

```

Εικόνα 12: Παράδειγμα DTD (1)

Στο παραπάνω παράδειγμα βλέπουμε την προσπάθεια να στείλει ο Γιάννης ενα μήνυμα στον Βαγγέλη. Δεν θα ήτανε σωστό εαν ο Γιάννης είχε την ικανότητα στο πεδίο «to» να βάλει το πεδίο «body» γιατί έτσι δεν θα έφτανε ποτέ το μήνυμα στον Βαγγέλη. Οπως βλέπουμε στο πάνω- πάνω μέρος της δήλωσης του DTD αρχικά ορίζουμε το DOCTYPE note που ουσιαστικά είναι η ρίζα του XML (root) .

Κατόπιν ανάλογα με την μορφή του XML απο πάνω προς τα κατω ορίζουμε περιορισμούς. Σε αυτό το απλό παράδειγμα έχουμε την δεσμευμένη λέξη #PCDATA που αυτό σημαίνει οτι μας επιτρέπεται να βάλουμε μόνο τα δεδομένα που βρίσκονται μεταξύ μιας ετικέτας ανοίγματος και κλεισίματος.

Ειναι θεμιτό σε αυτό το σημείο να παρουσιάσουμε μερικούς ειδικούς **χαρακτήρες (entities)** που στην XML έχουν ειδική σημασία :

- **<**
- **>**
- **&**
- **"**
- **'**

Αν πάρουμε τη σειρά από πάνω προς τα κάτω αντιπροσωπεύουν τους παρακάτω χαρακτήρες:

- <
- >
- &
- "
- '

Για να κάνουμε ενσωμάτωση του DTD αρχείου στο XML αρχείο μας κάνουμε κάτι το παρακάτω μέσα ακριβώς μετά τη δήλωση του XML αρχείου(W3Schools, 2017):

```
<!DOCTYPE note SYSTEM "note.dtd">
```

Άλλες δεσμευμένες λέξεις όπως είναι το **!ELEMENT** ορίζει ένα στοιχείο στο XML (element). Όσο αφορά το δεύτερο element του παραδείγματος ουσιαστικά παίρνει τρεις παραμέτρους καθώς είναι το element (root) που εμπεριέχει τα άλλα τρία elements. Έχουμε πεί ότι ένα element μπορεί να περιέχει και άλλο element (παιδιά)

Εκτός από elements μπορούμε να ορίσουμε και ιδιότητες (attributes). Εάν για παράδειγμα διαθέτουμε το παρακάτω παράδειγμα XML:

```
< name type ="lastname" />
```

Μπορούμε να ορίσουμε το attribute του XML element σε DTD ως εξής :

```
< !ATTLIST name type (lastname) >
```

Πρωτού πάμε στο τελευταίο παράδειγμα μας πρέπει να ορίσουμε κάποια **ειδικά σύμβολα** που μας δείχνουν εάν ένα element περιέχει μηδενικά ή πολλά ή ένα στοιχείο στο XML αρχείο(**quantifiers**) :

- +. Υπάρχει ένα ή περισσότερα αντικείμενα ενός συγκεκριμένου element
- *. Μηδέν ή ένα ακριβώς αντικείμενο υπάρχει σε συγκεκριμένο element
- ?. Εδώ εννοούμε ότι τα αντικείμενα που εμφανίζονται στο XML αρχείου και είναι συγκεκριμένου τύπου, είναι ακριβώς ένα.
- **No quantifier.**

Έστω ότι διαθέτουμε έναν κατάλογο προϊόντων και χρειαζόμαστε να τον γράψουμε σε XML αλλά να δημιουργήσουμε και ένα κατάλληλο DTD(*Εικόνα : 13*)

```
<!DOCTYPE CATALOG [  
  
  <!ENTITY AUTHOR "John Doe">  
  <!ENTITY COMPANY "JD Power Tools, Inc.">  
  <!ENTITY EMAIL "jd@jd-tools.com">  
  
  <!ELEMENT CATALOG (PRODUCT+)>  
  
  <!ELEMENT PRODUCT  
    (SPECIFICATIONS+,OPTIONS?,PRICE+,NOTES?)>  
  <!ATTLIST PRODUCT  
    NAME CDATA #IMPLIED  
    CATEGORY (HandTool|Table|Shop-Professional) "HandTool"  
    PARTNUM CDATA #IMPLIED  
    PLANT (Pittsburgh|Milwaukee|Chicago) "Chicago"  
    INVENTORY (InStock|Backordered|Discontinued) "InStock">  
  
  <!ELEMENT SPECIFICATIONS (#PCDATA)>  
  <!ATTLIST SPECIFICATIONS  
    WEIGHT CDATA #IMPLIED  
    POWER CDATA #IMPLIED>
```

Εικόνα 13: Παράδειγμα DTD'S (2a) [Retrieved from [Here](#)]

Παρατηρούμε ότι ορίζουμε τρεις οντότητες. Έναν συγγραφέα, μια εταιρεία και το email αυτής. Όμως ο κατάλογος διαθέτει προϊόντα επομένως στο element CATALOG θα βάλουμε το element PRODUCT.

Θα προσθέσουμε + σαν quantifier γιατί υπάρχουν περισσότερα του ενός προϊόντα στο κατάστημα. Παρομοίως ορίζεται και το element PRODUCT το οποίο με τη σειρά του διαθέτει τεσσερα παιδιά. Κάθε παιδί έχει τον δικό του quantifier. Με παρόμοιο τρόπο ορίζουμε και το **!ATTLIST**.

Υπάρχει φυσικά και νεότερος τρόπος να ορίσουμε περιορισμούς για τα αρχεία XML που ονομάζεται **XML Schema**. Στην δική μας εργασία χρησιμοποιήσαμε DTD αντί του XML Σχήματος. Είναι μια πιο σύγχρονη τεχνολογία σε σχέση με το DTD. Θα δώσουμε το παρακάτω παράδειγμα(*Εικόνα: 14*)

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="https://www.w3schools.com"
xmlns="https://www.w3schools.com"
elementFormDefault="qualified">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>

```

Εικόνα 14: Παράδειγμα XML Schema [Retrieved from [Here](#)]

Παραπάνω βλέπουμε ότι είναι πολύ πιο εύκολο να ορίσουμε ένα XML Schema παρά ένα DTD αρχείο. Οπως φαίνεται δηλώνουμε το element «**to**» να είναι string και να είναι σε μια ακολουθία με τα άλλα elements ενόσω όλα αυτά μαζί βρίσκονται μέσα στη ρίζα που ονομάζεται note.

Ένα XML Schema **υποστηρίζει** τις παρακάτω μεταβλητές :

- **xs:string**
- **xs:decimal**
- **xs:integer**
- **xs:boolean**
- **xs:date**
- **xs:time**

Εξηγήσαμε στο παραάνω παράδειγμα την λειτουργία ενός element. Δηλαδή ενός απλού element. Στο XML Schema τα elements χωρίζονται σε **Simple** και **Complex types**. Ένα Complex Type element εσωκλείει και άλλα elements. Ενώ ένα Simple element όχι. Ένα παράδειγμα χρήσης Complex Type φαίνεται παρακάτω(*Εικόνα : 15*)


```

<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Εικόνα 15: Παράδειγμα με Complex Type [Retrieved from [Here](#)]

ΑΝΑΖΗΤΗΣΗ ΚΑΙ ΜΟΡΦΟΠΟΙΗΣΗ(ΧΡΑΤΗ – XSLT)

Για να μπορέσουμε να κάνουμε μορφοποίηση σε ένα XML αρχείο ώστε να φαίνεται όπως ακριβώς εμείς θέλουμε χρησιμοποιούμε XSLT ενώ για να κάνουμε αναζήτηση χρησιμοποιούμε ΧΡΑΤΗ. Επειδή δεν μας ζητήθηκε κάτι ανάλογο, θα περιγράψουμε επιγραμματικά τις λειτουργίες αυτών των δύο(W3schools, 2017).

Όσο αφορά την αναζήτηση έστω οτι μας δίνεται το παρακάτω XML αρχείο(*Εικόνα: 16*)

```

<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>

```

Εικόνα 16: Αναζήτηση σε XML αρχείο [Retrieved from [Here](#)]

Τότε για να βρούμε για παράδειγμα για να βρούμε το πρώτο βιβλίο εφαρμόζουμε το παρακάτω XPATH:

/bookstore/book[1]

Όπως βλέπουμε ψάχνουμε σε όλο το directory του bookstore, για να βρούμε το πρώτο βιβλίο. Επίσης για παράδειγμα εάν θέλουμε να βρούμε τα βιβλία που έχουν τιμή μεγαλύτερη απο 35 τότε :

/bookstore/book[price>35.00]/title

Δηλαδή να ψάξουμε στο bookstore να βρούμε βιβλία τα οποία έχουν τιμή μεγαλύτερη απο 35 και να επιστραφεί ο τίτλος τους. Δεν θα επεκταθούμε όμως περισσότερο.

Όσο αφορά την μορφοποίηση ας δούμε το παρακάτω παράδειγμα(*Εικόνα: 17*)

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="artist"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
```

Εικόνα 17: Μορφοποίηση XML με χρήση XSLT [Retrieved from [Here](#)]

Βλέπουμε οτι συνδυάζουμε την γλώσσα HTML και διάφορες εντολές της XSLT για να παρουσιάσουμε ένα XML αρχείο. Χρησιμοποιούμε τις εντολές for-each και select για να επιλέξουμε elements απο το XML αρχείο(θα μπορούσαμε να εφαρμόσουμε παράλληλα και XPATH μαζί με το XSLT). Ισχυουν όλοι οι τύποι, κανονες της HTML κανονικά.

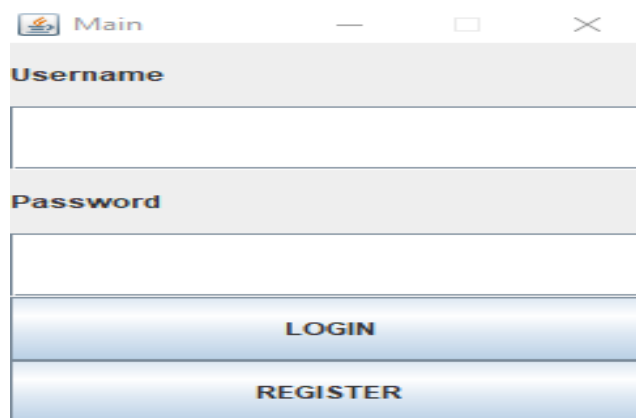
ΚΕΦΑΛΑΙΟ 4: ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Στο παρόν κεφάλαιο θα παρουσιάσουμε βήμα βήμα και θα τρέξουμε τον κώδικα τον οποίο δημιουργήσαμε που αφορά την κατασκευή ενός ηλεκτρονικού καταστήματος με χρήσης της γλώσσας προγραμματισμού Java και XML. Θα προσπαθήσουμε όσο το δυνατόν πιο αναλυτικά, χωρίς να κουράζει να εξηγήσουμε τον κώδικα που δημιουργήσαμε.

ΟΙ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ΠΕΛΑΤΗ

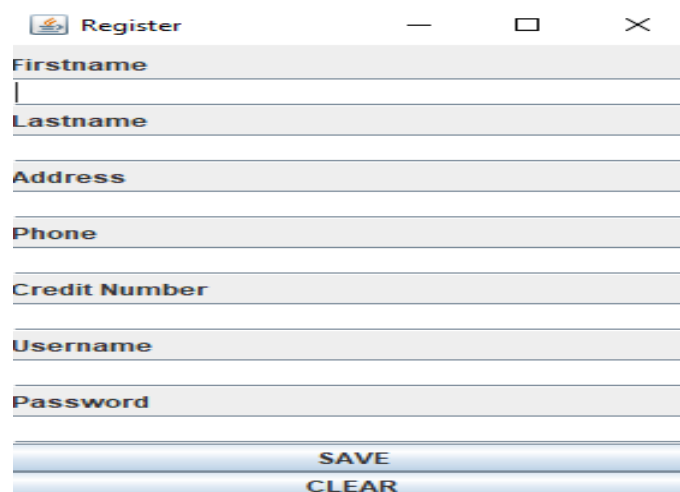
ΕΓΓΡΑΦΗ ΠΕΛΑΤΗ ΣΤΟ ΣΥΣΤΗΜΑ

Μπορούμε να κάνουμε εγγραφή ενός πελάτη στο σύστημα. Αρχικά βλέπουμε το JFrame που δημιουργείται απο το τρέξιμο της εφαρμογής μας(**Εικόνα: 18**).



Εικόνα 18: Εγγραφή πελάτη στο σύστημα

Βλέπουμε ένα JFrame το οποίο διαθέτει δύο JLabels, δύο JTextFields και δύο buttons. Κάθε button δημιουργεί ένα ActionListener το οποίο μέσω ενός event (πάτημα κουμπιού κάνει μια ενέργεια). Στη προκειμένη περίπτωση πατάμε το κουμπί register(**Εικόνα : 19**)



Εικόνα 19: Εγγραφή πελάτη στο σύστημα(2)

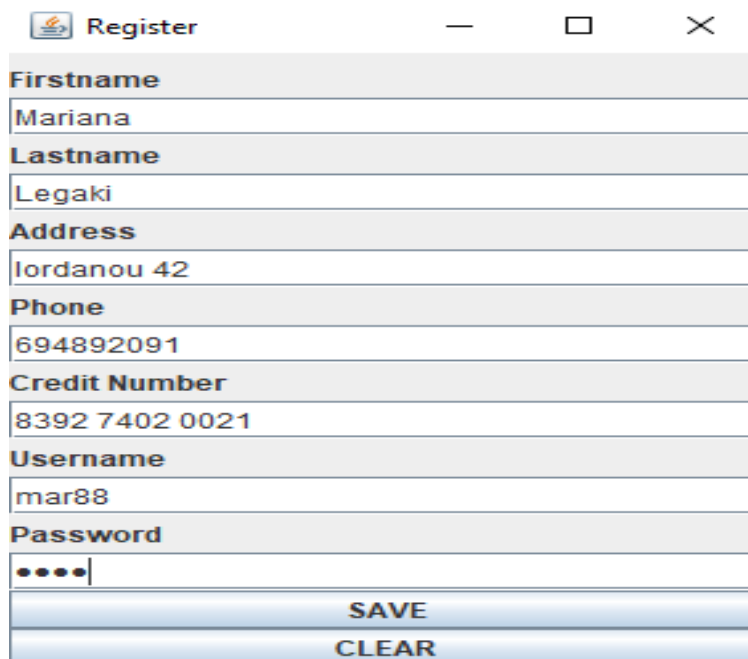
Αν τοποθετήσουμε στοιχεία στη φόρμα που δημιουργήθηκε και κάνουμε κλικ στο κουμπί save αυτος ο πελάτης θα αποθηκευτεί στο XML αρχείο customers_XML.xml ενώ αν πατήσουμε clear καθαρίζει η φόρμα μας απο τα δοσμένα στοιχεία του νέου πελάτη.

Αρχικά, επειδή έχουμε τρέξει μερικές φορές την εφαρμογή μας, μπορούμε να δούμε μερικούς πελάτες στο XML αρχείο(**Εικόνα : 20**)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><customers>
<customer><fname>Eleni</fname><lname>Mixailidou</lname><address>Maxis Kritis78</address><phone>6947826748</phone><credit_number>
<customer><fname>Ioannis</fname><lname>Nousos</lname><address>Athitaki 12</address><phone>6938728930</phone><credit_number>7832
<customer><fname>Theodoros</fname><lname>Kalimanidis</lname><address>Makariou 51</address><phone>6987658902</phone><credit_numb
<customer><fname>Anastasia</fname><lname>Kalidou</lname><address>Gortunias 82</address><phone>6948729091</phone><credit_number>
```

Εικόνα 20: Το XML αρχείο customers_XML.xml

Εαν δώσουμε για παράδειγμα προσπαθήσουμε να γράψουμε εναν πελάτη(**Εικόνα: 21**)



Firstname
Mariana
Lastname
Legaki
Address
Iordanou 42
Phone
694892091
Credit Number
8392 7402 0021
Username
mar88
Password
••••
SAVE
CLEAR

Εικόνα 21: Εγγραφή ενός πελάτη

Επίσης έχουμε βάλει περιορισμό ώστε οταν γίνεται εγγραφή να μη γίνεται να δωθεί το ίδιο όνομα χρήστη ή TIN(δηλαδή αριθμό φορολογικού μητρώου). Δεν θέλουμε να υπάρχουν

διπλότυπα usernames ή αριθμοί φορολογικού μητρώου. Μολις πατήσουμε save αμέσως γίνεται update το XML file(*Εικόνα: 22*)

Επειδή χρησιμοποιούμε API το οποίο γράφει το XML αρχείο σε μία σειρά αν και διαθέτε κανονική μορφοποίηση (root, elements, children of elements) για δική μας ευκολία και παρουσίαση τραβάμε τη γραμμή κάτω απο το XML για να δούμε τη νέα εγγραφή. Φυσικά δεν είναι απαραίτητο αυτό.

Όταν επεξεργαζόμαστε XML αρχεία, το σύστημα δε νοιάζεται αν γράφονται σε μία σειρά ή σε κανονική «μορφοποιημένη» μορφή.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><customers>
<customer><fname>Eleni</fname><lname>Mixailidou</lname><address>Maxis Kritis78</address><phone>6947826748</phone><credit_number>
<customer><fname>Ioannis</fname><lname>Nousos</lname><address>Athitaki 12</address><phone>6938728930</phone><credit_number>7832
<customer><fname>Theodoros</fname><lname>Kalimanidis</lname><address>Makariou 51</address><phone>6987658902</phone><credit_num
<customer><fname>Anastasia</fname><lname>Kalidou</lname><address>Gortunias 82</address><phone>6948729091</phone><credit_number>
<customer><fname>Mariana</fname><lname>Legaki</lname><address>Iordanou 42</address><phone>694892091</phone><credit_number>8392
```

Εικόνα 22: Εγγραφή του χρήστη στο σύστημα – XML

ΕΙΣΟΔΟΣ ΠΕΛΑΤΗ ΣΤΟ ΣΥΣΤΗΜΑ

Για να ξεχωρίσουμε την είσοδο ενός πελάτη στο σύστημα απο έναν admin(μας ζητήθηκε να περιέχει έναν και μονο admin). Χρησιμοποιούμε ActionListener για να ανοίξει το XML αρχείο, να κάνει αναζήτηση και στη συνέχεια να ψάξει για elements τύπου username και τιμές αυτών σαν αυτήν που δώσαμε.

Αν δώσουμε μια τιμή username που υπάρχει στο σύστημα αυτόματα ανοίγει ένα κεντρικό JFrame με τις επιλογές του πελάτη. Με παρόμοια λογική λειτουργεί και η είσοδος του admin(*Εικόνα: 23*)

```

try {
    File file = new File("customers_XML.xml");
    // Create builders
    DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
    Document doc = dBuilder.parse(file);
    doc.getDocumentElement().normalize();
    //XML/JAVA has problem with multiple for's , if's so we split them
    NodeList nList = doc.getElementsByTagName("customer");
    for (int i = 0; i < nList.getLength(); i++) {
        NodeList childList = nList.item(i).getChildNodes();
        Node childNode = childList.item(5);
        Node childNode1 = childList.item(6);
        //check if the user is customer
        if (childNode.getTextContent().equals(username) && childNode1.getTextContent().equals(password)) {
            //customer's world
            JFrame customer_frame = new JFrame();
            customer_frame.setSize(300, 400);
            customer_frame.setTitle("Customer Panel");
            customer_frame.setLayout(new BorderLayout());
            JPanel customer_panel = new JPanel();
            customer_frame.add(customer_panel);
            customer_panel.setLayout(new GridLayout(4, 1));
            JButton print_products = new JButton("Print Products");
            JButton delete_order = new JButton("Delete order");
            JButton shopping_basket = new JButton("Shopping basket");
            JButton logout_customer = new JButton("Logout");
            customer_panel.add(print_products);
            customer_panel.add(delete_order);
            customer_panel.add(shopping_basket);
            customer_panel.add(logout_customer);
        }
    }
}

```

Εικόνα 23: Είσοδος του πελάτη στο σύστημα (1)

Ενώ όταν εισέλθει στο σύστημα χρησιμοποιούνται οι παρακάτω actionListeners που μας μεταβαίνουν σε άλλες κλάσεις που υλοποιούν με παρόμοια λογική τα κατάλληλα XML αρχεία(Εικόνα: 24)

```

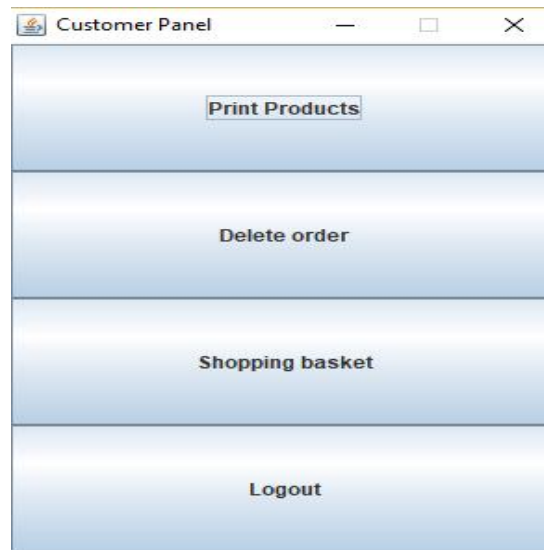
//actionListeners for customer choices
print_products.addActionListener(new PrintProducts());
delete_order.addActionListener(new ActionListener() {...15 lines });
shopping_basket.addActionListener(new ShoppingBasket());
logout_customer.addActionListener(new ActionListener() {...10 lines });

```

Εικόνα 24: Είσοδος πελάτη στο σύστημα (2)

Ο πελάτης μπορεί να καλέσει τις παρακάτω κλάσεις χρησιμοποιώντας το παρακάτω JFrame μόλις κάνουμε είσοδο στο σύστημα ως πελάτες. Το κουμπί logout_customer απλά κάνει

dispose το JFrame για να βγούμε απο το σύστημα και να συνδεθούμε σαν admin(**Εικόνα : 25**)

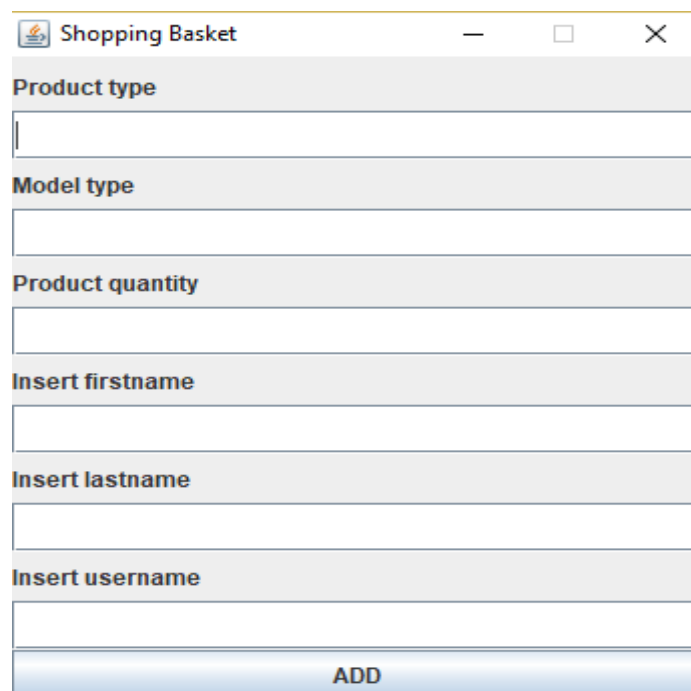


Εικόνα 25: Το κεντρικό JFrame του πελάτη

Εφόσον έχουμε κάνει είσοδο, τώρα μένει να δούμε τι θα συμβεί όταν επιλέξουμε κάθε φορά μια απο τις παραπάνω λειτουργίες.

ΠΡΟΣΘΗΚΗ ΣΤΟ ΚΑΛΑΘΙ ΠΡΟΙΟΝΤΩΝ

Μολις πατήσουμε το κουμπί μας εμφανίζεται το παρακάτω JFrame(**Εικόνα : 26**)

A screenshot of a Java Swing window titled "Shopping Basket". The window has a standard title bar. The main content area is a light gray gradient with several input fields and a button. The fields are labeled: "Product type", "Model type", "Product quantity", "Insert firstname", "Insert lastname", and "Insert username". At the bottom, there is a blue button labeled "ADD".

Εικόνα 26: Προσθήκη προϊόντων στο καλάθι αγορών

Επειδή σε κάθε XML αρχείο μπορούμε να γράφουμε προϊόντα με συγκεκριμένη μορφή, οπότε θα ήτανε αρκετά δύσκολο να γράψουμε laptops, televisions και mobiles σε ένα XML αρχείο. Οπότε τα διαχωρίσαμε. Όμως με την κατάλληλη λογική όλα αυτά τα τρία XML αρχεία θεωρούνται ως μια παραγγελία (admin logic). Να τονίσουμε, ότι ζητάμε το username ώστε να γνωρίζει ο admin με την κατάλληλη λογική το ποιος έκανε την παραγγελία.

Για να μπορέσουμε να βάλουμε προϊόντα στο καλάθι, τα προϊόντα πρέπει ήδη να υπάρχουν στο σύστημα. Θα πρέπει ήδη ο admin να έχει προσθέσει laptops, mobiles και televisions.

Έστω ότι έχει βάλει ήδη. Ας δούμε τα laptops που ανήκουν στο αρχείο **unfinished_order_laptops_XML.xml**(**Εικόνα: 27**)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><laptops>
<laptop><laptop_model>test_model</laptop_model><laptop_screenSize>test_scrSize</laptop_screenSize><laptop_GPU>test_gpu</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
</laptops>
```

Εικόνα 27: Το αρχείο unfinished_order_laptops_XML.xml

Όπως βλέπουμε ο admin ήδη έχει προσθέσει προϊόντα στο σύστημα. Ας προσπαθήσουμε να βάλουμε ένα προϊόν στο καλάθι (**Εικόνα: 28**). Διαθέταμε έξι προϊόντα τύπου LENOVO G750 ενώ με την προσθήκη στο καλάθι διαθέτουμε οκτώ.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><laptops>
<laptop><laptop_model>test_model</laptop_model><laptop_screenSize>test_scrSize</laptop_screenSize><laptop_GPU>test_gpu</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
</laptops>
```

Εικόνα 28: Update του αρχείου unfinished_order_laptops_XML.xml

Όσο αφορά τον κώδικα μέσω actionListener προσπαθούμε να ανοίξουμε το αρχείο laptops_XML.xml να βρούμε εάν το μοντέλο του προϊόντος που δώσαμε υπάρχει εκεί και στη συνέχεια με κατάλληλο τρόπο να το γράψουμε στο αρχείο unfinished_order_XML.xml. Η διαφορά των αρχείων είναι ότι το αρχείο laptops_XML.xml δεν περιέχει τα πεδία fname,

lname, username που έχει το άλλο XML αρχείο. Παρακάτω βλέπουμε οτι κάνουμε append τα πεδία που χρειαζόμαστε(*Εικόνα: 29*)

```
Element laptop_company_name_v = doc.createElement("company_name");
laptop_company_name_v.appendChild(doc.createTextNode(laptop_product_companyName_to_Add));
name_v.appendChild(laptop_company_name_v);

Element customer_laptops_buy_fname = doc.createElement("customer_fname");
customer_laptops_buy_fname.appendChild(doc.createTextNode(customer_get_firstname));
name_v.appendChild(customer_laptops_buy_fname);

Element customer_laptops_buy_lname = doc.createElement("customer_lname");
customer_laptops_buy_lname.appendChild(doc.createTextNode(customer_get_lastname));
name_v.appendChild(customer_laptops_buy_lname);

Element customer_laptops_buy_Username = doc.createElement("customer_username");
customer_laptops_buy_Username.appendChild(doc.createTextNode(customer_username));
name_v.appendChild(customer_laptops_buy_Username);
```

Εικόνα 29: Κανοντας append τρία νέα πεδία

Αμα προσπαθούσαμε να το πάρουμε απο την αρχή αρχικά (*Εικόνα : 30*) ανοίγουμε τα αρχεία laptops_XML.xml, mobiles_XML.xml, televisions_XML.xml.

```
//get ActionListener values
String get_product_type = text1_pr_product_type.getText();
String get_model_type = text2_pr_model_type.getText();
String get_product_quantity = text3_pr_quantity.getText();
//we want to convert the quantity given from string to int in order to add
//the same amount of products to the unfinished_ordr_laptops_XML.xml file
int given_quantity_int = Integer.parseInt(get_product_quantity);
//builders for the products files
File file1 = new File("laptops_XML.xml");
File file2 = new File("mobiles_XML.xml");
File file3 = new File("televisions_XML.xml");

// Create builders for each XML file
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc1 = dBuilder.parse(file1);
Document doc2 = dBuilder.parse(file2);
Document doc3 = dBuilder.parse(file3);
doc1.getDocumentElement().normalize();
doc2.getDocumentElement().normalize();
doc3.getDocumentElement().normalize();
/* create the appropriate XML files for each kind of product which customer has chose the files are
1) unfinished_order_laptops_XML.xml
2) unfinished_order_mobiles_XML.xml
3) unfinished_order_televisions_XML.xml
In order to append same products, using XML we need to have at least one product
in our file. So we will create a "test" product but when we need to extract the contents
of this file, we can ignore it simply.
_,
```

Εικόνα 30: Ανοίγουμε τα unfinished_order αρχεία

Ανοίγουμε και τα τρία αρχεία διότι ο πελάτης μπορεί να βάλει και σαν επιλογή mobiles ή televisions. Κατόπιν (*Εικόνα: 31*)

```

NodeList nList1 = doc1.getElementsByTagName("laptop");
for (int i = 0; i < nList1.getLength(); i++) {
    NodeList childList = nList1.item(i).getChildNodes();
    Node childNode0 = childList.item(0);
    Node childNode1 = childList.item(1);
    Node childNode2 = childList.item(2);
    Node childNode3 = childList.item(3);
    Node childNode4 = childList.item(4);
    Node childNode5 = childList.item(5);
    Node childNode6 = childList.item(6);
    Node childNode7 = childList.item(7);
    Node childNode8 = childList.item(8);

    if (childNode0.getTextContent().equals(get_model_type)) {
        /*this model type laptop, exists in our file so we must add the same model to :
        unfinished_order_laptops_XML.xml file (as shopping basket's product...)
        */
        String laptop_product_model_to_Add = childNode0.getTextContent();
        String laptop_product_ScreenSize_to_Add = childNode1.getTextContent();
        String laptop_product_GPU_to_Add = childNode2.getTextContent();
        String laptop_product_CPU_to_Add = childNode3.getTextContent();
        String laptop_product_RAM_to_Add = childNode4.getTextContent();
        String laptop_product_cost_to_Add = childNode5.getTextContent();
        String laptop_product_warranty_to_Add = childNode6.getTextContent();
        String laptop_product_substoreName_to_Add = childNode7.getTextContent();
        String laptop_product_companyName_to_Add = childNode8.getTextContent();
        /*-----*/
    }
}

```

Εικόνα 31: Παίρνουμε τα elements των αρχείων unfinished_order

Στη συνέχεια (*Εικόνα: 32*), βάζουμε μια for που μας βοηθά να ξέρουμε πόσες φορές θα γράψουμε αυτό το προϊόν στο καλάθι, καθώς μπορεί να θέλουμε για παράδειγμα 2 laptop και όχι ένα.

```

for (int count = 0; count < given_quantity_int; count++) {
    File file = new File("unfinished_order_laptops_XML.xml");

    Document doc = dBuilder.parse(file);
    doc.getDocumentElement().normalize();

    Element laptop_v = doc.createElement("laptop");

    Element lp_model = doc.createElement("laptop_model");
    lp_model.appendChild(doc.createTextNode(laptop_product_model_to_Add));
    laptop_v.appendChild(lp_model);

    Element lp_ScreenSize = doc.createElement("laptop_screenSize");
    lp_ScreenSize.appendChild(doc.createTextNode(laptop_product_ScreenSize_to_Add));
    laptop_v.appendChild(lp_ScreenSize);

    Element lp_GPU = doc.createElement("laptop_GPU");
    lp_GPU.appendChild(doc.createTextNode(laptop_product_GPU_to_Add));
    laptop_v.appendChild(lp_GPU);

    Element lp_CPU = doc.createElement("laptop_CPU");
    lp_CPU.appendChild(doc.createTextNode(laptop_product_CPU_to_Add));

    Element lp_RAM = doc.createElement("laptop_RAM");
    lp_RAM.appendChild(doc.createTextNode(laptop_product_RAM_to_Add));
    laptop_v.appendChild(lp_RAM);

    Element lp_cost = doc.createElement("laptop_cost");
    lp_cost.appendChild(doc.createTextNode(laptop_product_cost_to_Add));
    laptop_v.appendChild(lp_cost);

    Element lp_warranty = doc.createElement("laptop_warranty");
    lp_warranty.appendChild(doc.createTextNode(laptop_product_warranty_to_Add));
    laptop_v.appendChild(lp_warranty);

    Element lp_company_name = doc.createElement("substore_name");
    lp_company_name.appendChild(doc.createTextNode(laptop_product_substoreName_to_Add));
    laptop_v.appendChild(lp_company_name);

    Element laptop_company_name_v = doc.createElement("company_name");
    laptop_company_name_v.appendChild(doc.createTextNode(laptop_product_companyName_to_Add));
    laptop_v.appendChild(laptop_company_name_v);

    Element customer_laptops_buy_fname = doc.createElement("customer_fname");
    customer_laptops_buy_fname.appendChild(doc.createTextNode(customer_get_firstname));
    laptop_v.appendChild(customer_laptops_buy_fname);
}

```

Εικόνα 32: Κάνουμε appending στα αρχεία unfinished_order

Τώρα απλά μένει να γράψουμε στο αρχείο(**Εικόνα: 33**) τα δεδομένα μας. Η ίδια λογική ακριβώς ακολουθείται και για τα υπόλοιπα **unfinished_order** αρχεία στην κλάση **ShoppingBasket()**

```
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(new File("unfinished_order_laptops_XML.xml"));
transformer.transform(source, result);
```

Εικόνα 33: Πώς γράφουμε σε ένα αρχείο XML με χρήση DOM API

ΔΙΑΓΡΑΦΗ ΠΑΡΑΓΓΕΛΙΑΣ

Για να διαγράψουμε μια παραγγελιά πρώτου θεωρηθεί ως ολοκληρωμένη απο τον admin είναι να σβήσουμε προϊόντα απο τα unfinished_order αρχεία. Ομως, πρέπει να σβήσουμε με βάση το username που δώσαμε οταν συνδεθήκαμε στο σύστημα. Για αυτό τον λόγο χρησιμοποιήσαμε τα αρχεία αυτα ώστε να διαθέτουν τρία έξτρα πεδία σε σχέση με τα αρχεία laptops_XML.xml κ.α .

Με το που κάνουμε delete order το σύστημα ψάχνει και στα τρία αρχεία , ανοίγωντας τα πρώτα και στη συνέχεια ψάχνει να βρεί προϊόντα τύπου laptops ,mobiles ή televisions που να έχουν ως πεδίο το συγκεκριμένο username. Μολις τα σβήσει χρησιμοποιείται και μια ειδική συνάρτηση που καθαρίζει απο «κενούς κόμβους» το αρχείο.

Ας πάμε στον κώδικα(**Εικόνα : 34**).

```
delete_order.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        try {
            //If we want to delete an order we must pass the username in function then open
            //the appropriate files for deleting
            deleteOrderByUsernameGivenLaptops(username);
            deleteOrderByUsernameGivenMobiles(username);
            deleteOrderByUsernameGivenTelevisions(username);
        } catch (ParserConfigurationException | SAXException | IOException | TransformerException ex) {
            Logger.getLogger(EShop_Project.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
});
```

Εικόνα 34: delete_order button

Όταν γίνει το event του actionListener στο κουμπί delete_order καλούνται τρεις συναρτήσεις κάθε μια εκ των οποίων σβήνει και ένα διαφορετικό τύπο προϊόντων αλλά παίρνει ως όρισμα το username του πελάτη. Για συντομία, θα δείξουμε το πώς για παράδειγμα σβήνει η συνάρτηση **deleteOrderByGivenUsernameMobiles** με όρισμα το username(*Εικόνα: 35*)

```
private static void deleteOrderByUsernameGivenMobiles(String username) throws ParserConfigurationException
//delete mobiles from order
File file = new File("unfinished_order_mobiles_XML.xml");
// Create builders
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(file);
doc.getDocumentElement().normalize();

//function for removing products from XML file
NodeList nList = doc.getElementsByTagName("mobile");
for (int i = 0; i < nList.getLength(); i++) {
    NodeList childList = nList.item(i).getChildNodes();
    Node childNode0 = childList.item(0);
    Node childNode1 = childList.item(1);
    Node childNode2 = childList.item(2);
    Node childNode3 = childList.item(3);
    Node childNode4 = childList.item(4);
    Node childNode5 = childList.item(5);
    Node childNode6 = childList.item(6);
    Node childNode7 = childList.item(7);
    Node childNode8 = childList.item(8);
    Node childNode9 = childList.item(9);
    Node childNode10 = childList.item(10);
    Node childNode11 = childList.item(11);
    Node childNode12 = childList.item(12);

    if (childNode12.getTextContent().equals(username)) {
        childNode0.getParentNode().removeChild(childNode0);
        childNode1.getParentNode().removeChild(childNode1);
        childNode2.getParentNode().removeChild(childNode2);
        childNode3.getParentNode().removeChild(childNode3);
        childNode4.getParentNode().removeChild(childNode4);
    }
}
```

Εικόνα 35: Διαγραφή mobiles απο το καλάθι (1)

Βλέπουμε οτι ανοίγει το αρχείο **unfinished_order_mobiles_XML.xml** και πάει να βρεί προϊόντα τύπου mobile σύμφωνα με το username που δώσαμε. Τότε καλούνται οι **getParentNode()** και **removeChild()** για να σβήσουμε τα παιδιά του element που έχει όνομα ίδιο με αυτό που δώσαμε.

Καλείται οπως πριν αναφέραμε και η **removeEmptyNodes** για να σβήσουμε τους κενούς κόμβους που απέμειναν μετά τη διαγραφή(Έμεινε ουσιαστικά ενα κενό element. Αυτό φαινεται στην παρακάτω εικόνα (*Εικόνα: 36*)

```

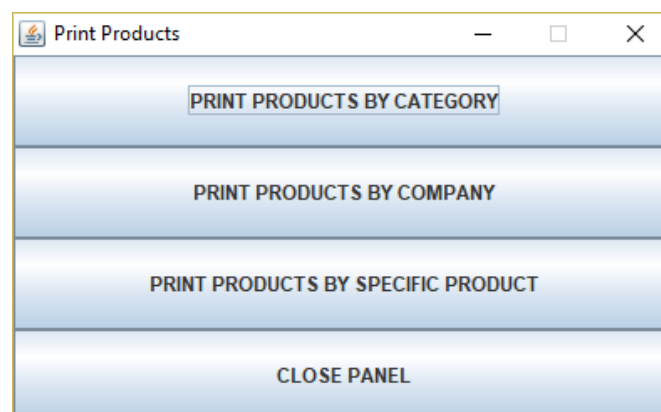
//remove a product using a function
String mobile = "mobile";
removeEmptyNodes(doc, mobile);
// write the content into xml file
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(new File("unfinished_order_mobiles_XML.xml"));
transformer.transform(source, result);

```

Εικόνα 36: Κλήση της συνάρτησης removeEmptyNodes και update του XML

ΕΚΤΥΠΩΣΗ ΠΡΟΙΟΝΤΩΝ

Σε αυτή τη λειτουργία προσπαθούμε να εκτυπώσουμε τα προϊόντα. Δεν ζητήθηκε αλλά το κρίναμε απαραίτητο για να μπορεί ένας πελάτης να ψωνίσει προϊόντα. Δηλαδή, υλοποιεί έναν κατάλογο προϊόντων. Με το που κάνουμε κλικ στο κουμπί print products μας ανοίγει ένα νέο JFrame (*Εικόνα: 37*)



Εικόνα 37: JFrame για εκτύπωση προϊόντων

Αρχικά ανοίγουμε τα αρχεία, έπειτα χρησιμοποιούμε for statements και JTextArea και JScrollPane για να δούμε τα προϊόντα μας. Εάν για παράδειγμα πατήσουμε την επιλογή print products by category, τότε μας ζητείται να βάλουμε την κατηγορία.

Αν θέσουμε televisions(θα μπορούσαμε να βάλουμε και mobiles ή laptops) . Ένα μικρό μέρος του JTextArea φαίνεται παρακατω (*Εικόνα: 38*)

Model	Size	Color	TV type	Internet	Cost	Sub/ire	Company
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Public Alexadroupoli	Company : KOTSOVOLOS
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Kotsovolos Rethymno	Company : PUBLIC
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Germanos Rhodes	Company : GERMANOS
Model : LG 60UH605V	Size : 62 Inches	Color : black	TV type : Full HD	Internet : yes	Cost : 899 Euro	Sub/ire : Plaisio Patras	Company : PLAISIO
Model : Panasonic TX-40EX600E	Size : 49 Inches	Color : black	TV type : Full HD	Internet : yes	Cost : 650 Euro	Sub/ire : Plaisio Heraklion	Company : PLAISIO
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Kotsovolos Rethymno	Company : KOTSOVOLOS
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Kotsovolos Chania	Company : KOTSOVOLOS
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Public Alexadroupoli	Company : KOTSOVOLOS
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Kotsovolos Rethymno	Company : PUBLIC
Model : LG 60SJ810V	Size : 60 Inches	Color : black	TV type : Ultra HD	Internet : yes	Cost : 1448 Euro	Sub/ire : Germanos Rhodes	Company : GERMANOS
Model : LG 60UH605V	Size : 62 Inches	Color : black	TV type : Full HD	Internet : yes	Cost : 899 Euro	Sub/ire : Plaisio Patras	Company : PLAISIO
Model : Samsung UE49M6300	Size : 49 Inches	Color : black	TV type : Full HD	Internet : yes	Cost : 830 Euro	Sub/ire : Public Athens	Company : PUBLIC

Εικόνα 38: JTextArea για εκτύπωση τηλεοράσεων

Με παρόμοιο τρόπο εαν επιλέξουμε (Εικόνα: 39) το υποκατάστημα PUBLIC θα μας ανοίξει τρία JTextArea που θα περιέχουν τα προϊόντα κάθε υποκαταστήματος της εταιρείας όσο αφορά τα προϊόντα mobiles, laptops και televisions

Εικόνα 39: Το JFrame για εκτύπωση προϊόντων ανα εταιρεία

Επομένως (Εικόνα : 40) βλέπουμε τα mobiles της εταιρείας

Company's Laptops																	
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Athens Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Athens Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Athens Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Athens Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Xaidari Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Xaidari Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Athens Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Athens Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Xaidari Com/ny	Com/ny	: PUBLIC
Model	: LENOVO G750	Size	: 16 Inches	GPU	: NVIDIA GTX1050GB	CPU	: 3.6Ghz	RAM	: 16GB	Cost	: 1450 Euro	Warr	: 2 Years	Sub/re	: Public Athens Com/ny	Com/ny	: PUBLIC
Model	: ACER C740-C9QX	Size	: 15.4 Inches	GPU	: NVIDIA GTX980GB	CPU	: 3.2Ghz	RAM	: 8GB	Cost	: 1100 Euro	Warr	: 3 Years	Sub/re	: Public Xaidari Com/ny	Com/ny	: PUBLIC

Εικόνα 42: Τα laptops της εταιρείας PUBLIC

Για παράδειγμα εάν θέλουμε να εκτυπώσουμε ανα προϊόν βλέποντας τον κώδικα(Εικόνα : 43)

```
//print laptops. We dont want to print the "Test entry"
if (!childNodes0.getTextContent().equals("test_model")) {
    String printing_result
        = "Model : " + childNode0.getTextContent() + "
        + "Size : " + childNode1.getTextContent() + " Inches"
        + " GPU : " + childNode2.getTextContent() + "GB "
        + " CPU : " + childNode3.getTextContent() + "Ghz "
        + "RAM : " + childNode4.getTextContent() + "GB "
        + "Cost : " + childNode5.getTextContent() + " Euro "
        + "Warr : " + childNode6.getTextContent() + " Years "
        + "Sub/re : " + childNode7.getTextContent()
        + " Com/ny : " + childNode8.getTextContent() + "\n"
        + "-----"
        + "-----"
        + "-----"
        + "-----" + "\n ";
    printing_category_area.setEditable(false);
    //laptops_printing.add(printing_category_area);
    JScrollPane scrollPane = new JScrollPane(printing_category_area);
    laptops_printing.add(scrollPane);
    printing_category_area.append(printing_result);
}
```

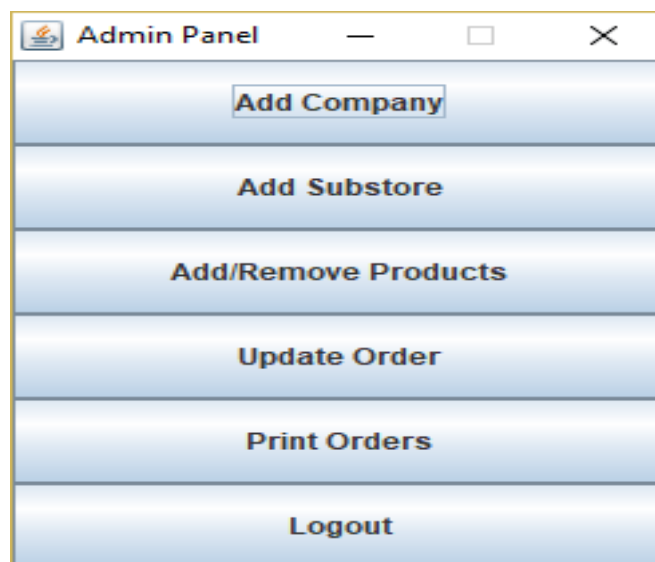
Εικόνα 43: Κώδικας για εκτύπωση laptops

Να σημειώσουμε ότι για να γίνει append σε ένα αρχείο πρέπει να έχουμε ήδη ένα entry μέσα. Αυτό το entry που βάλουμε είναι ένα «test entry» που δεν διαγράφεται αλλά ούτε εκτυπώνεται πουθενά.

Βλέποντας τον κώδικα κάθε φορά ανοίγουμε το αρχείο (είτε προκειται για εκτύπωση ανα εταιρεία είτε ανα προϊόν είτε κάτι άλλο) παίρνουμε τα elements και στη συνέχεια τα εκτυπώνουμε μορφοποιημένα με τη χρήση JTextArea και JScrollPane

ΟΙ ΛΕΙΤΟΥΡΓΙΕΣ ΤΟΥ ADMIN

Ο admin εισέρχεται στο σύστημα με **username = dim93** και **password = 09**. Αφού εισέλθει εμφανίζεται το παρακάτω JFrame (*Εικόνα : 44*)



Εικόνα 44: Το κεντρικό JFrame του admin

Επομένως μπορούμε να επιλέξουμε κάποιες από τις παραπάνω επιλογές πατώντας το κατάλληλο κουμπί που θα δημιουργήσει το κατάλληλο event. Με το κουμπί logout απλά αποσυνδεόμαστε από το σύστημα.

ADD COMPANY

Με το που κάνουμε κλικ στο κουμπί add company μας ανοίγει το παρακάτω JFrame (*Εικόνα :45*)

The image shows a Java Swing window titled "Add Company". It contains three text input fields stacked vertically. The first field is labeled "Company Name", the second "Company Address", and the third "Company TIN". At the bottom of the window is a blue button labeled "ADD".

Εικόνα 45: Το JFrame add company

Οι εταιρείες βρίσκονται στο αρχείο company_XML.xml όπως φαίνεται παρακάτω (*Εικόνα :46*)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><companies>
<company><name>PLASIO</name><address>Neolaias 243</address><TIN>289202938</TIN></company>
<company><fname>GERMANOS</fname><lname>Parlama 7</lname><address>827384982</address></company>
<company><fname>KOTSOVOLOS</fname><lname>Lexou 111</lname><address>898720087</address></company>
<company><name>PUBLIC</name><address>Ionias 72 </address><TIN>112309182</TIN></company>
<company><name>E-SHOP</name><address>Euripidou 12</address><TIN>133309132</TIN></company>
<company><name>PLEXOUDAKIS</name><address>Gournas 4</address><TIN>327381982</TIN></company></companies>
```

Εικόνα 46: Το αρχείο companies_XML.xml

Εαν προσπαθήσουμε να εισάγουμε μια εταιρεία όπως φαίνεται και στην παρακάτω εικόνα (*Εικόνα : 47*)

The image shows the "Add Company" dialog box with the following data entered: "Company Name" is "ZAFEIRIADIS", "Company Address" is "Leontiadi 42", and "Company TIN" is "827389098". The "ADD" button is highlighted at the bottom.

Εικόνα 47: Εισαγωγή εταιρείας σε XML αρχείο

Τότε θα δούμε οτι το XML file έχει αλλάξει και έχει προστεθεί ως νέα εγγραφή η παραπάνω εταιρεία (*Εικόνα : 48*)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><companies>
<company><name>PLASIO</name><address>Neolaias 243</address><TIN>289202938</TIN></company>
<company><fname>GERMANOS</fname><lname>Parlama 7</lname><address>827384982</address></company>
<company><fname>KOTSOVOLOS</fname><lname>Lexou 111</lname><address>898720087</address></company>
<company><name>PUBLIC</name><address>Ionias 72 </address><TIN>112309182</TIN></company>
<company><name>E-SHOP</name><address>Euripidou 12</address><TIN>133309132</TIN></company>
<company><name>PLEXOUDAKIS</name><address>Gournas 4</address><TIN>327381982</TIN></company>
<company><name>ZAFEIRIADIS</name><address>Leontiadi 42</address><TIN>827389098</TIN></company></companies>
```

Εικόνα 48: Η εταιρεία ZAFEIRIADIS έχει προστεθεί στο XML αρχείο

Όσο αφορά τον κώδικα(*Εικόνα : 49*)

```
public class AddCompany implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent ae) {
        JFrame company_frame = new JFrame();
        company_frame.setTitle("Add Company");
        company_frame.setSize(250, 300);
        company_frame.setLayout(new BorderLayout());
        JPanel company_panel = new JPanel();
        company_frame.add(company_panel);
        company_panel.setLayout(new GridLayout(7, 1));
        JLabel label1 = new JLabel("Company Name");
        JLabel label2 = new JLabel("Company Address");
        JLabel label3 = new JLabel("Company TIN");
        JTextField text1 = new JTextField(40);
        JTextField text2 = new JTextField(40);
        JTextField text3 = new JTextField(40);
        JButton add = new JButton("ADD");
        company_panel.add(label1);
        company_panel.add(text1);
        company_panel.add(label2);
        company_panel.add(text2);
        company_panel.add(label3);
        company_panel.add(text3);
        company_panel.add(add);
        add.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
```

Εικόνα 49: Εισαγωγή JFrame για την company

Αρχικά δημιουργούμε ένα JFrame και χρησιμοποιούμε ένα button το οποίο διαθέτει ActionListener. Αρχικά πρέπει να ελέγξουμε εάν υπάρχει ήδη άλλο ονομα εταιρείας με το ίδιο ονομα ή τον ίδιο αριθμό μητρώου που αυτό γίνεται με έναν counter.

Αφού λάβουμε τα δεδομένα της φόρμας από το JFrame τότε ανοίγουμε το αρχείο companies_XML.xml και χρησιμοποιούμε μια μεταβλητή η οποία αυξάνεται ή μεινεί στάσιμη ανάλογα με το εάν βρήκαμε κάτι μέσα στο αρχείο που ήδη δώσαμε μέσω της φόρμας.

Εαν όλα είναι εντάξει τότε καλείται η συνάρτηση `addCompany()` που δέχεται ως ορίσματα το όνομα, την διεύθυνση και το αριθμο φορολογικού μητρώου που δώσαμε. Στην παρακάτω εικόνα βλέπουμε την όλη λογική που επεξηγήσαμε(*Εικόνα : 50*)

```
try {
    //BONUS : check if this company or TIN exists in our system.
    //we must not have two companies with the same name and same TIN
    File file0 = new File("companies_XML.xml");
    // Create builders
    DocumentBuilderFactory dbFactory0 = DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder0 = dbFactory0.newDocumentBuilder();
    Document doc0 = dBuilder0.parse(file0);
    doc0.getDocumentElement().normalize();
    int counter = 1;

    NodeList nList0 = doc0.getElementsByTagName("company");
    for (int i = 0; i < nList0.getLength(); i++) {
        NodeList childList = nList0.item(i).getChildNodes();
        Node childNode0 = childList.item(0);
        Node childNode2 = childList.item(2);
        if (childNode0.getTextContent().equals(name) || childNode2.getTextContent().equals(TIN)) {
            counter++;
            break;
        }
    }
    if (counter <= 1) {
        addCompany(name, address, TIN);
        company_frame.dispose();
    }
    if (counter > 1) {
        System.out.println("This company or TIN belongs to our system!");
    }
}
```

Εικόνα 50: Έλεγχος για διπλότυπα ονόματα και TIN εταιριών

Η συνάρτηση απλά κάνει append την συγκεκριμένη εταιρεία που δώσαμε(*Εικόνα :51*)

```
File file = new File("companies_XML.xml");
// Create builders
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(file);
doc.getDocumentElement().normalize();

//write content to XML file
Element company_v = doc.createElement("company");

//add children to customer element
Element name_v = doc.createElement("name");
name_v.appendChild(doc.createTextNode(name));
company_v.appendChild(name_v);

Element address_v = doc.createElement("address");
address_v.appendChild(doc.createTextNode(address));
company_v.appendChild(address_v);

Element TIN_v = doc.createElement("TIN");
TIN_v.appendChild(doc.createTextNode(TIN));
company_v.appendChild(TIN_v);

//append company element to companies root
Element companies_v;
companies_v = (Element) doc.getElementsByTagName("companies").item(0);
companies_v.appendChild(company_v);
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(new File("companies_XML.xml"));
transformer.transform(source, result);
```

Εικόνα 51: appending εταιριών στο companies_XML.xml

ADD SUBSTORE

Κάνοντας κλικ στο κουμπί add substore ανοίγει το κάτωθι JFrame(*Εικόνα : 52*)

Εικόνα 52: Το JFrame για εισαγωγή substore

Για να εισάγουμε στο σύστημα ένα substore χρειαζόμαστε τα παραπάνω πεδία καθώς και το όνομα της εταιρείας για να μπορέσουμε να βρούμε συσχετίσεις μεταξύ του αρχείου των εταιρειών και των υποκαταστημάτων .

Το αρχείο substores_XML.xml περιέχει υποκαταστήματα και είναι ως εξής (**Εικόνα : 53**)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><substores>
<substore><substore_name>Public Athens</substore_name><substore_address>Bouliagmenis 42</substore_address><company_name>PUBLIC</
<substore><substore_name>Plaisio Heraklion</substore_name><substore_address>Evans 78</substore_address> <company_name>PLAISIO</c
<substore><substore_name>Germanos Rhodes</substore_name><substore_address>Vakxou 99</substore_address><company_name>GERMANOS</c
<substore><substore_name>Kotsovolos Chania</substore_name><substore_address>Terpou 23</substore_address><company_name>KOTSOVOLOS
<substore><substore_name>Germanos Kalamata</substore_name><substore_address>Vryxelwn 43</substore_address><company_name>GERMANC
<substore><substore_name>Kotsovolos Rethymno</substore_name><substore_address>Venezourou 19</substore_address><company_name>KOTS
<substore><substore_name>Plaisio Patras</substore_name><substore_address>Kornoualis 1</substore_address><company_name>PLAISIO</c
<substore><substore_name>Public Xaidari</substore_name><substore_address>Ermou 12</substore_address><company_name>PUBLIC</compan
<substore><substore_name>Public Alexandroupoli</substore_name><substore_address>Vamonos 4</substore_address><company_name>PUBLIC
<substore><substore_name>Kotsovolos Larissa</substore_name><substore_address>Merarxias 87</substore_address><company_name>KOTSOV
<substore><substore_name>Plexoudakis Heraklion</substore_name><substore_address>Papandreou 224</substore_address><company_name>P
```

Εικόνα 53: Το αρχείο substores_XML.xml

Εάν προσπαθήσουμε να βάλουμε ένα υποκατάστημα για παράδειγμα στην εταιρία PUBLIC(**Εικόνα : 54**)

Substore Name	Public Chios
Substore Address	Nearxou 11
Company Name	PUBLIC
Substore TIN	872111783
Chief Firstname	Emanouil
Chief Lastname	Mastorakis
Chief Telephone	6940088672
ADD	

Εικόνα 54: Εισαγωγή υποκαταστήματος στο σύστημα

Τότε θα δούμε το αρχείο substores_XML.xml να αλλάξει(**Εικόνα : 55**)

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><substores>
<substore><substore_name>Public Athens</substore_name><substore_address>Bouliagmenis 42</substore_address><company_name>PUBLIC</c
<substore><substore_name>Plaisio Heraklion</substore_name><substore_address>Evans 78</substore_address> <company_name>PLAISIO</c
<substore><substore_name>Germanos Rhodes</substore_name><substore_address>Vakxou 99</substore_address><company_name>GERMANOS</c
<substore><substore_name>Kotsovolos Chania</substore_name><substore_address>Terpou 23</substore_address><company_name>KOTSOVOLOS
<substore><substore_name>Germanos Kalamata</substore_name><substore_address>Vryxelwn 43</substore_address><company_name>GERMANC
<substore><substore_name>Kotsovolos Rethymno</substore_name><substore_address>Venezourou 19</substore_address><company_name>KOTS
<substore><substore_name>Plaisio Patras</substore_name><substore_address>Kornoualis 1</substore_address><company_name>PLAISIO</c
<substore><substore_name>Public Kaidari</substore_name><substore_address>Ermou 12</substore_address><company_name>PUBLIC</compar
<substore><substore_name>Public Alexandroupoli</substore_name><substore_address>Vamonos 4</substore_address><company_name>PUBLIC
<substore><substore_name>Kotsovolos Larissa</substore_name><substore_address>Merarxias 87</substore_address><company_name>KOTSOV
<substore><substore_name>Plexoudakis Heraklion</substore_name><substore_address>Papandreou 224</substore_address><company_name>E
<substore><substore_name>Public Chios</substore_name><substore_address>Nearxou 11</substore_address><company_name>PUBLIC</compar

```

Εικόνα 55: Εισαγωγή του υποκαταστήματος Public Chios

Ο κώδικας είναι παρόμοιος με το προηγούμενο κομμάτι όπου αναφερθήκαμε στο πώς μπορούμε να εισάγουμε μια εταιρεία στο σύστημα. Αρχικά δηλαδή δημιουργούμε ένα JFrame όπου μέσω κατάλληλων JTextFields θα πάρουμε τα δεδομένα και στη συνέχεια (**Εικόνα : 56**) θα κάνουμε έλεγχο για το εάν υπάρχει ήδη ένα υποκατάστημα με το ίδιο όνομα ή φορολογικό μητρώο.

```

public void actionPerformed(ActionEvent ae) {
    String substore_name = text1.getText();
    String substore_address = text2.getText();
    String company_name = text3.getText();
    String substore_TIN = text4.getText();
    String chief_fname = text5.getText();
    String chief_lname = text6.getText();
    String chief_telephone = text7.getText();
    //BONUS IS to check if this substore name or TIN exist in our system
    try {
        File file0 = new File("substores_XML.xml");
        // Create builders
        DocumentBuilderFactory dbFactory0 = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder0 = dbFactory0.newDocumentBuilder();
        Document doc0 = dBuilder0.parse(file0);
        doc0.getDocumentElement().normalize();
        int counter = 1;

        NodeList nList0 = doc0.getElementsByTagName("substore");
        for (int i = 0; i < nList0.getLength(); i++) {
            NodeList childList = nList0.item(i).getChildNodes();
            Node childNode0 = childList.item(0);
            Node childNode3 = childList.item(3);
            if (childNode0.getTextContent().equals(substore_name) || childNode3.getTextContent().equals(substore_TIN))
                counter++;
            break;
        }
    }
}

```

Εικόνα 56: JFrame και έλεγχος διπλότυπων χαρακτηριστικών για το substores_XML.xml

```

        Node childNode0 = childList.item(0);
        Node childNode3 = childList.item(3);
        if (childNode0.getTextContent().equals(substore_name) || childNode3.getTextContent().equals(substore_
            counter++;
            break;
        }
    }
    if (counter <= 1) {
        addSubstore(substore_name, substore_address, company_name, substore_TIN, chief_fname, chief_lname, ch
        substore_frame.dispose();
    }
    if (counter > 1) {
        System.out.println("This substore or TIN belongs to our system!");
    }
} catch (ParserConfigurationException | SAXException | IOException | TransformerException ex) {
    Logger.getLogger(AddCompany.class.getName()).log(Level.SEVERE, null, ex);
    substore_frame.dispose();
}
}
});

substore_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
substore_frame.setVisible(true);
substore_frame.setResizable(false);
}

```

Εικόνα 57:Κλήση κατάλληλης συνάρτησης για εισαγωγή ενός υποκαταστήματος

Ευθύς αμέσως βλέπουμε την κλήση της συνάρτησης `addSubstore` που δέχεται ως ορίσματα τα δεδομένα των `JTextFields` που στείλαμε μέσω του `JFrame` ως φόρμα εγγραφής ενός υποκαταστήματος στο σύστημα.

Ο κώδικας της συνάρτησης φαίνεται παρακάτω (**Εικόνα : 58**) . Για εξοικονόμηση χώρου δεν βάζουμε όλο τον κώδικα επακριβώς γιατί το αρχείο θα γινότανε πολύ μεγάλο σε μέγεθος απο το πλήθος των εικόνων.

```
if (j == 0) {
    File file = new File("substores_XML.xml");
    // Create builders
    DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
    Document doc = dBuilder.parse(file);
    doc.getDocumentElement().normalize();

    //write content to XML file
    Element substore_v = doc.createElement("substore");
    //add children to customer element
    Element substore_name_v = doc.createElement("substore_name");
    substore_name_v.appendChild(doc.createTextNode(substore_name));
    substore_v.appendChild(substore_name_v);

    Element substore_address_v = doc.createElement("substore_address");
    substore_address_v.appendChild(doc.createTextNode(substore_address));
    substore_v.appendChild(substore_address_v);

    Element company_name_v = doc.createElement("company_name");
    company_name_v.appendChild(doc.createTextNode(company_name));
    substore_v.appendChild(company_name_v);

    Element substore_TIN_v = doc.createElement("substore_TIN");
    substore_TIN_v.appendChild(doc.createTextNode(substore_TIN));
    substore_v.appendChild(substore_TIN_v);
}
```

Εικόνα 58: appending υποκαταστήματα στο `substores_XML.xml`

Βλέπουμε ότι όπως και τις άλλες φορές κάνουμε `append` στο αρχείο τα δοσμένα στοιχεία του υποκαταστήματος που έδωσε ο πελάτης.

ADD/REMOVE PRODUCTS

Πατώντας το κουμπί add/remove products μας ανοίγει το παρακάτω κεντρικό JFrame που μας δίνει τη δυνατότητα να προσθέσουμε ή να αφαιρέσουμε προϊόντα απο το σύστημα(

Εικόνα : 59)

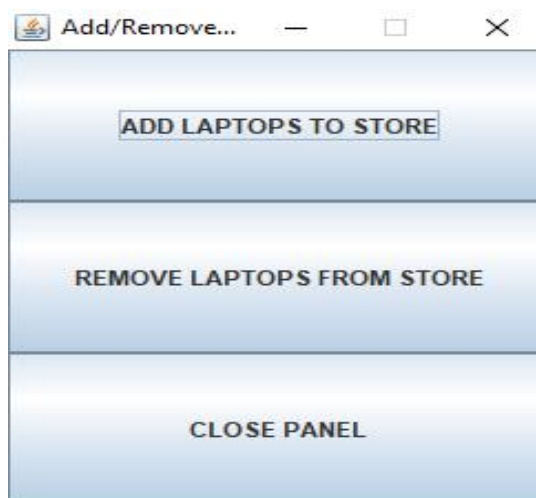


Εικόνα 59: Κεντρικό JFrame για add/remove προϊόντων

Ξεχωριστά θα παρουσιάσουμε την εισαγωγή και την διαγραφή προϊόντων .

ADD/REMOVE LAPTOPS

Επιλέγοντας το κουμπί add/remove laptops μας εμφανίζεται το παρακάτω(*Εικόνα : 60)*



Εικόνα 60: Add/remove laptops απο το σύστημα

Αν πατήσουμε add laptops to store τότε (*Εικόνα : 61*)

Εικόνα 61: Εισαγωγή laptop στο σύστημα

Τα προϊόντα τύπου laptop γράφονται στο αρχείο laptops_XML.xml και η λογική είναι απλά να διαβάσουμε τα δεδομένα της φόρμας και στη συνέχεια να τα πάρουμε με τη χρήση των JTextFields και στη συνέχεια να ανοίξουμε το αρχείο, να δούμε εάν υπάρχει κάτι παρόμοιο μέσα. Εάν δεν υπάρχει τότε κάνουμε append κανονικά με το γνωστό τρόπο

Ας δούμε για παράδειγμα το αρχείο laptops_XML.xml που περιέχει μοντέλα τύπου laptop(*Εικόνα : 62)*

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><laptops>
<laptop><laptop_model>test_model</laptop_model><laptop_screenSize>test_scrSize</laptop_screenSize><laptop_GPU>test_gpu</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>LENOVO G750</laptop_model><laptop_screenSize>16</laptop_screenSize><laptop_GPU>NVIDIA GTX1050</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
<laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop_GPU>
</laptops>
```

Εικόνα 62: Το αρχείο laptops_XML.xml

Ας βάλουμε το παρακάτω laptop στο σύστημα μας (*Εικόνα : 63*)

Laptop model	
SAMSUNG HT-82	
Laptop screen size	
17.3	
Laptop GPU	
NVIDIA GTX150	
Laptop CPU	
3.2	
Laptop RAM	
32	
Laptop cost	
2950	
Laptop warranty	
2	
Substore Name	
Public Athens	
Company Name	
Public	
Laptops Quantity	
1	
ADD ITEMS	

Εικόνα 63: Εισαγωγή laptop στο laptops_XML.xml

Τότε θα δούμε οτι το αρχείο laptops_XML.xml έχει αλλάξει(*Εικόνα : 64*)

```

14 <laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop
15 <laptop><laptop_model>ACER C740-C9QX</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>NVIDIA GTX980</laptop
16 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
17 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
18 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
19 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
20 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
21 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
22 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
23 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
24 <laptop><laptop_model>Dell Inspiron 3162</laptop_model><laptop_screenSize>15.4</laptop_screenSize><laptop_GPU>HD 5500</laptop_G
25 <laptop><laptop_model>SAMSUNG HT-82</laptop_model><laptop_screenSize>17.3</laptop_screenSize><laptop_GPU>NVIDIA GTX150</laptop_G

```

Εικόνα 64: Το μοντέλο SAMSUNG HT-82 έχει εγγραφεί στο αρχείο

Ας δούμε το πώς λειτουργεί ο κώδικας (*Εικόνα : 65*)

```

public class AddLaptops implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent ae) {
        JFrame prod_frame = new JFrame();
        prod_frame.setTitle("Add Laptops");
        prod_frame.setSize(300, 600);
        prod_frame.setLayout(new BorderLayout());
        JPanel prod_panel = new JPanel();
        prod_frame.add(prod_panel);
        prod_panel.setLayout(new GridLayout(21, 1));
        JLabel prod_label1 = new JLabel("Laptop model");
        JLabel prod_label2 = new JLabel("Laptop screen size");
        JLabel prod_label3 = new JLabel("Laptop GPU");
        JLabel prod_label4 = new JLabel("Laptop CPU");
        JLabel prod_label5 = new JLabel("Laptop RAM");
        JLabel prod_label6 = new JLabel("Laptop cost");
        JLabel prod_label7 = new JLabel("Laptop warranity");
        JLabel prod_label8 = new JLabel("Substore Name");
        JLabel prod_label9 = new JLabel("Company Name");
        JLabel prod_label10 = new JLabel("Laptops Quantity");

        JTextField prod_text1 = new JTextField(30);
        JTextField prod_text2 = new JTextField(30);
        JTextField prod_text3 = new JTextField(30);
        JTextField prod_text4 = new JTextField(30);
        JTextField prod_text5 = new JTextField(30);
        JTextField prod_text6 = new JTextField(30);
        JTextField prod_text7 = new JTextField(30);
    }
}

```

Εικόνα 65: Κώδικα για την εισαγωγή ενός laptop

Αρχικά δημιουργούμε ένα JFrame το οποίο θα μας βοηθήσει να πάρουμε τα δεδομένα από τον admin με σκοπό την εισαγωγή ενός laptop στο σύστημα. Κατόπιν (**Εικόνα : 66**) καλούμε την συνάρτηση AddLaptop με ορίσματα , τα δεδομένα που έδωσε ο χρήστης.

```

add_laptops_to_store.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        //get form values
        String laptop_model = prod_text1.getText();
        String laptop_screenSize = prod_text2.getText();
        String laptop_GPU = prod_text3.getText();
        String laptop_CPU = prod_text4.getText();
        String laptop_RAM = prod_text5.getText();
        String laptop_cost = prod_text6.getText();
        String laptop_warranty = prod_text7.getText();
        String substore_name = prod_text8.getText();
        String laptop_company_name = prod_text9.getText();
        String laptop_quantity = prod_text10.getText();

        //logic for add a laptop
        try {
            AddLaptop(laptop_model, laptop_screenSize, laptop_GPU, laptop_CPU, laptop_RAM, laptop_cost, lap
            prod_frame.dispose();
        } catch (ParserConfigurationException | SAXException | IOException | TransformerException ex) {
            Logger.getLogger(AddLaptops.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
});
prod_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
prod_frame.setVisible(true);
prod_frame.setResizable(false);
}

```

Εικόνα 66:Κώδικα για την εισαγωγή ενός laptop (2)

Στη συνέχεια μόλις καλέσουμε την συνάρτηση, τότε αυτή ανοίγει το αρχείο laptops_XML.xml . Τέλος εισάγει τα στοιχεία με τη χρήση append στο αρχείο(*Εικόνα : 67*)

```

if (j == 0) {
    //we need to add laptops using quantity given
    for (int k = 0; k < laptops_quantity; k++) {
        File file = new File("laptops_XML.xml");
        // Create builders
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
        Document doc = dBuilder.parse(file);
        doc.getDocumentElement().normalize();

        //write content to XML file
        Element laptop_v = doc.createElement("laptop");

        //add children to customer element
        Element lp_model = doc.createElement("laptop_model");
        lp_model.appendChild(doc.createTextNode(laptop_model));
        laptop_v.appendChild(lp_model);

        Element lp_ScreenSize = doc.createElement("laptop_screenSize");
        lp_ScreenSize.appendChild(doc.createTextNode(laptop_screenSize));
        laptop_v.appendChild(lp_ScreenSize);

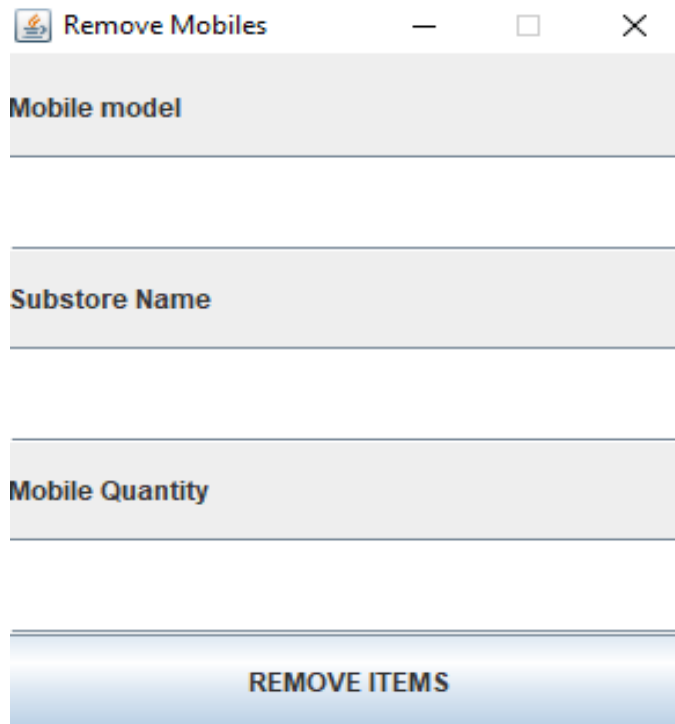
        Element lp_GPU = doc.createElement("laptop_GPU");
        lp_GPU.appendChild(doc.createTextNode(laptop_GPU));
        laptop_v.appendChild(lp_GPU);
    }
}

```

Εικόνα 67: Κλήση της συνάρτησης AddLaptop

Με παρόμοιο τρόπο γίνεται και η εισαγωγή ενός mobile ή μιας television. Όμως για να μπορέσουμε να διαγράψουμε ένα προϊόν από τα XML αρχεία (laptops_XML.xml, mobiles_XML.xml, televisions_XML.xml) πρέπει να ακολουθήσουμε αρκετά διαφορετική τακτική από την προσθήκη προϊόντων.

Εστω για παράδειγμα ότι θέλουμε να σβήσουμε ένα mobile από το αρχείο mobiles_XML.xml(**Εικόνα : 68**)



The image shows a Java Swing window titled "Remove Mobiles". It has a standard title bar with a minimize button, a maximize button, and a close button. The window contains three text input fields stacked vertically. The first field is labeled "Mobile model", the second is labeled "Substore Name", and the third is labeled "Mobile Quantity". Below these fields is a blue button with the text "REMOVE ITEMS".

Εικόνα 68: JFrame για διαγραφή προϊόντος από το mobiles_XML.xml

Για να σβήσουμε ένα κινητό απαιτείται να δώσουμε το μοντέλο του κινητού τηλεφώνου που φυσικά πρέπει ήδη να υπάρχει στο αρχείο, το υποκατάστημα από το οποίο εμείς θέλουμε να σβήσουμε και την ποσότητα(εφόσον υπάρχει).

Το βασικό JFrame δημιουργείται στην κλάση AddRemoveProduct() όπου μέσω actionListeners στέλνουμε κάθε φορά ένα event στην κατάλληλη remove κλάση. Εμείς θα μελετήσουμε την κλάση RemoveMobiles().

Πιο κάτω βλέπουμε ότι μέσω του JFrame ζητάμε κάποια στοιχεία τα οποία θα χρησιμοποιηθούν για να σβήσουμε(**Εικόνα: 69**)

```

@Override
public void actionPerformed(ActionEvent ae) {
    JFrame prod_frame = new JFrame();
    prod_frame.setTitle("Remove Mobiles");
    prod_frame.setSize(300, 350);
    prod_frame.setLayout(new BorderLayout());
    JPanel prod_panel = new JPanel();
    prod_frame.add(prod_panel);
    prod_panel.setLayout(new GridLayout(7, 1));
    JLabel prod_label1 = new JLabel("Mobile model");
    JLabel prod_label2 = new JLabel("Substore Name");
    JLabel prod_label3 = new JLabel("Mobile Quantity");
    JTextField prod_text1 = new JTextField(30);
    JTextField prod_text2 = new JTextField(30);
    JTextField prod_text3 = new JTextField(30);
    JButton remove_mobiles_from_store = new JButton("REMOVE ITEMS");

    prod_panel.add(prod_label1);
    prod_panel.add(prod_text1);
    prod_panel.add(prod_label2);
    prod_panel.add(prod_text2);
    prod_panel.add(prod_label3);
    prod_panel.add(prod_text3);
    prod_panel.add(remove_mobiles_from_store);
    remove_mobiles_from_store.addActionListener(new ActionListener() {

```

Εικόνα 69: Βασικό JFrame για διαγραφή mobile

Ας δούμε όμως το αρχείο των mobiles πρώτου διαγράψουμε κινητά τηλέφωνα και έπειτα το αρχείο, το οποίο πλέον δεν τα διαθέτει(Εικόνα : 70)

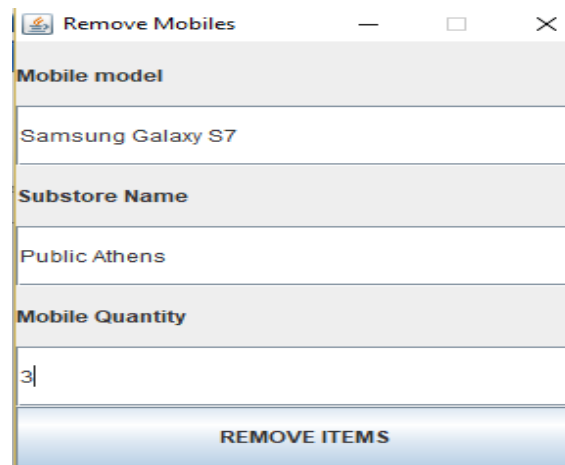
```

1  <?xml version="1.0" encoding="UTF-8" standalone="no"?><mobiles>
2  <mobile><mobile_model>test_model</mobile_model><mobile_front_camera>test_FrCam</mobile_front_camera><mobile_back_camera>test_Bc
3  <mobile><mobile_model>Samsung Galaxy S6</mobile_model><mobile_front_camera>12</mobile_front_camera><mobile_back_camera>6</mobil
4  <mobile><mobile_model>Samsung Galaxy S6</mobile_model><mobile_front_camera>12</mobile_front_camera><mobile_back_camera>6</mobil
5  <mobile><mobile_model>Samsung Galaxy S6</mobile_model><mobile_front_camera>12</mobile_front_camera><mobile_back_camera>6</mobil
6  <mobile><mobile_model>Samsung Galaxy S6</mobile_model><mobile_front_camera>12</mobile_front_camera><mobile_back_camera>6</mobil
7  <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
8  <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
9  <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
10 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
11 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
12 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
13 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
14 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
15 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
16 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
17 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
18 <mobile><mobile_model>Samsung Galaxy S7</mobile_model><mobile_front_camera>16</mobile_front_camera><mobile_back_camera>8</mobil
19 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_
20 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_
21 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_
22 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_
23 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>6</mobile_back_
24 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_
25 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_
26 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_
27 <mobile><mobile_model>i-Phone S7</mobile_model><mobile_front_camera>14</mobile_front_camera><mobile_back_camera>8</mobile_back_

```

Εικόνα 70: Το αρχείο mobiles_XML.xml πριν τη διαγραφή mobiles

Ας σβήσουμε για παράδειγμα τρία κινητά Samsung S7 (*Εικόνα: 71*)



Εικόνα 71: Διαγραφή mobile απο το αρχείο mobiles_XML.xml

Τότε η γραμμή 11 του αρχείου έχει σβηστεί (*Εικόνα : 72*)

```
10 CPU<mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>German
11
12 CPU<mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>Public
13 CPU<mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>German
14 CPU<mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>German
```

Εικόνα 72: Διαγραφή 11ης γραμμής απο το αρχείο mobiles_XML.xml

Επισης και η γραμμή 36 (*Εικόνα: 73*)

```
33 obile_screen_size>5.4</mobile_screen_size><mobile_RAM>4</mobile_RAM><mobile_cost>1250</mobile_cost><substore_name>Kotsovolos Rh
34 obile_screen_size>5.4</mobile_screen_size><mobile_RAM>4</mobile_RAM><mobile_cost>1250</mobile_cost><substore_name>Kotsovolos Rh
35 obile_screen_size>5.4</mobile_screen_size><mobile_RAM>4</mobile_RAM><mobile_cost>1250</mobile_cost><substore_name>Kotsovolos Rh
36
37 CPU<mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>Public
38 CPU<mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>German
39 CPU<mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>German
```

Εικόνα 73 : Διαγραφή 36ης γραμμής απο το αρχείο mobiles_XML.xml

Το ίδιο συμβαίνει και με την γραμμή 54(**Εικόνα : 74**)

```
50 <mobile_screen_size>5.4</mobile_screen_size><mobile_RAM>4</mobile_RAM><mobile_cost>1250</mobile_cost><substore_name>Kotsovolos Rh  
51 <mobile_screen_size>5.4</mobile_screen_size><mobile_RAM>4</mobile_RAM><mobile_cost>1250</mobile_cost><substore_name>Public Athen  
52 <mobile_screen_size>5.4</mobile_screen_size><mobile_RAM>4</mobile_RAM><mobile_cost>1250</mobile_cost><substore_name>Germanos Athe  
53 <mobile_screen_size>5.4</mobile_screen_size><mobile_RAM>4</mobile_RAM><mobile_cost>1250</mobile_cost><substore_name>Germanos Athe  
54  
55 <CPU><mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>Public  
56 <CPU><mobile_screen_size>5.8</mobile_screen_size><mobile_RAM>6</mobile_RAM><mobile_cost>1100</mobile_cost><substore_name>German
```

Εικόνα 74: Διαγραφή 54ης γραμμής απο το αρχείο mobiles_XML.xml

Στη συνέχεια, για να διαγράψουμε ένα κινητό (**Εικόνα : 75**) καλείται η συνάρτηση RemoveMobile που δέχεται ως όρισμα τα δεδομένα του JFrame. Κατόπιν όπως και πάντα άλωστε βάζουμε τα κατάλληλα catches.

```
remove_mobiles_from_store.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
        String mobile_model = prod_text1.getText();  
        String sub_store_name = prod_text2.getText();  
        String mobile_quantity = prod_text3.getText();  
        try {  
            RemoveMobile(mobile_model, sub_store_name, mobile_quantity);  
            prod_frame.dispose();  
        } catch (ParserConfigurationException | SAXException | IOException | TransformerException ex) {  
            Logger.getLogger(RemoveMobiles.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
});  
prod_frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
prod_frame.setVisible(true);  
prod_frame.setResizable(false);
```

Η συνάρτηση αρχικά ανοίγει το αρχείο mobiles_XML.xml . Παράλληλα, χρησιμοποιεί μια Nodelist (όπως πάντα) για να βρεί όλα τα elements του αρχείου που είναι τύπου mobile. Για να γίνει αυτό θέλουμε πρώτα να βρούμε το μέγεθος της χρησιμοποιώντας .getLength() .

Έπειτα τρέχουμε μια for statement για να πάρουμε όλες τις τιμές όλων των παιδιών όλων των elements αυτής της λίστας.

Το **σημαντικό** σημείο εδώ είναι ότι έχοντας την ποσότητα κινητών τηλεφώνων που εμείς θέλουμε να σβήσουμε, και χρησιμοποιώντας μια μεταβλητή τύπου counter για να βρούμε πόσα κινητά υπάρχουν στη λίστα, σαν το συγκεκριμένο είδος που εμείς ζητήσαμε να σβήσουμε;

Εάν επαρκεί τότε χρησιμοποιώντας πάλι for statement θα σβήσουμε την ποσότητα κινητών τηλεφώνων που δώσαμε. Όμως επειδή μετά από μια διαγραφή μένουν συνήθως κενοί κόμβοι, χρησιμοποιούμε τη συνάρτηση removeEmptyNodes() για να καθαρίσουμε το αρχείο(**Εικόνα: 77**)

```
//checking if node specific laptop_model and sub_store_name exists
if (childNodes0.getTextContent().equals(mobile_model) && childNode8.getTextConte:
    //delete all children of this node
    childNode0.getParentNode().removeChild(childNode0);
    childNode1.getParentNode().removeChild(childNode1);
    childNode2.getParentNode().removeChild(childNode2);
    childNode3.getParentNode().removeChild(childNode3);
    childNode4.getParentNode().removeChild(childNode4);
    childNode5.getParentNode().removeChild(childNode5);
    childNode6.getParentNode().removeChild(childNode6);
    childNode7.getParentNode().removeChild(childNode7);
    childNode8.getParentNode().removeChild(childNode8);
    childNode9.getParentNode().removeChild(childNode9);
    max_delete++;
}
}
}
//remove a product using a function
String mobile = "mobile";
removeEmptyNodes(doc, mobile);
// write the content into xml file
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
```

Εικόνα 75: Διαγραφή κινητού τηλεφώνου από το αρχείο mobiles_XML.xml

UPDATE ORDER

Μόλις φτάσαμε στο βασικότερο κομμάτι του συστήματος μας. Το πώς δηλαδή μπορούμε να χρησιμοποιήσουμε όλα αυτά τα αρχεία που δημιουργήσαμε για να θεωρήσουμε μια παραγγελία ότι είναι τελειωμένη(**Εικόνα : 76**).

Η λογική εδώ είναι να χρησιμοποιήσουμε τα αρχεία **unfinished_mobiles_XML.xml**, **unfinished_laptops_XML.xml** και **unfinished_televisions.XML.xml**

```
to_finish_order.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        /* WE WILL USE FUNCTIONS IN ORDER TO SIMPLIFY THE CODE WE WANT TO BE READABLE PLUS EFFICIENCY
        FILLING process_order_laptops_XML.xml, process_order_mobiles_XML.xml ,process_televisions_laptops_X
        REMOVING these products from shopping basket's files in order he customer to re-buy products
        these new process files have an extra field ,beause we need to know how much is taken to finish an
        String customer_username = finish_order_text1.getText();
        String customer_time_to_finish = finish_order_text2.getText();
        try {
            removeLaptopsBasketFillProccessLaptopFile(customer_username, customer_time_to_finish);
            removeMobilesBasketFillProccessMobileFile(customer_username, customer_time_to_finish);
            removeTelevisionsBasketFillProccessTelevisionFile(customer_username, customer_time_to_finish);
        } catch (ParserConfigurationException | SAXException | IOException | TransformerException ex) {
            Logger.getLogger(UpdateOrder.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

Εικόνα 76: Συναρτήσεις για θεώρηση μιας παραγγελίας ως ολοκληρωμένη

Εδώ χρησιμοποιούμε τρεις συναρτήσεις για να σημάνουμε μια πληροφορία έως ολοκληρωμένη. Παιρνόντας στις συναρτήσεις το username καθώς και τον χρόνο που υποτίθεται ότι τελείωσε η παραγγελία(απαιτείται απο την εκφώνηση της πτυχιακής εργασίας).

Ας δούμε όμως πως λειτουργούν οι συναρτήσεις. Θα περιγράψουμε **μια εξ αυτών** μιας και είναι παρόμοιες. Το μόνο που αλλάζει είναι ο τρόπος πρόσβασης στα αρχεία καθώς δεν διαθέτουν τα προϊόντα μας τα ίδια πεδία(elements)

Όμως πρέπει να εκτελεστούν και οι τρεις συναρτήσεις για να θεωρηθεί ότι μια παραγγελία έχει ολοκληρωθεί. Αρχικά ανοίγουμε τη συνάρτηση που αφορά τα laptops. Ανοίγουμε το αρχείο unfinished_order_laptops_XML.xml

Κατόπιν, εισάγουμε τα λαπτοπ που έχει επιλέξει ένας πελάτης που ήδη είναι συνδεδεμένος στο σύστημα(ή τα κινητά ή τις τηλεοράσεις ή όλα μαζί. Εξαρτάται τι έχει προσθεσει ο πελάτης στο καλάθι αγορών), στα αρχεία process_order_laptop_XML.xml,

process_order_mobiles_XML.xml, process_order_television_XML.xml με τη διαφορά ότι διαθέτουμε extra πεδίο το time finished (*Εικόνα : 77*)

```
customer_laptops_lname.appendChild(doc4.createTextNode(laptop_lname_get));
laptop_v.appendChild(customer_laptops_lname);

Element customer_laptops_username = doc4.createElement("customer_username");
customer_laptops_username.appendChild(doc4.createTextNode(laptop_username_get));
laptop_v.appendChild(customer_laptops_username);

//the extra node for knowing when a order is finished
Element time_finished = doc4.createElement("time_finished");
time_finished.appendChild(doc4.createTextNode(time_to_finish_order_get));
laptop_v.appendChild(time_finished);

Element laptops_v;
laptops_v = (Element) doc4.getElementsByTagName("laptops").item(0);
laptops_v.appendChild(laptop_v);
TransformerFactory transformerFactory4 = TransformerFactory.newInstance();
Transformer transformer4 = transformerFactory4.newTransformer();

DOMSource source4 = new DOMSource(doc4);
StreamResult result4 = new StreamResult(new File("process_order_laptops_XML.xml"));
transformer4.transform(source4, result4);
}
} catch (SAXException | IOException | ParserConfigurationException | TransformerException ex) {
    Logger.getLogger(UpdateOrder.class.getName()).log(Level.SEVERE, null, ex);
}
//CALL A REMOVE FUNCTION FOR DELETING LAPTOPS FROM SHOPPING BASKET
removeLaptopsFromShoppingBasket(customer_username_get);
```

Εικόνα 77: Εισαγωγή των laptops απο το unfinished_order στο finished_order αρχείο

Τα αρχεία τύπου process_order δεν είναι κενά καθώς περιέχουν test entries που δημιουργήθηκαν με την συνάρτηση TestEntry().

Εκτός αυτού, καλείται η συνάρτηση **removeLaptopsFromShoppingBasket** διότι θέλουμε ο πελάτης να μπορεί να ξαναγράψει στο αρχείο unfinished_laptops_XML.xml (το ίδιο ισχύει και για τα άλλα αρχεία τύπου unfinished_order)

```

DocumentBuilder dBuilder0 = abFactory0.newDocumentBuilder();
Document doc0 = dBuilder0.parse(file0);
doc0.getDocumentElement().normalize();
NodeList nList0 = doc0.getElementsByTagName("laptop");
for (int i = 0; i < nList0.getLength(); i++) {
    NodeList childList = nList0.item(i).getChildNodes();
    Node childNode0 = childList.item(0);
    Node childNode1 = childList.item(1);
    Node childNode2 = childList.item(2);
    Node childNode3 = childList.item(3);
    Node childNode4 = childList.item(4);
    Node childNode5 = childList.item(5);
    Node childNode6 = childList.item(6);
    Node childNode7 = childList.item(7);
    Node childNode8 = childList.item(8);
    Node childNode9 = childList.item(9);
    Node childNode10 = childList.item(10);
    Node childNode11 = childList.item(11);
    //getting values using username
    if (childNode11.getTextContent().equals(customer_username_get)) {
        childNode0.getParentNode().removeChild(childNode0);
        childNode1.getParentNode().removeChild(childNode1);
        childNode2.getParentNode().removeChild(childNode2);
        childNode3.getParentNode().removeChild(childNode3);
        childNode4.getParentNode().removeChild(childNode4);
        childNode5.getParentNode().removeChild(childNode5);
        childNode6.getParentNode().removeChild(childNode6);
        childNode7.getParentNode().removeChild(childNode7);
        childNode8.getParentNode().removeChild(childNode8);
        childNode9.getParentNode().removeChild(childNode9);
    }
}

```

Εικόνα 78: Αφαίρεση laptops από το αρχείο unfinished_order_laptops_XML.xml

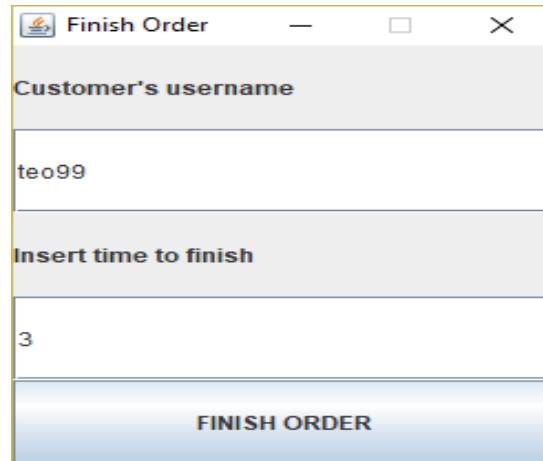
Επομένως προσθέτουμε τα στοιχεία των αρχείων unfinished_order στα αρχεία process_order σύν ότι διαγράφουμε ανάλογα με τον πελάτη που έχει εισέλθει στο σύστημα τα κατάλληλα προϊόντα.

Για να κάνουμε update order με το κατάλληλο event εμφανίζεται το παρακάτω JFrame(*Εικόνα : 79*)

The image shows a Java Swing window titled "Finish Order". It has a standard title bar with minimize, maximize, and close buttons. The window contains three text input fields stacked vertically. The first field is labeled "Customer's username", the second is labeled "Insert time to finish", and the third is a large blue button with the text "FINISH ORDER" in white capital letters.

Εικόνα 79: JFrame για finish order

Για παράδειγμα εάν θέλουμε να αφαιρέσουμε τα προϊόντα laptops, mobiles, televisions του πελάτη με username = teo 99 και να θέσουμε έως ημερομηνία ολοκλήρωσης της παραγγελίας τότε (**Εικόνα : 80**)



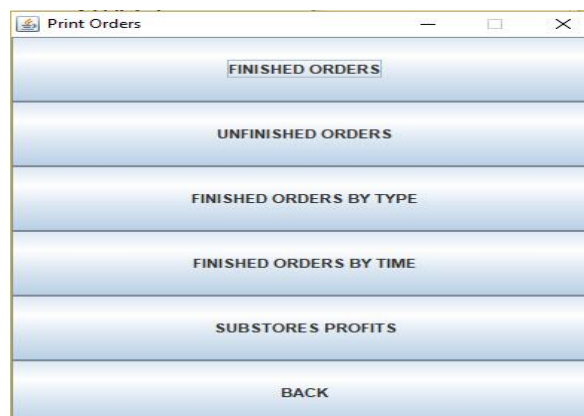
Εικόνα 80: Διαγραφή προϊόντων απο το καλάθι αγορών

Θα δούμε για παράδειγμα οτι προστέθηκαν δύο laptops στο process_order_laptops.XML που είναι αυτά που υποτίθεται έχει αγοράσει ο πελάτης. Αυτά τα λαπτοπ έχουν σβηστεί απο το αρχείο unfinished_order_laptops_XML.xml . Ομοίως για το εαν έχει αγοράσει κινητά και τηλεοράσεις.

PRINT ORDERS

Εδώ τα πράγματα είναι πολύ πιο απλά. Με κατάλληλη χρήση JTextAreas και JScrollPane θα προσπαθήσουμε να εκτυπώσουμε με διάφορους τρόπους τις παραγγελίες κ.α

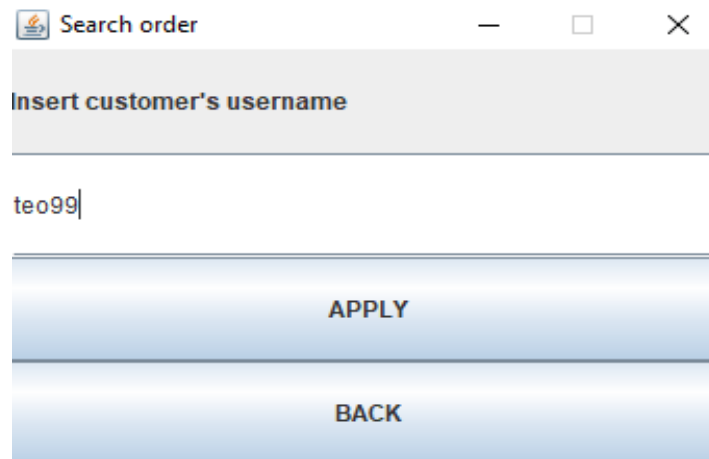
Εαν κάνουμε κλικ στο κουμπί print orders θα μας εμφανίσει το παρακάτω JFrame(**Εικόνα : 81**)



Εικόνα 81: Βασικό JFrame για εκτύπωση παραγγελιών και κέρδη υποκαταστημάτων

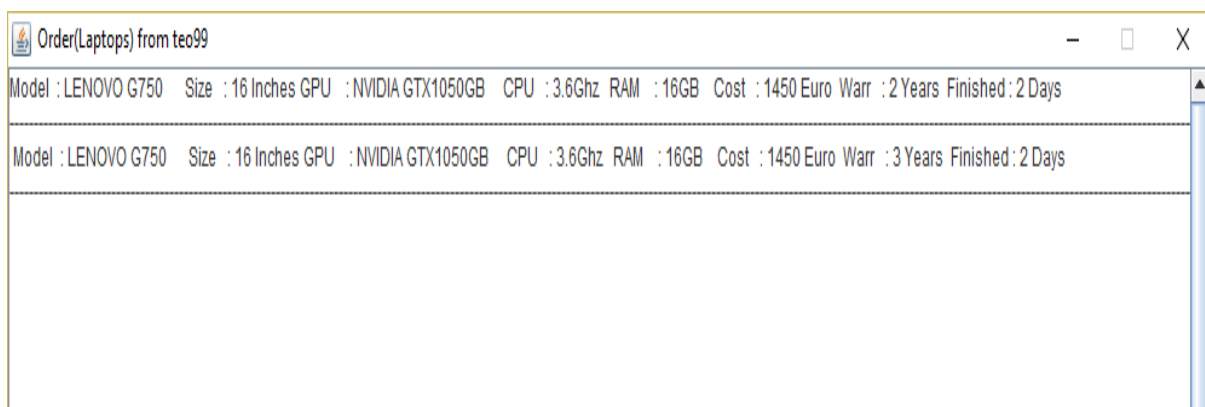
Αν επιλέξουμε το finished_orders τότε θα μας εμφανιστούν τρία JTextAreas που περιγράφουν την εκτύπωση και των τριών αρχείων unfinished_order .

Ομως δε γίνεται ένας χρήστης να εκτυπώσει όλες τις παραγγελίες το συστήματος! Για τον λόγο αυτό βάλαμε ακόμη ένα JFrame το οποίο μας δείχνει την εκτύπωση ανα customer username.(**Εικόνα : 82**)



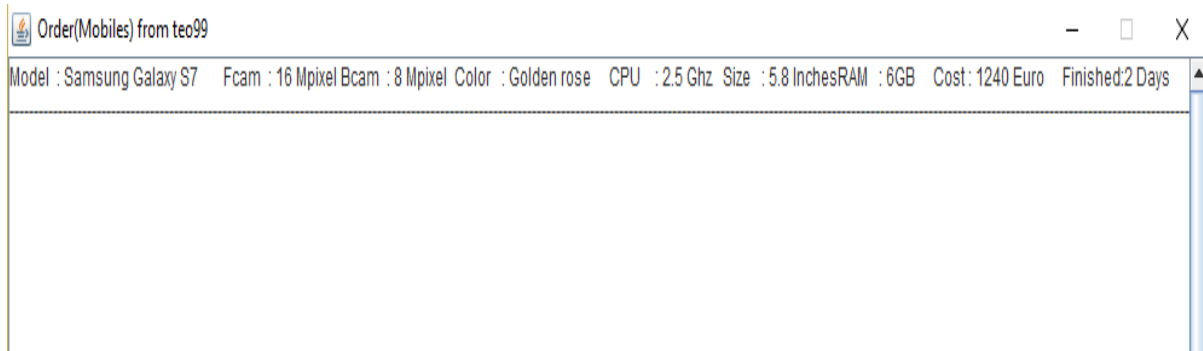
Εικόνα 82: Εύρεση παραγγελίας ανα username

Εαν για παράδειγμα επιλέξουμε ως username john92 τότε ας δούμε τα laptops που έχει αγοράσει ο teo99(**Εικόνα : 83**)



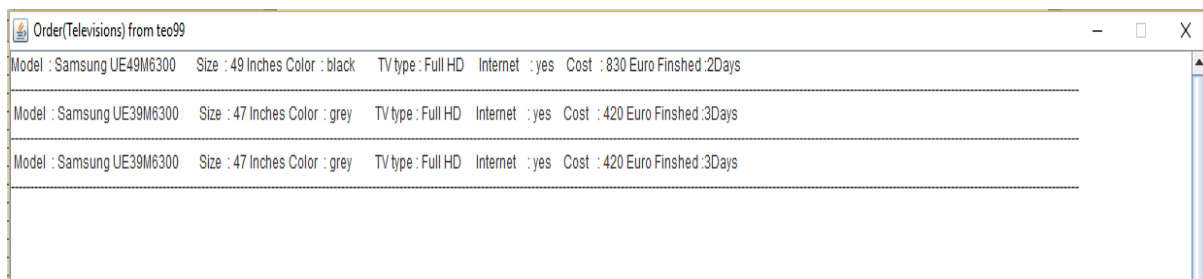
Εικόνα 83: Εκτύπωση παραγγελίας (laptops)

Τα κινητά τηλέφωνα (**Εικόνα : 84**)



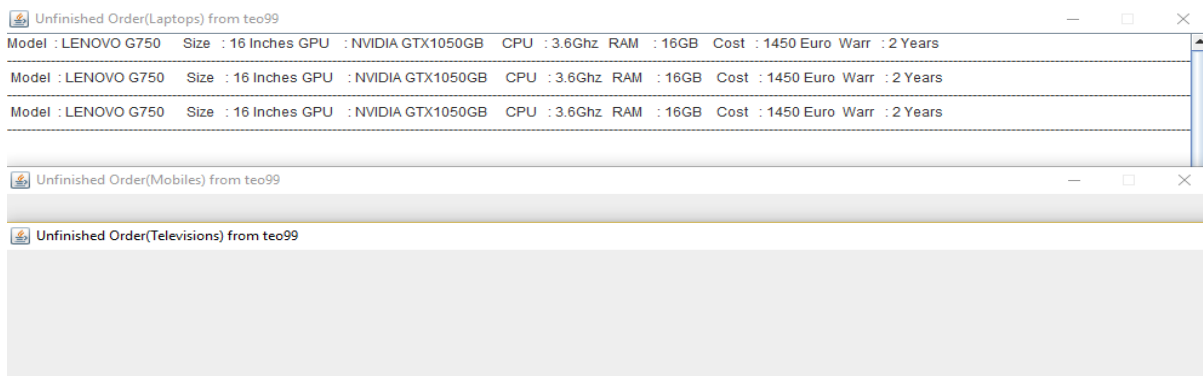
Εικόνα 84: Εκτύπωση παραγγελίας (mobiles)

Και τέλος οι τηλεοράσεις (**Εικόνα : 85**)



Εικόνα 85: Εκτύπωση παραγγελίας (televisions)

Αν κάνουμε ως επιλογή το να εκτυπώσουμε τις μη τελειωμένες παραγγελίες πολύ απλά είναι η εκτύπωση των unfinished_order προϊόντων. Δηλαδή εκτυπώνει το καλάθι αγορών του πελάτη, δηλαδή μη τελειωμένες παραγγελίες . Αν θελήσουμε να εκτυπώσουμε το καλάθι αγορών του πελάτη με username teo 99, τότε για τα laptops, mobiles και televisions έχουμε (**Εικόνα : 86**)

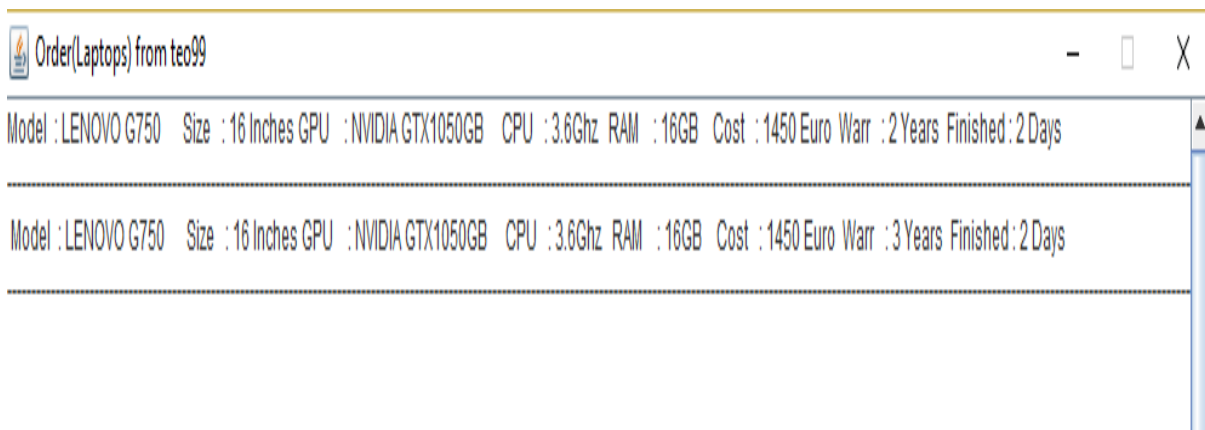


Εικόνα 86: Εκτύπωση καλαθιού αγοράς πελάτη

Παρατηρούμε οτι τα αρχεία των τηλεοράσεων και των κινητών είναι κενά. Αυτό συμβαίνει διότι ο admin για να σημάνει μια παραγγελία ως τελειωμένη χρησιμοποιεί και τον χρόνο που τελείωσε μια παραγγελία.

Οποτε οι παραγγελίες που δεν έχουν σημειωθεί ως τελειωμένες, δηλαδή να διαθέτουν χρόνο ολοκλήρωσης δεν αφαιρούνται. Αρα σε ένα δεύτερο run του προγράμματος μας εμείς προσθέσαμε και άλλα προϊόντα στο καλάθι τα οποία από την πλευρά του admin δεν τα ολοκληρώσαμε ως παραγγελία.

Με όμοιο τρόπο μπορούμε να εκτυπώσουμε τις παραγγελίες που ολοκληρώθηκαν ανά είδος προϊόντος (**Εικόνα : 87**)



Εικόνα 87: Εκτύπωση ανά είδος προϊόντων

Για να εκτυπώσουμε τα κινητά τηλέφωνα που βρίσκονται στις τελειωμένες παραγγελίες πολύ απλά ανοίγουμε το process_mobiles_XML.xml αρχείο κάνουμε iterate σε όλους τους κόμβους και κάνουμε printing εκτός από το test entry (**Εικόνα : 88**) . Χωρίς μεγάλες διαφορές εκτυπώνουμε και τα άλλα δυο JTextAreas.

```

if (!childNodes0.getTextContent().equals("test_model")) {
    String printing_result
        = "Model : " + childNode0.getTextContent() + "      "
        + "Fcam : " + childNode1.getTextContent() + " Mpixel "
        + "Bcam : " + childNode2.getTextContent() + " Mpixel "
        + "Color  : " + childNode3.getTextContent() + "      "
        + "CPU    : " + childNode4.getTextContent() + " Ghz   "
        + "Size   : " + childNode5.getTextContent() + " Inches"
        + "RAM    : " + childNode6.getTextContent() + "GB     "
        + "Cost   : " + childNode7.getTextContent() + " Euro   "
        + "Finished:" + childNode13.getTextContent() + " Days" + "\n"
        + "-----"
        + "-----"
        + "-----"
        + "-----"
        + "\n ";
    printing_category_areal.setEditable(false);
    JScrollPane scrollPanel = new JScrollPane(printing_category_areal);
    mobiles_printing.add(scrollPanel);
    printing_category_areal.append(printing_result);
}

```

Εικόνα 88: Εκτύπωση κινητών τηλεφώνων - ολοκληρωμένη παραγγελία

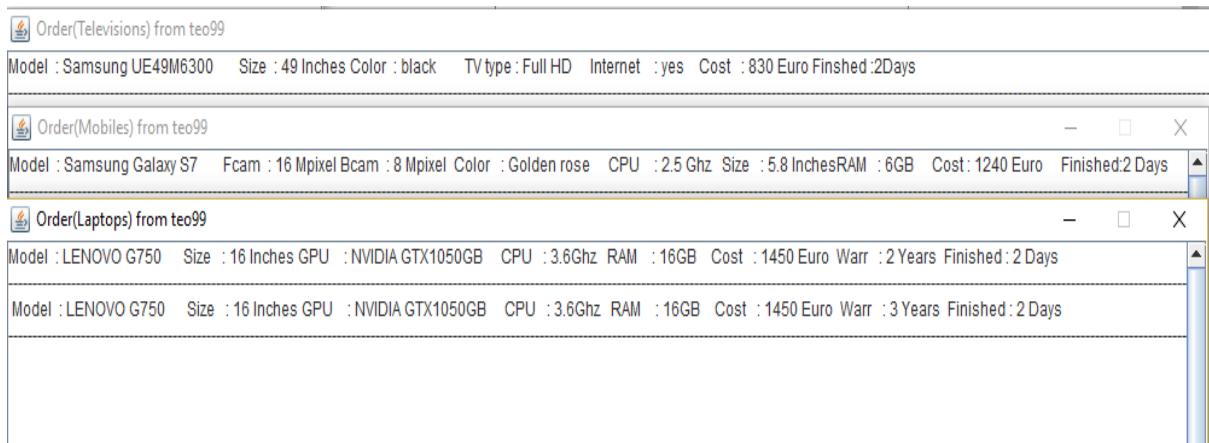
Αν θελήσουμε να εκτυπώσουμε με βάση το χρόνο που πήρε να τελειώσει μια παραγγελία(**Εικόνα : 89**)

The screenshot shows a standard Java Swing window with a title bar containing a printer icon, the text 'Print by finished time', and standard window control buttons (minimize, maximize, close). The main content area has a light gray background and contains the following elements from top to bottom:

- A label 'Insert customer's username' above a rectangular text input field.
- A label 'Insert time finished' above another rectangular text input field.
- A blue button with white text labeled 'APPLY'.
- A blue button with white text labeled 'BACK'.

Εικόνα 89: Εκτύπωση σε σχέση με το χρόνο ολοκλήρωσης της παραγγελίας

Βλέπουμε οτι εαν θέσουμε χρόνο ολοκλήρωσης τις δύο ημέρες τότε (**Εικόνα : 90**)



Εικόνα 90: Εκτύπωση σύμφωνα με το χρόνο ολοκλήρωσης παραγγελίας

Ο κώδικας δεν διαφέρει και πολυ. Το μόνο που αλλάζει είναι οτι κάνουμε αναζήτηση για κάθε προϊόν ξεχωριστά, ανάλογα με το element που αφορά τον χρόνο ολοκλήρωσης μιας παραγγελίας.

Το μόνο που έμεινε να εξηγήσουμε είναι το πώς θα πάρουμε τα κέρδη κάθε υποκαταστήματος κάθε εταιρείας και να τα εκτυπώσουμε στην οθόνη. Ας δούμε το παρακάτω κομάτι του κώδικα (**Εικόνα : 91**)

```
finished_orders_time_finish.addActionListener(new ActionListener() {  
    //ACTIONLISTENER FOR PRINTING FINAL PROFITS OF EACH SUBSTORE USING FUN  
    print_orders_final PROFITS.addActionListener(new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent ae) {  
            printAllProfits();  
        }  
    }  
});
```

Εικόνα 91: Κλήση της συνάρτησης printAllProfits()

Αρχικά καλούμε τη συνάρτηση printAllprofits(). Μολις κληθεί κατευθείαν ανοίγει το αρχείο substores_XML.xml και παίρνουμε το όνομα κάθε ενός απο τα υποκαταστήματα που υπάρχουν εκεί.

```

try {
    File file0 = new File("substores_XML.xml");
    // Create builders
    DocumentBuilderFactory dbFactory0 = DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder0 = dbFactory0.newDocumentBuilder();
    Document doc0 = dBuilder0.parse(file0);
    doc0.getDocumentElement().normalize();
    float final_profits = 0;
    StringBuilder sbuilder_substore_names_profits = new StringBuilder(600);
    //checking if the appropriate count of products exists in our system
    NodeList nList0 = doc0.getElementsByTagName("substore");
    sbuilder_substore_names_profits.append("Substore Name & Profits: ");
    for (int i = 0; i < nList0.getLength(); i++) {
        NodeList childList = nList0.item(i).getChildNodes();
        //get substore name from substores_XML.xml file
        Node childNode0 = childList.item(0);
        String substore_name = childNode0.getTextContent();
        /*Every time we find a substore name from substores_XML.xml IF exists in our process_files
        then we call a function in order to get the profits */
        final_profits = getProfitsWithGivenSubstoreName(substore_name);
        //we can use String Builder. So we can append substore_names and profits, for printing
        if (final_profits > 0) {
            //append profits to StringBuilder for substore_names
            sbuilder_substore_names_profits.append(substore_name);
            sbuilder_substore_names_profits.append(" ");
            //convert float to string
            String final_profits_string = Float.toString(final_profits);
            //append profits to StringBuilder for substore's final profits
            sbuilder_substore_names_profits.append(final_profits_string);
            sbuilder_substore_names_profits.append(" Euros ");
        }
    }
}

```

Εικόνα 92: Κλήση της συνάρτησης getProfitsWithGivenSubstoreName

Κατόπιν μέσω του ονομάτος που λαμβάνουμε κάθε φορά που διαβάζουμε το επόμενο element στο αρχείο αυτό, δίνεται ως όρισμα στη συνάρτηση getProfitsWithGivenSubstoreName. Η συνάρτηση έχει τιμή επιστροφής float. Τί ακριβώς επιτελεί όμως αυτή η συνάρτηση;(*Εικόνα : 93*)

```

File file = new File("process_order_laptops_XML.xml");
DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(file);
doc.getDocumentElement().normalize();
NodeList nList = doc.getElementsByTagName("laptop");
//we want to calculate the cost of a given substore's laptops
//also we need some counters
float final_laptops_cost = 0;
for (int i = 0; i < nList.getLength(); i++) {
    NodeList childList = nList.item(i).getChildNodes();
    Node childNode0 = childList.item(0);
    Node childNode7 = childList.item(7);
    Node childNode6 = childList.item(5);
    if (!childNode0.getTextContent().equals("test_model")) {
        if (childNode7.getTextContent().equals(substore_name)) {
            //convert string to float
            String laptop_cost = childNode6.getTextContent();
            float given_substore_laptops_costs = Float.parseFloat(laptop_cost);
            final_laptops_cost = final_laptops_cost + given_substore_laptops_costs;
        }
    }
}
}

```

Εικόνα 93: Υπολογισμός τελικού κέρδους για τα laptops

Ανοίγει το αρχείο process_order_laptops_XML.xml και αναλόγως το substore που δώσαμε παίρνει τις τιμές των elements που περιέχουν το κόστος του κάθε προϊόντος τύπου λαπτοπ.

Καθώς οι τιμές στο XML είναι string χρειαζόμαστε να τα μετατρέψουμε σε float και με ένα counter πήραμε το συνολικό κόστος των laptops. Εάν ένα υποκατάστημα δεν έχει τελειωμένη παραγγελία η συνάρτηση να μην περνάει από τα αρχεία αλλά βγάζει μηδενικό κέρδος.

Οπότε κάθε φορά, για κάθε διαφορετικό substore name που υπάρχει στο αρχείο substores_XML.xml, χρησιμοποιείται η συνάρτηση αυτή, για να υπολογίσουμε τα κόστη των laptops. Ομοίως για τα υπόλοιπα προϊόντα (*Εικόνα : 94*)

```
File file1 = new File("process_order_mobiles_XML.xml");
// Create builders
DocumentBuilderFactory dbFactory1 = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder1 = dbFactory1.newDocumentBuilder();
Document doc1 = dBuilder1.parse(file1);
doc1.getDocumentElement().normalize();
float final_mobiles_cost = 0;
NodeList nList2 = doc1.getElementsByTagName("mobile");
for (int i = 0; i < nList2.getLength(); i++) {
    NodeList childList = nList2.item(i).getChildNodes();
    Node childNode0 = childList.item(0);
    Node childNode7 = childList.item(7);
    Node childNode8 = childList.item(8); //substore name

    if (!childNode0.getTextContent().equals("test_model")) {
        if (childNode8.getTextContent().equals(substore_name)) {
            //convert string to float
            String mobile_cost = childNode7.getTextContent();
            float given_substore_mobiles_costs = Float.parseFloat(mobile_cost);
            final_mobiles_cost = final_mobiles_cost + given_substore_mobiles_costs;
        }
    }
}
```

Εικόνα 94:Υπολογισμός τελικού κέρδους για τα mobiles

Και για τις τηλεοράσεις (*Εικόνα : 95*)

```

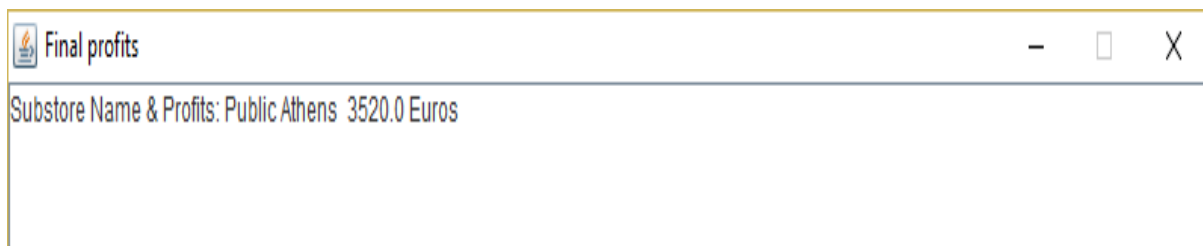
File file2 = new File("process_order_televisions_XML.xml");
// Create builders
DocumentBuilderFactory dbFactory2 = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder2 = dbFactory2.newDocumentBuilder();
Document doc2 = dBuilder2.parse(file2);
doc2.getDocumentElement().normalize();
float final_televisions_cost = 0;
NodeList nList3 = doc2.getElementsByTagName("television");
for (int i = 0; i < nList3.getLength(); i++) {
    NodeList childList = nList3.item(i).getChildNodes();
    Node childNode0 = childList.item(0);
    Node childNode5 = childList.item(5);
    Node childNode6 = childList.item(6);
    if (!childNode0.getTextContent().equals("test_model")) {
        if (childNode6.getTextContent().equals(substore_name)) {
            //convert string to float
            String television_cost = childNode5.getTextContent();
            float given_substore_televisions_costs = Float.parseFloat(television_cost);
            final_televisions_cost = final_televisions_cost + given_substore_televisions_costs;
        }
    }
}
float final_profits = final_laptops_cost + final_mobiles_cost + final_televisions_cost;
return final_profits;

```

Εικόνα 95: Υπολογισμός τελικού κέρδους για τις televisions

Στη συνέχεια επιστρέφεται το συνολικό κόστος mobiles, laptops, television για κάθε substore name που δέχεται ως όρισμα η συνάρτηση **getProfitsWithGivenSubstoreName()**.

Στην συνάτηση **printAllProfits()** εάν το κόστος που επιστραφεί είναι θετικό (για παράδειγμα εαν ενα substore δεν εχει πουλήσει ουτε κινητά ουτε τηλεοράσεις αλλα ουτε και λαπτοπ τοτε ειναι μηδεν) με χρήση **StringBuilder** και κατάλληλων μορφοποιήσεων δημιουργούμε ενα string το οποίο θα εκτυπώσουμε στην οθόνη με JTextArea και JScrollPane(**Εικόνα : 96**)



Εικόνα 96:Κέρδη υποκαταστημάτων

Ξεκάθαρα φαίνεται οτι απο όλα τα υποκαταστήματα μόνο το υποκατάστημα Public Athens έχει πουλήσει προϊόντα(αφού κάναμε παραγγελία με τον χρήστη teo99 , στα προϊόντα και

είχε αποθέματα εκείνη τη στιγμή το υποκατάστημα αυτό) κόστους 3520 ευρώ. Ενώ ο κώδικας (*Εικόνα : 97*)

```
        then we call a function in order to get the profits */
        final_profits = getProfitsWithGivenSubstoreName(substore_name);
        //we can use String Builder. So we can append substore_names and profits, fo:
        if (final_profits > 0) {
            //append profits to StringBuilder for substore_names
            sbuilder_substore_names_profits.append(substore_name);
            sbuilder_substore_names_profits.append(" ");
            //convert float to string
            String final_profits_string = Float.toString(final_profits);
            //append profits to StringBuilder for substore's final profits
            sbuilder_substore_names_profits.append(final_profits_string);
            sbuilder_substore_names_profits.append(" Euros ");
        }
    }
    //PRINTING IN JTextArea using JScrollPane and StringBuilders that we have created
    JFrame profits_print_frame = new JFrame("Final profits");
    profits_print_frame.setSize(800, 600);
    profits_print_frame.setLayout(new BorderLayout());
    profits_print_frame.setVisible(true);
    profits_print_frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    profits_print_frame.setResizable(false);
    JTextArea printing_profits_area = new JTextArea(20, 20);
    printing_profits_area.setPreferredSize(new Dimension(400, 400));
    String profits_printing = sbuilder_substore_names_profits.toString() + "\n";
    printing_profits_area.setEditable(false);
    //laptops_printing.add(printing_category_area);
    JScrollPane scrollPane = new JScrollPane(printing_profits_area);
    profits_print_frame.add(scrollPane);
    printing_profits_area.append(profits_printing);
} catch (ParserConfigurationException | SAXException | IOException ex) {
    Logger.getLogger(PrintOrders.class.getName()).log(Level.SEVERE, null, ex);
}
```

Εικόνα 97: Εκτύπωση κερδών υποκαταστημάτων

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην σημερινή εποχή στην οποία ζούμε, τα άτομα γύρω μας προσπαθούν ολο ένα και περισσότερο να εξοικονομήσουν ελεύθερο χρόνο για να τον περάσουν με τα αγαπημένα τους πρόσωπα. Απο την άλλη πλευρά, επιθυμούν να μπορούν να αγοράζουν φτηνά τα προϊόντα που την εκάστοτε στιγμή χρειάζονται.

Το ηλεκτρονικό εμπόριο, παρά τα μειονεκτήματά του, διαθέτει αυτά τα δύο δυνατά χαρακτηριστικά. Μπορείς να αγοράζεις γρήγορα, εύκολα και ευέλικτα και προπάντων οικονομικά. Κάνοντας είσοδο σε ένα τέτοιο σύστημα το μόνο που χρειάζεται συνήθως είναι να βάλεις τον κωδικό της πιστωτικής ή χρεωστικής σου κάρτας, αν και ακόμη υποστηρίζεται η μέθοδος της αντικαταβολής.

Μπορούμε να δούμε μια πληθώρα προϊόντων με μόλις ένα κλικ, χωρίς να χρειαστεί να χάσουμε χρόνο σε μαγαζιά που διαθέτουν φυσική τοποθεσία. Φυσικά οτιδήποτε βρίσκεται στο ίντερνετ είναι ευπαθές σε επιθέσεις. Θεωρούμε ότι μελλοντικά, όλα αυτά τα συστήματα και οι πλατφόρμες ηλεκτρονικού εμπορίου θα είναι πολύ πιο ασφαλείς.

Για την παρούσα πτυχιακή εργασία χρησιμοποιήσαμε την γλώσσα προγραμματισμού Java σε συνδυασμό με την γλώσσα XML. Επιλέξαμε XML διότι είναι στο διαδίκτυο η επικρατούσα τεχνολογία αποθήκευσης δεδομένων είναι τα XML αρχεία. Διαθέτει το μοναδικό χαρακτηριστικό ότι μπορεί να γίνει πολύ εύκολα αντιληπτή από τον άνθρωπο αλλά και από τις μηχανές, καθιστώντας την εύκολη στη χρήση και στην εκμάθηση.

Summary

In our days, the people around us are increasingly trying to save time to spend with their loved ones. On the other hand, they want to be able to buy the cheap products they need at any given time.

E-commerce, despite its disadvantages, has these two possible features. You can buy quickly, easily and flexibly and above all financially. By entering such a system, all you need is usually to enter your credit or debit card code, even though the method of cash on delivery is still supported.

We can see a lot of products with just one click, without having to waste time in shops that have a natural location. Of course anything on the internet is vulnerable to attacks. We believe that in the future, all these e-commerce platforms and platforms will be much safer.

For this project we used the Java programming language in combination with the XML language. We chose XML because it is on the Internet the prevailing data storage technology is XML files. It has the unique feature that it can be easily perceived by man and machine, making it easy to use and learn.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Blinder, A., S; Baumol, W., J, Gale, C., L. (June 2001). 11: Monopoly". Microeconomics: Principles and Policy. Thomson South-Western. p. 212.
- Bloomidea. Types of e-commerce. Retrieved from [Here](#) [accessed 1 December 2017]
- Borden, Neil H. (1965). The Concept of the Marketing Mix. In Schwartz, George. Science in marketing. Wiley marketing series. Wiley. p. 286. [accessed 1 December 2017]

- Bray, T., Frankston, C.; Malhotra, A., eds. (1998). Document Content Description for XML: Submission to the World Wide Web Consortium. World Wide Web Consortium.
- Bromley, D., W. (1991). Environment and Economy: Property Rights and Public Policy. Cambridge, MA: Blackwell Pub.
- Chamberlin, E. H. (1937). Monopolistic or Imperfect Competition?. The Quarterly Journal of Economics. 51 (4): 557–580
- Chisholm, H., ed. (1911). Commerce.- Encyclopædia Britannica. 6 (11th ed.). Cambridge University Press
- David R.(1971). Principles of Political Economy and Taxation . Pelican Books: London
- Donaldson, S., A. Federal(2007). Income Taxation of Individuals: Cases, Problems and Materials. 2th Ed. St. Paul: Thomson West, pp. 200.
- Frank, R., Microeconomics and Behavior 7th ed. McGraw-Hill: USA
- Gans, J., King, S., Stonecash, R., Mankiw, N. G. (2003). Principles of Economics. Thomson Learning.
- Gerard D.(1927). Theory of Value: An Axiomatic Analysis of Economic Equilibrium. Yale University Press, New Haven CT
- Goodwin, N., Nelson, J., Ackerman, F., Weisskopf, T (2009). Microeconomics in Context .2 Ed. Sharpe: USA
- Goodwin, N., Nelson, J., Ackerman, F., Weisskopf, T. (2009). Microeconomics in Context 2th Ed. Sharpe. p. 289
- Heyne, P., Boettke, P., J., Prychitko, D., L. (2014). The Economic Way of Thinking. 13th Ed. Pearson. pp. 130–132.
- Huczynski, A. et al. (2001): Organizational Behavior, 4th Edition, Harlow : USA
- J. Downes, J. E. Goodman (2003). Dictionary of Finance & Investment Terms, Barron's Financial Guides
- Johnston, R., Clark, G. (2008). Service Operations Management: Improving Service Delivery. Financial Times: Prentice Hall
- Klein, D., B., Robinson, J.(2011). Property: A Bundle of Rights? Prologue to the Symposium. Econ Journal Watch 8(3): 193–204
- Kotler, P., Armstrong, G. (1989). Principles of Marketing, 4th Ed. Prentice-Hall, Inc. pp. 639

- Kotler, P., Armstrong, G., Brown, L., Adam, S. (2006) Marketing, 7th Ed. Pearson Education Australia/Prentice Hall.
- Krugman & Wells(2009). Microeconomics, 2 Ed. Worth: USA
- Krugman, P., Obstfeld, M. (2008). International Economics: Theory and Policy. Addison-Wesley
- Marshall, A. (1890). Principles of Economics. Retrieved from [Here](#) [accessed 1 December 2017]
- Osborne, M., J., Rubinstein, A. (1994). Chapter 6: Extensive Games with Perfect Information. A Course in Game Theory. Cambridge M.A.: The MIT Press.
- O'Sullivan, A., Sheffrin, Steven M. (2004). Economics: Principles in action. Pearson Prentice Hall
- O'Sullivan, A., Sheffrin, Steven M. (2004). Economics: Principles in action. Pearson Prentice Hall
- Roberts, J. (1987). Perfectly and imperfectly competitive markets, The New Palgrave: A Dictionary of Economics. v. 3, pp. 837–41. [accessed 1 December 2017]
- Samuelson & Nordhaus(2001). Microeconomics. 17th Ed. McGraw-Hill 2001, pp. 184.
- W3Schools. DTD examples. Retrieved from [Here](#) [accessed 1 December 2017]
- W3schools. XML examples. Retrieved from [Here](#) [accessed 1 December 2017]
- W3Schools. XML Schema examples. Retrieved from [Here](#) [accessed 1 December 2017]
- W3Schools. XPATH examples. Retrieved from [Here](#) [accessed 1 December 2017]
- W3Schools. XSLT examples. Retrieved from [Here](#) [accessed 1 December 2017]
- Webster, E., Jensen, P., H. (2006). Investment in Intangible Capital: An Enterprise Perspective. The Economic Record, Vol. 82, No. 256, pp. 82-96.
- World Wide Web Consortium. Extensible Markup Language (XML) 1.1 . 2th Ed. Retrieved from [Here](#) [accessed 1 December 2017]
- World Wide Web Consortium. Extensible Markup Language (XML). Retrieved from [Here](#) [accessed 1 December 2017]
- Yutaka , K., Nobuyoshi Y., Sadayoshi T.(2005). Global information technology and competitive financial alliances. Idea Group Inc.