ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.Ι ΗΡΑΚΛΕΙΟΥ ΚΡΗΤΗΣ

E-SHOP ΓΙΑ ΥΠΟΛΟΓΙΣΤΕΣ & ΗΛΕΚΤΡΟΝΙΚΑ ΕΙΔΗ

**Σταυρουλάκης Βασίλειος ΑΜ 3428**

Ηράκλειο, 2018

Table of Contents

[ΕΥΧΑΡΙΣΤΙΕΣ 5](#_Toc517399360)

[ABSTRACT 6](#_Toc517399361)

[1. ΠΕΡΙΛΗΨΗ 7](#_Toc517399362)

[2. ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΚΑΤΑΣΤΗΜΑ( E – SHOP) 8](#_Toc517399363)

[2.1 ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ 8](#_Toc517399364)

[2.2 Η ΛΕΙΤΟΥΡΓΙΑ ΕΝΟΣ ESHOP & ΓΝΩΣΤΕΣ ΠΛΑΤΦΟΡΜΕΣ ΠΑΡΟΧΗΣ ΠΡΟΙΟΝΤΩΝ ΚΑΙ ΥΠΗΡΕΣΙΩΝ 10](#_Toc517399365)

[2.2.1 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ 10](#_Toc517399366)

[2.2.2 AMAZOON & EBAY 10](#_Toc517399367)

[3. ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ NETBEANS & ΛΟΙΠΩΝ ΕΡΓΑΛΕΙΩΝ 12](#_Toc517399368)

[3.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA RUNTIME 12](#_Toc517399369)

[3.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS IDE 12](#_Toc517399370)

[3.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS 8.2 IDE 13](#_Toc517399371)

[4. ΠΑΡΟΥΣΙΑΣΗ/RUNNING ΤΗΣ ΕΦΑΡΜΟΓΗΣ 16](#_Toc517399372)

[4.1 Η ΠΛΕΥΡΑ ΤΟΥ ADMIN ( ADMIN SIDE ) 16](#_Toc517399373)

[4.2.1 ΕΙΣΟΔΟΣ ΩΣ ADMIN ΣΤΟ ΣΥΣΤΗΜΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ 19](#_Toc517399374)

[4.2.1.1 ADD PRODUCTS 22](#_Toc517399375)

[4.2.1.2 PRINT PRODUCTS 28](#_Toc517399376)

[4.2.1.3 DELETE PRODUCTS 30](#_Toc517399377)

[4.2.1.4 PRINT CUSTOMERS 35](#_Toc517399378)

[4.2.1.5 ADD COMPANY 36](#_Toc517399379)

[4.2.1.6 REMOVE COMPANY & PRINT COMPANIES 39](#_Toc517399380)

[4.2.1.7 FINISH CUSTOMER ORDER 40](#_Toc517399381)

[4.2.1.8 PRINT FINISHED ORDERS 41](#_Toc517399382)

[4.2.1.9 LOGOUT 42](#_Toc517399383)

[4.3 Η ΠΛΕΥΡΑ ΤΟΥ ΠΕΛΑΤΗ ( CLIENT SIDE ) 42](#_Toc517399384)

[4.3.1 PRINT PRODUCTS 42](#_Toc517399385)

[4.3.2 SHOPPING BASKET 46](#_Toc517399386)

[4.3.3 LOGOUT 47](#_Toc517399387)

[5. ΜΕΛΛΟΝΤΙΚΟ ΕΡΓΟ( FUTURE WORK) 48](#_Toc517399388)

[5.1 Η ΧΡΗΣΗ ΤΗΣ SQL ΔΙΝΕΙ ΕΥΚΟΛΟΤΕΡΗ ΥΛΟΠΟΙΗΣΗ 48](#_Toc517399389)

[5.2 Η ΣΥΝΔΕΣΗ ΤΗΣ SQL ΜΕ ΤΗΝ JAVA 49](#_Toc517399390)

[5.3 SQL – JAVA ΥΠΕΡΩΤΗΜΑΤΑ(QUERIES) 50](#_Toc517399391)

[ΣΥΜΠΕΡΑΣΜΑΤΑ 51](#_Toc517399392)

[ΑΝΑΦΟΡΕΣ 52](#_Toc517399393)

**Table of Figures**

[Figure 1: H πλατφόρμα Amazoon 10](#_Toc516971660)

[Figure 2: Η πλατφόρμα ebay 11](#_Toc516971661)

[Figure 3: Η εγκατάσταση της JAVA 12](#_Toc516971662)

[Figure 4: Η εγκατάσταση της πλατφόρμας NETBEANS 8.2 IDE + JDK 13](#_Toc516971663)

[Figure 5: Αρχικό πάνελ του NETBEANS 14](#_Toc516971664)

[Figure 6: Η επιλογή File του NETBEANS 14](#_Toc516971665)

[Figure 7: Η επιλογή Edit του NETBEANS 15](#_Toc516971666)

[Figure 8: JFrame εισόδου και εγγραφής 16](#_Toc516971667)

[Figure 9: JFrame για εισοδο/εξοδο χρήστη απο σύστημα 16](#_Toc516971668)

[Figure 10: actionListener() για το κουμπί register 17](#_Toc516971669)

[Figure 11: CheckUser (a) 17](#_Toc516971670)

[Figure 12: RegisterClass 18](#_Toc516971671)

[Figure 13: CheckUser (b) 19](#_Toc516971672)

[Figure 14: Admin Panel 20](#_Toc516971673)

[Figure 15: admin main panel (a) 21](#_Toc516971674)

[Figure 16: admin main panel (b) 21](#_Toc516971675)

[Figure 17: Add Camera 22](#_Toc516971676)

[Figure 18: Add camera (b) 23](#_Toc516971677)

[Figure 19: Add camera (c) 23](#_Toc516971678)

[Figure 20: Add camera 24](#_Toc516971679)

[Figure 21: Αρχειο cameras.txt πριν την εισαγωγή νέων καμερών 24](#_Toc516971680)

[Figure 22: Φορμα add camera 25](#_Toc516971681)

[Figure 23: Αρχειο cameras.txt μετα την εισαγωγή (κομμάτι αυτού) 25](#_Toc516971682)

[Figure 24: Φορμα προσθήκης gaming console 26](#_Toc516971683)

[Figure 25: Αρχειο gaming\_consoles πριν την εισαγωγή κονσόλας 26](#_Toc516971684)

[Figure 26: Η προσθήκη μιας κονσόλας στο αρχείο 27](#_Toc516971685)

[Figure 27: To αρχειο gaming\_consoles μετα την προσθήκη κονσόλων 27](#_Toc516971686)

[Figure 28: Φόρτωση αρχείων για εκτύπωση(κομμάτι κώδικα) 28](#_Toc516971687)

[Figure 29: Εκτύπωση συνολικών προιόντων 29](#_Toc516971688)

[Figure 30: Εκτύπωση προιόντων (κομματι κώδικα) 30](#_Toc516971689)

[Figure 31: Κώδικας μετάβασης για διαγραφή κατάλληλου προιόντος 31](#_Toc516971690)

[Figure 32: Διαγραφή προιόντος(κεντρικό πανελ) 31](#_Toc516971691)

[Figure 33: Διαγραφή ενός υπολογιστή απο το αρχειο 32](#_Toc516971692)

[Figure 34: Ανακατασκευή του arrayList pcs\_list 32](#_Toc516971693)

[Figure 35: Διαγραφή ποσότητας προιόντων απο arrayList 33](#_Toc516971694)

[Figure 36: Αποθήκευση της νέας arrayList στο αρχειο 34](#_Toc516971695)

[Figure 37: Αρχειο personalComputers πρίν τη διαγραφή 34](#_Toc516971696)

[Figure 38: Αρχειο personalComputers μετα την διαγραφή 34](#_Toc516971697)

[Figure 39: Εκτύπωση πελατών(κομμάτι κώδικα) 35](#_Toc516971698)

[Figure 40: Εκτύπωση πελατών 36](#_Toc516971699)

[Figure 41: JFrame προσθήκης εταιρείας στο κατάστημα 36](#_Toc516971700)

[Figure 42: Ανακατασκευή της arrayList των εταιρειων 37](#_Toc516971701)

[Figure 43: Ελεγχος για υπαρξη όμοιας εταιρείας στο σύστημα 37](#_Toc516971702)

[Figure 44: Προσθήκη εταιρείας στο σύστημα 38](#_Toc516971703)

[Figure 45: Αρχειο εταιρείων πρίν την προσθήκη νεας εταιρείας 38](#_Toc516971704)

[Figure 46: Προσθήκη εταιρείας στο αρχειο 39](#_Toc516971705)

[Figure 47: Αρχειο μετά την προσθήκη εταιρείας 39](#_Toc516971706)

[Figure 48: Εκτύπωση παραγγελιων 42](#_Toc516971707)

[Figure 49: Δυο τρόποι εκτυπώσεων για τον πελάτη 43](#_Toc516971708)

[Figure 50: Κώδικας υλοποίησης JFrame εκτυπώσεων 43](#_Toc516971709)

[Figure 51: Εκτυπωση προιόντων ανα ειδος 44](#_Toc516971710)

[Figure 52: Κλήση συναρτήσεων για εκτυπωση διαφορετικών προιόντων 44](#_Toc516971711)

[Figure 53: Εκτύπωση τηλεοράσεων 45](#_Toc516971712)

[Figure 54: Κώδικας εκτύπωσης τηλεοράσεων 45](#_Toc516971713)

[Figure 55: JFrame για το καλάθι αγορών 46](#_Toc516971714)

[Figure 56: Προσθήκη προιόντος στο καλάθι αγορών 46](#_Toc516971715)

# ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω αρχικά τους γονείς που χωρίς την αγάπη αλλα και την στήριξη τους σε όλα τα χρόνια των σπουδών μου, δεν θα είχα καταφέρει τίποτα στη ζωή μου. Επίσης θέλω να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου Παπαδάκη Νικόλαο, που χωρίς τις πολύτιμες συμβουλές και την καλή διάθεση προς επίλυση αποριών δεν θα έβγαινε εις πέρας η πτυχιακή μου εργασία. Τέλος θα ήθελα να ευχαριστήσω ιδαίτερα τον Όγλου Γεώργιο, χρόνια φίλο και συμφοιτητή μου για τις πολύτιμες συμβουλές του στο θέμα της γλώσσας προγραμματισμού Java.

# ABSTRACT

As technology evolves on a daily basis at a rapid pace, today's man has found many ways to make his life easier in every aspect of his life.

Whether **its** smart devices such as smart phones, or this is a new way of buying and selling products and services with a major profit at the time that is so precious in our times. Now a person can buy and sell via the internet without the need to go to the store of his choice.

It can be very easy to have a good internet connection and little computer knowledge to go to a website that supports an online store and to buy or sell it quickly and quickly as he wishes.

The purpose of this thesis is to introduce to us some e-commerce feature. Also, to give us an understanding of what an e-shop is, and for what reason we must visit a website that has an online store or not.

Then we will try to install the basic tools for developing our software while in the last part we will describe in detail the implementation of an e-shop using the Java programming language.

# 1. ΠΕΡΙΛΗΨΗ

Καθώς η τεχνολογία καθημερινά εξελίσεται με ραγδαίους ρυθμούς, ο σημερινός άνθρωπος έχει βρεί πάρα πολλούς τρόπους για να κάνει τη ζωή του ευκολότερη σε κάθε τομέα της ζωής του.

Είτε αυτό αφορά έξυπνες συσκευές όπως έξυπνα κινητά είτε αυτό αφορά εναν νέο τρόπο αγοραπωλησίας υπηρεσιών και προιόντων με κύριο κέρδος τον χρόνο που στην σημερινη εποχή μας είναι τόσο πολύτιμος. Πλέον ενας άνθρωπος μπορεί να αγοράσει και να πουλήσει μέσω διαδικτύου χωρίς να υπάρχει η ανάγκη να μεταβεί στο κατάστημα της αρεσκείας του.

Μπορεί πολύ έυκολα διαθέτωντας μια καλή σύνδεση στο διαδίκτυο και ελάχιστες γνώσεις υπολογιστών να μεταβεί σε μια ιστοσελίδα που υποστηρίζει ενα ηλεκτρονικό κατάστημα και να αγοράζει ή να πουλήσει άμεσα και γρήγορα ότι αυτός επιθυμεί.

Ο σκοπός της πτυχιακής εργασίας είναι να μας παρουσιάζει μερικά χαρακτηριστικά του ηλεκτρονικού εμπορίου, να μας δώσει να καταλάβουμε το τί ειναι ενα ηλεκτρονικό κατάστημα και για ποίο λόγο και αν, πρέπει να επισκεφτούμε μια ιστοσελίδα που διαθέτει ενα ηλεκτρονικό κατάστημα ή οχι.

Στη συνέχεια θα προσπαθήσουμε να εγκαταστήσουμε τα βασικά έργαλεια για την ανάπτυξη του λογισμικού μας ενώ στο τελευταίο μέρος θα περιγράψουμε αναλυτικά την υλοποίηση ενός ηλεκτρονικού καταστήματος με χρήση της γλώσσας προγραμματισμού Java.

# 2. ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΚΑΤΑΣΤΗΜΑ( E – SHOP)

## 2.1 ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ

Το ηλεκτρονικό εμπόριο ειναι μια διαδικασία κατα την οποία πωλούνται και αγοράζονται προιόντα και υπηρεσίες διαμέσου του διαδικτύου. Συνήθως οι επιχειρήσεις ηλεκτρονικού εμπορίου απευθύνονται απευθείας στους καταναλωτές μέσω ιστοσελίδων οπως και εφαρμογών για κινητά τηλέφωνα (Wills, 2014)

Στην σημερινή εποχή αν όχι εξ ολοκλήρου ενα μεγάλο μέρος των εταιρείων έχουν στραφεί στο να πωλούν τα προιόντα ή τις υπηρεσίες τους μέσω του διαδικτύου. Για παράδειγμα ένα ηλεκτρονικό κατάστημα μπορεί να παρέχει προιόντα κινητής τηλεφωνίας ή ακομη και υπηρεσίες όπως online αποθηκευτικό χώρο ( cloud services). Μερικές απο τις πιο γνωστές πλατφόρμες ηλεκτρονικού εμπορίου είναι το Amazoon και το e-Bay (Wienclaw, Ruth, 2013)

Το ηλεκτρονικό εμπορίο έχει τα παρακάτω **θετικά** χαρακτηριστικά που το κάνουν να **ξεχωρίζει** απο το απλό «τυπικό» εμπόριο που τοσα χρόνια γνωρίζαμε (Khurana, 2017) :

* Πλέον ενας πελάτης δεν χρειάζεται να μεταβεί στο κατάστημα για να αγοράσει κάτι. Ενα μόνο κλίκ αρκεί ώστε η παραγγελία του να έρθει στο σπίτι του εντός ολίγων ημερών.
* Ενας πελάτης μπορεί μπορεί μέσω της επίσκεψης του στον ιστότοπο του καταστήματος να δεί αναλυτικά τα χαρακτηριστικά των προιόντων που τον ενδιαφέρουν καθώς και εικόνες των προίοντων. Αυτό βοηθά στην ταυτόχρονη σύγκριση τιμών και χαρακτηριστικών απο άλλα καταστήματα, του ίδιου προιόντος. Το μονο που χρειάζεται κανεις ειναι μια απλή σύνδεση στο διαδίκτυο
* Λόγω του οτι ενα ηλεκτρονικό κατάστημα στεγάζεται στο διαδίκτυο, επόμενο είναι να μη απασχολεί πάρα πολλούς εργαζόμενους με αποτέλεσμα να μπορεί να «ρίξει» τις τιμές ενος προιόντος ή μιας υπηρεσίας
* Λογω της ραγδαίας ανάπτυξης της τεχνολογίας, υπάρχουν προχωρημένα συστήματα προγραμματισμού που επιτρέπουν στο ηλεκτρονικό κατάστημα να καταγράφει τις συνήθειες των πελατών με αποτέλεσμα να προτείνει παρόμοια προιόντα για αγορά χωρίς να χρειαστεί να διερευνήσει μόνος του ολόκληρο τον κατάλογο προιόντων.
* Ενας πελάτης που θα σκεφτεί να αγοράσει ενα προιόν ή μια υπηρεσία μπορεί να αφήσει μια κριτική κάτω απο το προιόν. Επομένως ενας επόμενος πελάτης που θα σκεφτεί να αγοράσει το ιδιο προιόν/υπηρεσία θα μπορέσει να καταλάβει εκ των προτέρων εαν το προιόν ειναι λειτουργικό και άξιο αγοράς πρωτού το αγοράσει.
* Ενα ηλεκτρονικό κατάστημα μπορεί να στείλει τα προιόντα του σε όλο τον κόσμο όπου και αν αυτός βρίσκεται. Ακόμη και σε διαφορετική ήπειρο.
* Ενα ηλεκτρονικό κατάστημα καθώς «τρέχει» στο διαδίκτυο, μένει ανοικτό εικοσιτέσσερις ώρες το εικοσιτετράωρο
* Πολλές φορές κάποιοι χρειάζονται να αλλάξουν το είδος του καταστήματος τους ή να προσθέσουν προιόντα ή υπηρεσιες. Αυτό δε μπορεί να γινει εύκολα με ενα κατάστημα που διαθέτει φυσική τοποθεσία! Απο την άλλη πλευρά, ενα ηλεκτρονικό κατάστημα μπορεί να αλλάξει το στύλ του (niche) πάρα πολύ εύκολα.

Απο την άλλη πλευρά το **ηλεκτρονικό εμπόριο** έχει τα παρακάτω αρνητικά (Khurana, 2018):

* Τα προιόντα που μπορεί να παραγγείλει κάποιος, δεν έρχονται απευθείας και σε υπερβολικά σύντομο χρονικό διάστημα στην τοποθεσία που όρισες για την παραλαβή αυτών. Πολλές φορές η μεταφορά των προιόντων μπορεί να αργήσει μέχρι και έναν μήνα εαν για παράδειγμα παραγγείλει κάποιος κάτι απο το εξωτερικό.
* Πολλά προιόντα είναι δύσκολο να επιστραφούν λόγω των μεταφορικών που αναλαμβάνει η εταιρεία για να πάρει πίσω. Ένα τέτοιο παράδειγμα είναι η αγορά επίπλων απο το διαδίκτυο.
* Παρόλο που τα προιόντα διαθέτουν κριτικές κάτω απο την περιγραφή, ο κάθε άνθρωπος είναι διαφορετικός, οπότε η κριτική ενός προιόντος είναι υποκειμενική με αποτέλεσμα το προιόν που είμαστε διαθέσιμοι να αγοράσουμε να μην ταιριάζει απόλυτα σε εμάς (Chisholm, 1911)
* Ενα μεγάλο θέμα , είναι η ασφάλεια των συναλλαγών. Το ίντερνετ, η γρήγορη εξάπλωση της τεχνολογίας αλλα και η χρήση συνεχώς των πιστωτικών μας καρτών στο διαδίκτυο προκαλεί πολλές φορές «πατήματα» για κακόβουλους χρήστες/ανθρώπους να βρούν τρόπο να υποκλέψουν τους τραπεζικούς μας λογαριασμούς. Οπότε ενα ηλεκτρονικό κατάστημα πρέπει να διαθέτει άριστη προστασία των προσωπικών μας δεδομένων.

## 2.2 Η ΛΕΙΤΟΥΡΓΙΑ ΕΝΟΣ ESHOP & ΓΝΩΣΤΕΣ ΠΛΑΤΦΟΡΜΕΣ ΠΑΡΟΧΗΣ ΠΡΟΙΟΝΤΩΝ ΚΑΙ ΥΠΗΡΕΣΙΩΝ

### 2.2.1 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ

Το ηλεκτρονικο κατάστημα δεν ειναι τίποτε αλλο απο μια ιστοσελίδα που περιέχει προιόντα και υπηρεσίες διαθέσιμα/ες προς τον πελάτη. Ενας πελάτης για να κάνει είσοδο στο ηλεκτρονικό κατάστημα πρέπει αρχικά να έχει δημιουργήσει ενα λογαριασμό με τα βασικά στοιχεία οπως ειναι για παράδειγμα ενα username ενα password και το e-mail του.

Στη συνέχεια το κατάστημα στέλνει ενα e-mail επιβεβαίωσης στο e-mail του. Ο χρήστης πλέον έχει γραφεί στο σύστημα. Στη συνέχεια μόλις κάνει είσοδο στο κατάστημα για την πρώτη φορά πρεπει να συμπληρώσει το προφίλ του , δηλαδή τα βασικά στοιχεία που θα ητανε απαραίτητα για να μπορέσει κάποιος να αγοράσει απο το συγκεκριμένο κατάστημα.

Για παράδειγμα θα ζητηθεί ενα τηλέφωνο επικοινωνίας, σίγουρα μια διευθυνση παραλαβής, και αναλόγα το κατάστημα, πιθανόν και τρόπο πληρωμής ( αντικαταβολή ή χρεωστική/ πιστωτική καρτα).

Ενας πελάτης κάθε φορά που θελήσει να αγοράσει κάτι απο το ηλεκτρονικό κατάστημα, πολύ απλά θα κάνει είσοδο με το username και το password και στη συνέχεια πηγαίνοντας στην κατάλληλη κατηγορία προιόντων, βρισκει το προιόν που τον ενδιαφέρει και το προσθέτει στο καλάθι αγορών του(και την ποσότητα προιόντων απο το συγκεκριμένο είδος που επέλεξε).

Οταν είναι σίγουρος οτι θέλει να αγοράσει, πατάει επικύρωση και η παραγγελία είναι έτοιμη να φύγει απο το κατάστημα. Αρκετές φορές μπορεί να χρειαστεί να καλέσουν τον πελάτη τηλεφωνικά για να επιβεβαιώσει την παραγγελία. Απο εκεί και πέρα όλα τα υπόλοιπα τα αναλαμβάνει η εταιρεία ταχυμεταφορών(courier) που συνεργάζεται το εκάστωτε κατάστημα.

### 2.2.2 AMAZOON & EBAY

Μερικές απο τις πιο γνωστές πλατφόρμες πώλησης διαδικτυακών προιόντων είναι η Amazoon και το e-Bay. Παρακάτω βλέπουμε την κεντρική σελίδα του Amazoon(**Figure : 1**)

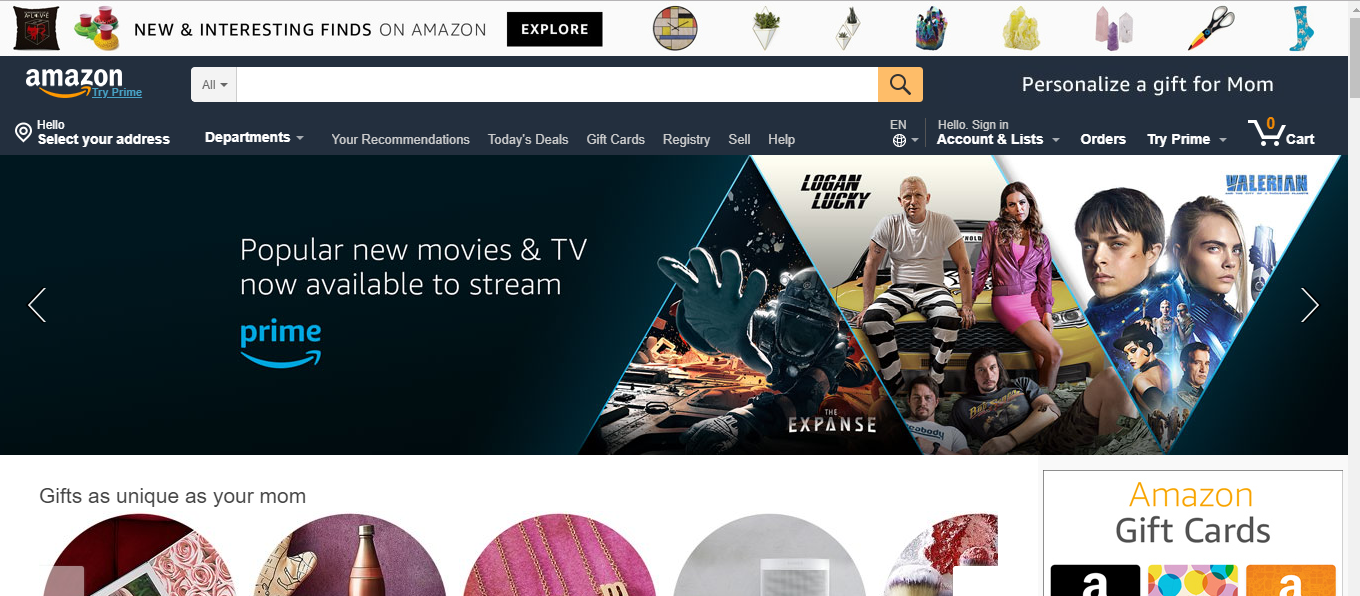


Figure 1: H πλατφόρμα Amazoon

Παρατηρούμε οτι η ιστοσελίδα διαθέτει είσοδο στο σύστημα ωστε ενας πελάτης να εισέλθει μετα προσωπικά του στοιχεία για να μπορέσει να αγοράσει προιόντα(δεξιά φαινεται το καλάθι αγορών). Με παρόμοιο τρόπο έχει στηθεί και η πλατφόρμα e-Bay (**Figure: 2**)

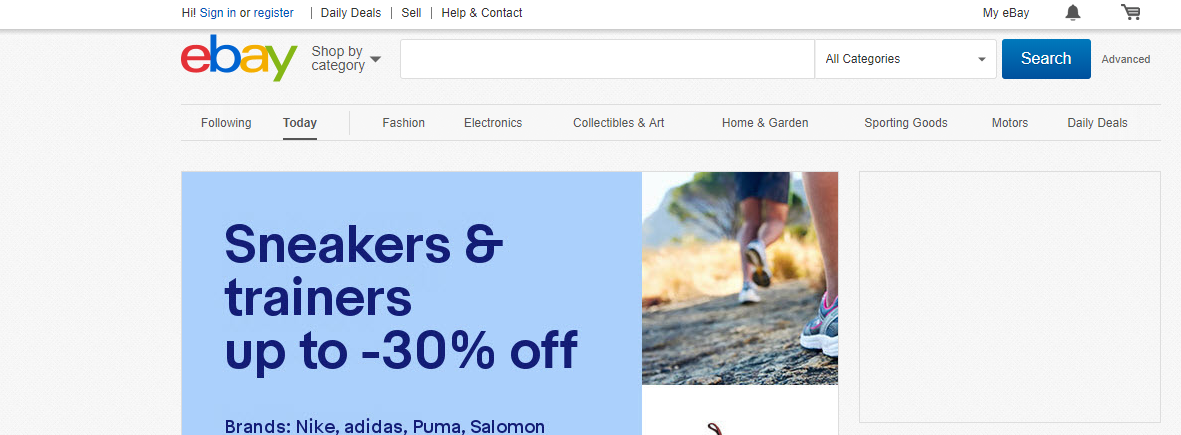


Figure 2: Η πλατφόρμα ebay

# 3. ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ NETBEANS & ΛΟΙΠΩΝ ΕΡΓΑΛΕΙΩΝ

Για να μπορέσουμε να υλοποιήσουμε την εφαρμογή , χρειαζόμαστε το σύστημα μας να τρέχει Java και την πλατφόρμα NETBEANS IDE το οποιο είναι μια πλατφόρμα μέσω της οποίας θα γράψουμε αλλα και θα τρέξουμε την εφαρμογή μας. Η εγκατάσταση των παραπάνω φαίνεται παρακάτω:

## 3.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA RUNTIME

Αρχικά μεταβαίνουμε στην ιστοσελίδα:

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

Και κατεβάζουμε την παρακάτω εφαρμογή(**Figure: 3**)

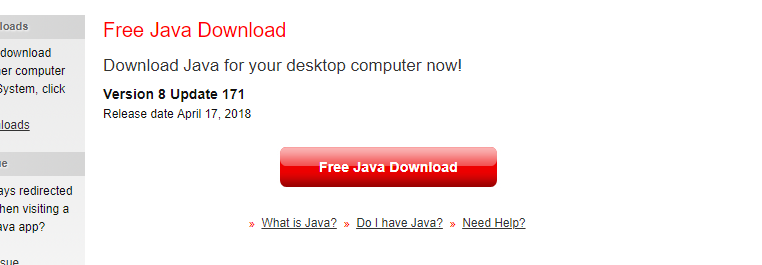


Figure 3: Η εγκατάσταση της JAVA

## 3.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS IDE

Στη συνέχεια μεταβαίνουμε στην ιστοσελίδα:

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

Κατεβάζουμε το NETBEANS 8.2 IDE μαζί με το JDK αφού πρώτα συμφωνήσουμε στους όρους που θέτει η εταιρεία(**Figure: 4**)

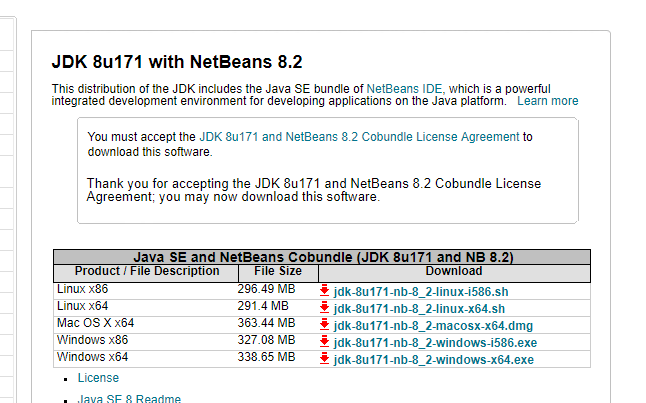


Figure 4: Η εγκατάσταση της πλατφόρμας NETBEANS 8.2 IDE + JDK

Αφού κατεβάσουμε τα παραπάνω για το δικό μας λειτουργικό σύστημα( εμείς διαθέτουμε Windows και οχι Linux) πατάμε σε όλα ναι και η εγκατάσταση σε 10-12 λεπτά έχει τελειώσει.

Τώρα είμαστε έτοιμοι να ανοίξουμε την εφαρμογή NETBEANS και να γράψουμε κώδικα. Ομως πρωτού προχωρήσουμε στην επεξήγηση του κώδικα που έχουμε γράψει, θα κάνουμε μια μικρή παρουσίαση του τρόπου λειτουργίας της πλατφόρμας NETBEANS.

## 3.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS 8.2 IDE

Μόλις ανοίξουμε την εφαρμογή βλέπουμε το παρακάτω κεντρικό πανελ( **Figure: 5**)

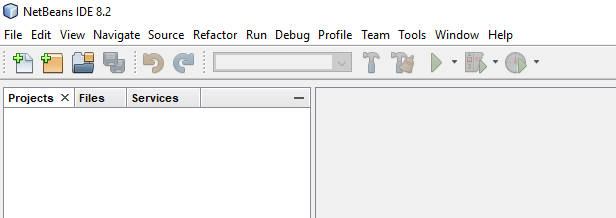


Figure 5: Αρχικό πάνελ του NETBEANS

Στο σημείο projects εμφανίζονται τα project τα οποία έχουμε ανοίξει ενώ στο File μπορουμε να ανοίξουμε ενα ήδη υπάρχων project , να δημιουργήσουμε κάποιο νέο , όπως και το να κλείσουμε κάποιο απο τα προτζεκτ μας ( **Figure: 6**)

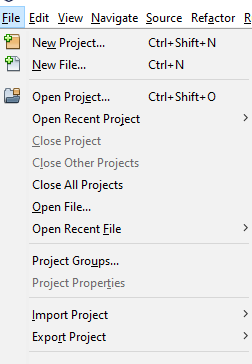


Figure 6: Η επιλογή File του NETBEANS

Με την επιλογή Edit μπορούμε να κανουμε copy/paste/cut τον κώδικα μας οπως και να χρησιμοποιήσουμε την επιλογή Find & Replace για να βρούμε κάτι που χρειαζόμαστε στον κώδικα ή να αντικαταστήσουμε κάποιο μέρος του χωρίς να χρειάζται να ψάξουμε τον κώδικα γραμμή παρα γραμμή( **Figure: 7** )

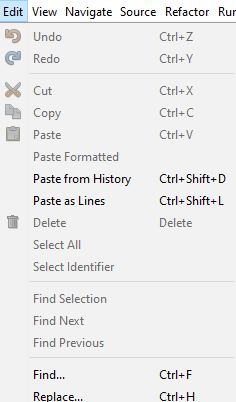


Figure 7: Η επιλογή Edit του NETBEANS

**Απο όλες τις υπόλοιπες** επιλογές μας αφορά η επιλογή Run μεσω της οποίας τρέχει ο κώδικας. Στο κάτω μέρος του κεντρικού πάνελ του NETBEANS υπάρχει το console στο οποίο εμφανίζονται τα τυχόν λάθη του κώδικα ή η επιτυχής υλοποίηση του.

# 4. ΠΑΡΟΥΣΙΑΣΗ/RUNNING ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Για να κάνουμε είσοδο στο σύστημα ώς admin βάζουμε ως username : vasilis και password : 12345. Αρχικά με το που ανοίξουμε την εφαρμογή μας εφανίζει ενα JFrame panel που μεσω αυτού μπορούμε είτε να κάνουμε είσοδο είτε εγγραφη ενός νέου χρήστη (**Figure: 8**)

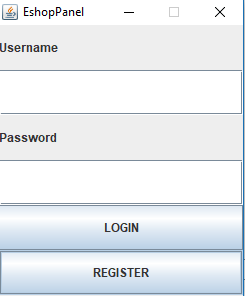


Figure 8: JFrame εισόδου και εγγραφής

Θα δούμε στη συνέχεια τον κώδικα που χρειαστήκαμε για να υλοποιηθεί η είσοδος ενός χρήστη στο σύστημα( **Figure: 9 )**

## 4.1 Η ΠΛΕΥΡΑ ΤΟΥ ADMIN ( ADMIN SIDE )

Μολις κάνουμε είσοδο ως admin εμφανίζεται το παρακάτω JFrame(**Figure: 10**)

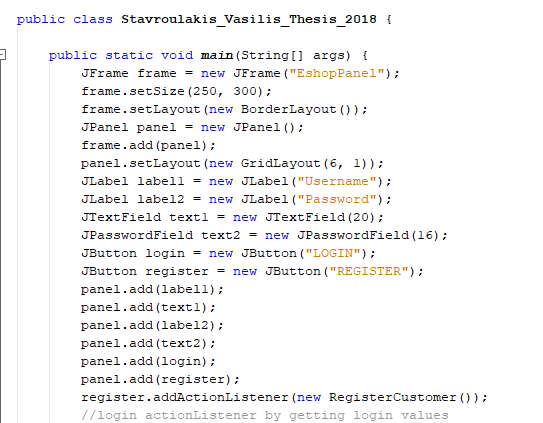


Figure 9: JFrame για εισοδο/εξοδο χρήστη απο σύστημα

Βλέπουμε οτι δημιουργήσαμε ενα JFrame, στη συνέχεια ενα JPanel , τα κατάλληλα JTextFields και JLabels και δώσαμε στο κουμπί register εναν actionListener(). Ο actionListener του register υλοποιεί αρχικά τα παρακάτω( **Figure: 10** )

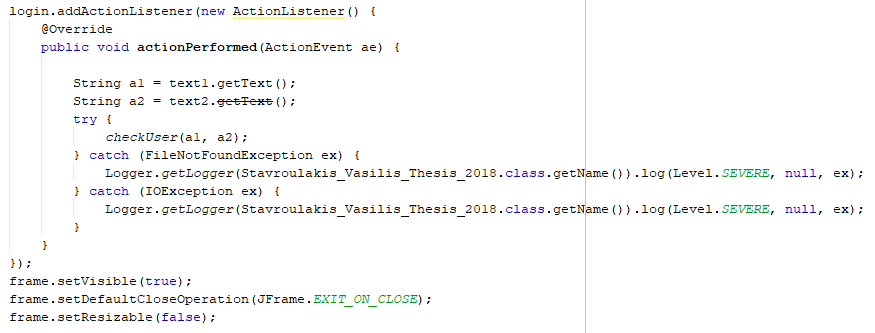


Figure : actionListener() για το κουμπί register

Βλέπουμε οτι πήραμε τις τιμές των JTextFields που δημιουργήσαμε πιο πάνω δηλαδή τις τιμές username και password και τις περάσαμε σε μια κλάση που την ονομάσαμε checkUser η οποία θα ελέγξει με κατάλληλους τρόπους την είσοδο ενος admin ή customer. Φυσικά τα κατάλληλα exceptions και τις υποχρεωτικές λειτουργίες για να εμφανίζεται το frame στην οθόνη. Ας δούμε το πώς λειτουργεί η κλάση checkUser.

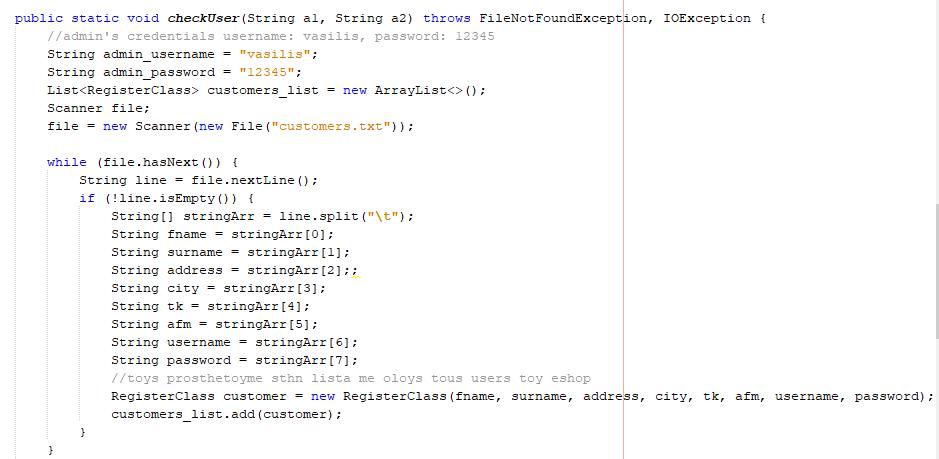


Figure 11: CheckUser (a)

Εδώ αρχικά περνάμε στη κλάση τις τιμές username , password που λάβαμε απο τον χρήστη, στη συνέχεια δημιουργούμε ενα arrayList που είναι σύμφωνα με το RegisterClass και ανοίγουμε το αρχείο customers.txt και μέσω της χρήσης μιας κλάσης while προσπαθούμε να φορτώσουμε στη λίστα customers\_list τα περιέχομενα του .txt αρχείου. Πρωτού συνεχίσουμε ας δούμε το πώς δημιουργήθηκε η RegisterClass ( **Figure : 12**)

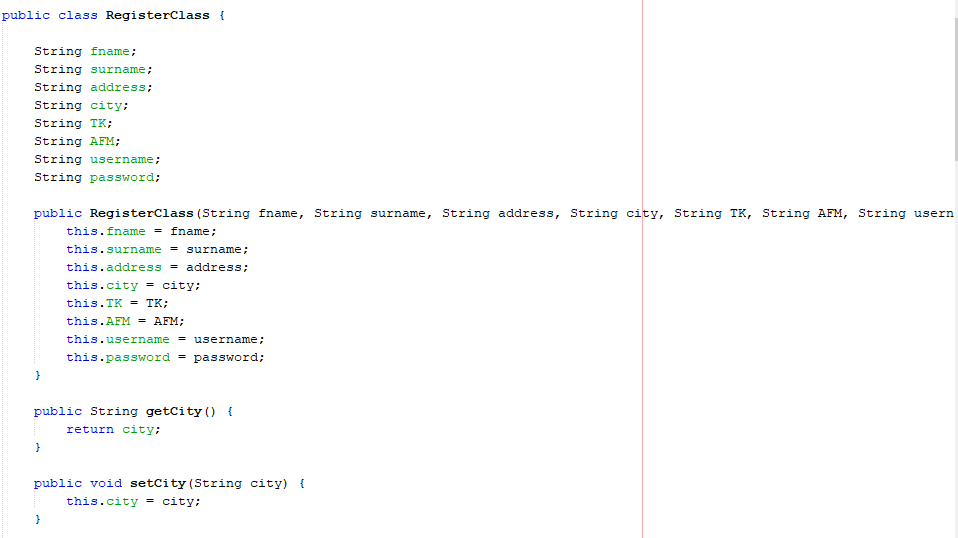


Figure 12: RegisterClass

Δώσαμε μερικά στοιχεία που πιθανόν να είχε ενας πελάτης, δημιουργήσαμε τον κατάλληλο constructor και γράψαμε τις απαραίτητητες getters and setters. Το σύστημα μας διαθέτει μόνο εναν admin οπότε θα κάνουμε παρακάτω έλεγχο για το εαν τα στοιχεία που δώσαμε είναι αυτά που γνωρίζουμε για τον admin αλλιως θα ψάξουμε μέσα στο αρχείο των customers για να δούμε εαν πρόκειται για πελάτη( **Figure: 13**)

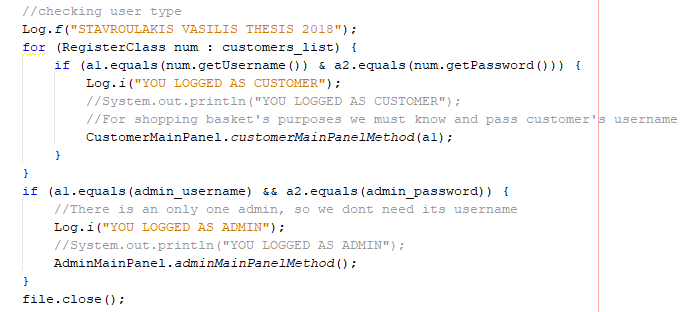


Figure 13: CheckUser (b)

Εδώ ελέγχουμε εαν στη λίστα που ανακατασκευάσαμε απο το αρχείο customers.txt υπάρχει πελάτης σύμφωνα με τα στοιχεία που έδωσε ο χρήστης. Εαν ναί τότε έκανε είσοδο ως πελάτης και μεταβαίνει στο κεντρικό μενού(πάνελ) του πελάτη.

Εαν όμως δώσαμε στοιχεία που ειναι ίδια με αυτά του admin ( a1 ειναι το username , ενω a2 το password του admin). Εαν ειναι ίδια, τότε μεταβαίνουμε στο κεντρικό μενού(πάνελ) του admin. Φυσικά πάντα κλείνουμε τα αρχεία για να μην υπάρξουν προβλήματα.

## 4.2.1 ΕΙΣΟΔΟΣ ΩΣ ADMIN ΣΤΟ ΣΥΣΤΗΜΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ

Εαν κάνουμε είσοδο ώς admin τότε θα ανοίξει το παρακάτω JFrame(**Figure: 14** )



Figure 14: Admin Panel

Οπως βλέπουμε έχουμε υλοποιήσει τις παραπάνω λειτουργίες. Ενας admin δηλαδή μπορεί να εκτυπώσει τον κατάλογο προιόντων, να προσθέσει προιόντα στο υποκατάστημα να αφαιρέσει τα προιόντα όπως και να εκτυπώσει πελάτες.

Επίσης μπορεί να προσθέσει/αφαιρέσει εταιρείες ή να οριστικοποιήσει μια παραγγελία που έχει κανει πρωτύτερα ενας πελάτης. Τέλος μπορεί να δεί τις παλίες παραγγελίες ενός πελάτη και να κανει έξοδο απο το σύστημα.

Αναλυτικά θα περιγράψουμε το πώς λειτουργεί απο πλευράς κώδικα και το πώς εμφανίζεται η όλη λειτουργία στη συνέχεια. Αρχικά όμως ας δούμε το πώς δημιουργήθηκε το κεντρικό admin panel( **Figure : 15**)



Figure : admin main panel (a)

Παραπάνω φαίνονται οι λειτουργίες του admin. Δημουργήσαμε JFrame και ενα panel και ενσωματώσαμε σε αυτό κάποια buttons που υλοποιούν τις παραπάνω λειτουργίες. Στη συνέχεια βλέπουμε οτι ( **Figure: 16**) κάθε κουμπί μας στέλνει σε διαφορετική κλάση.

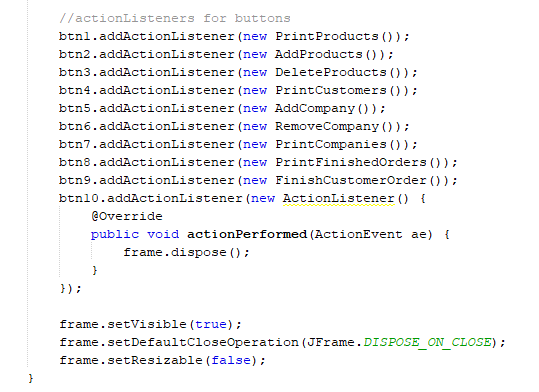


Figure : admin main panel (b)

Θα περιγράψουμε κάθε κλάση ξεχωριστά στις παρακάτω υποενότητες.

### 4.2.1.1 ADD PRODUCTS

Η κλαση AddProducts.java ουσιαστικά μας στέλνει με κατάλληλα κουμπιά σε διαφορετικά JFrame για να προσθέσουμε προιόντα ανάλογα με το είδος τους. Για να προσθέσουμε μια κάμερα δημιουργούμε ενα JFrame και ενα JPanel. Αυτό φαινεται παρακάτω(**Figure: 17**)

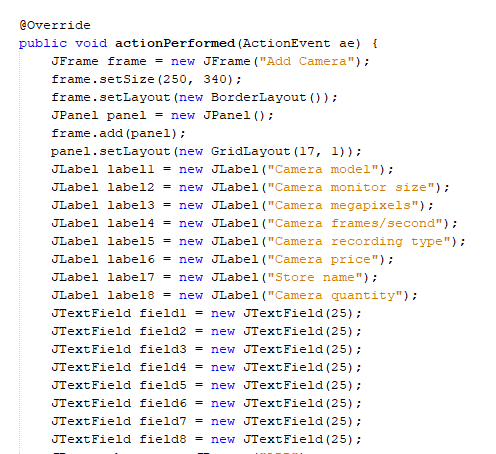


Figure 17: Add Camera

Κατόπιν τα κατάλληλα JTextFields, JLabels με σκοπό μέσω ενός actionListener να μπορέσουμε να δημιουργήσουμε ένα νέο αντικείμενο τύπο κάμερας. Επιπρόσθετα ο αριθμός των αντικειμένων που δημιουργούμε είναι σύμφωνα με την ποσότητα των καμερών που έδωσε ο πελάτης στη φόρμα του προιόντος.

Κάθε φορά καλείται η κλάση writeProduct (εδώ writeCamera) η οποία καλείται τόσες φορές, οσες ειναι τα αντικείμενα που δημιουργήσαμε. Η συνέχεια του κώδικα φαίνεται παρακάτω(**Figure: 18**)

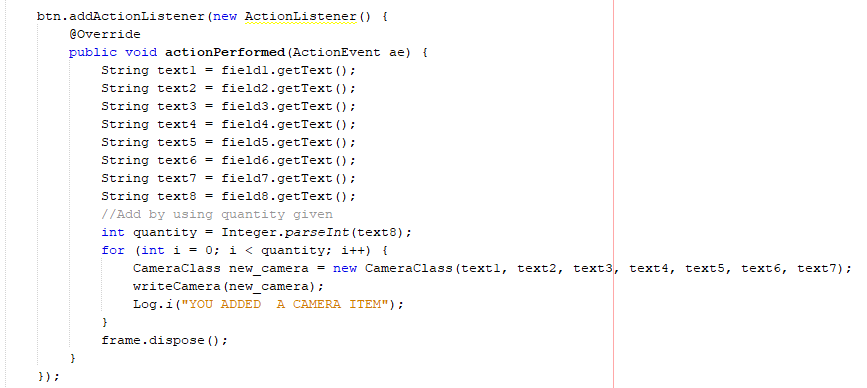


Figure 18: Add camera (b)

Τέλος, η κλάση ανοίγει το αρχείο cameras.txt και γράφει τα αντικείμενα στο αρχείο. Κάθε αντικείμενο γράφεται με την βοήθεια της .toString() κάθε κλάσης(προιόντος) και τα κατάλληλα FileWriter και BufferedReader(**Figure: 19)**

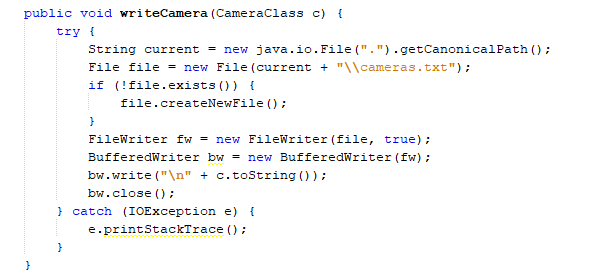


Figure 19: Add camera (c)

Το JFrame της προσθήκης μιας κάμερας φαίνεται παρακάτω(**Figure: 20**)

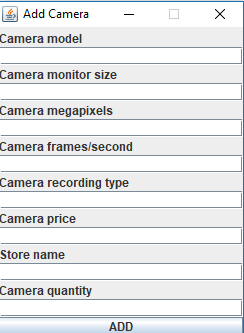


Figure 20: Add camera

Τώρα θα δούμε το πώς μπορούμε να προσθέσουμε για παράδειγμα 5 κάμερες στο αρχείο cameras.txt. Το αρχικό αρχείο ήδη είχε μερικές κάμερες μέσα(**Figure: 21**)

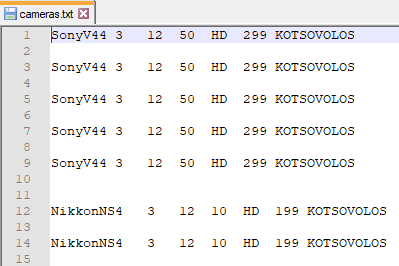


Figure 21: Αρχειο cameras.txt πριν την εισαγωγή νέων καμερών

Μετά την προσθήκη των παρακάτω στοιχείων στη φόρμα(**Figure: 22**)

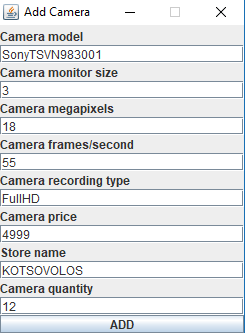


Figure 22: Φορμα add camera

Μετά την υποβολή των παραπάνω στοιχείων το αρχείο έχει αλλάξει( **Figure: 23**)

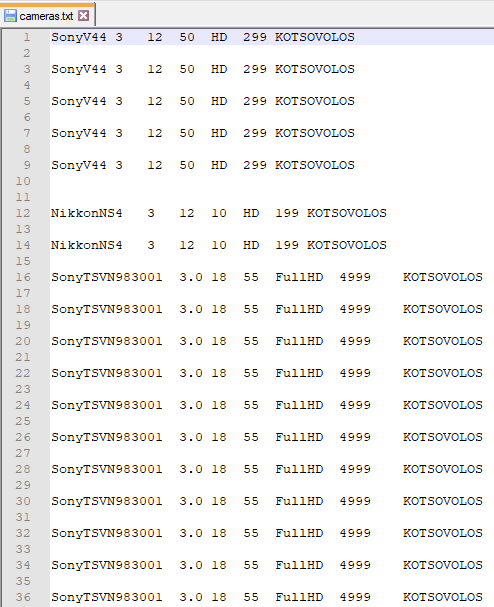


Figure 23: Αρχειο cameras.txt μετα την εισαγωγή (κομμάτι αυτού)

Με παρομοιο τρόπο λειτουργούν και οι υπόλοιπες προσθήκες των προιόντων για παράδειγμα εαν θέλουμε να προσθέσουμε κονσόλες παιχνιδιών στο κατάστημα μας τότε το κατάλληλο JFrame ( φορμα προσθήκης προιόντος ) που εμφανίζεται στον admin ειναι το παρακάτω(**Figure: 24**)

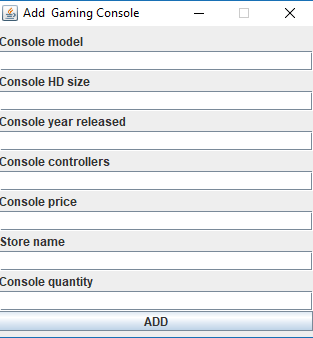


Figure 24: Φορμα προσθήκης gaming console

Ενώ το αρχείο πρωτού γινει update με το νέο προιόν/οντα ειναι το εξής(**Figure: 25**)

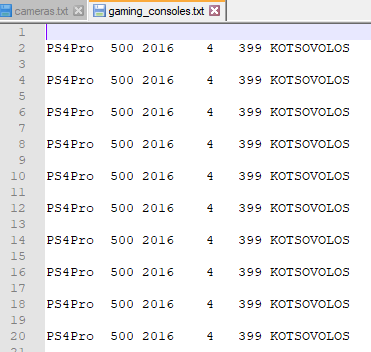


Figure 25: Αρχειο gaming\_consoles πριν την εισαγωγή κονσόλας

Εαν προσθέσουμε το παρακάτω προιόν(**Figure: 26**)

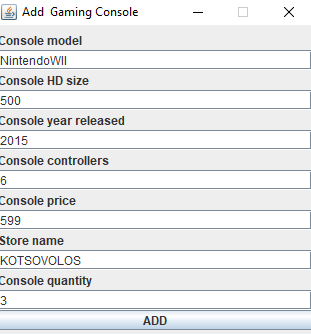


Figure 26: Η προσθήκη μιας κονσόλας στο αρχείο

Τότε το αρχείο gaming\_consoles.txt θα αλλάξει( **Figure: 27**)



Figure : To αρχειο gaming\_consoles μετα την προσθήκη κονσόλων

### 4.2.1.2 PRINT PRODUCTS

Επιλέξαμε να κάνουμε εκτύπωση των συνολικών προιόντων. Μιας και έχουμε 5 προιόντα στο κατάστημα, ουσιαστικά θα ανοίξουμε τα 3 αρχεία, κατόπιν θα τα φορτώσουμε σε κατάλληλα arrayLists και θα τα κάνουμε εκτύπωση χρησιμοποιώντας JTextArea και JScrollPane. Ο κώδικας για να φορτώσουμε αρχεία φαίνεται παρακάτω( **Figure: 28**)

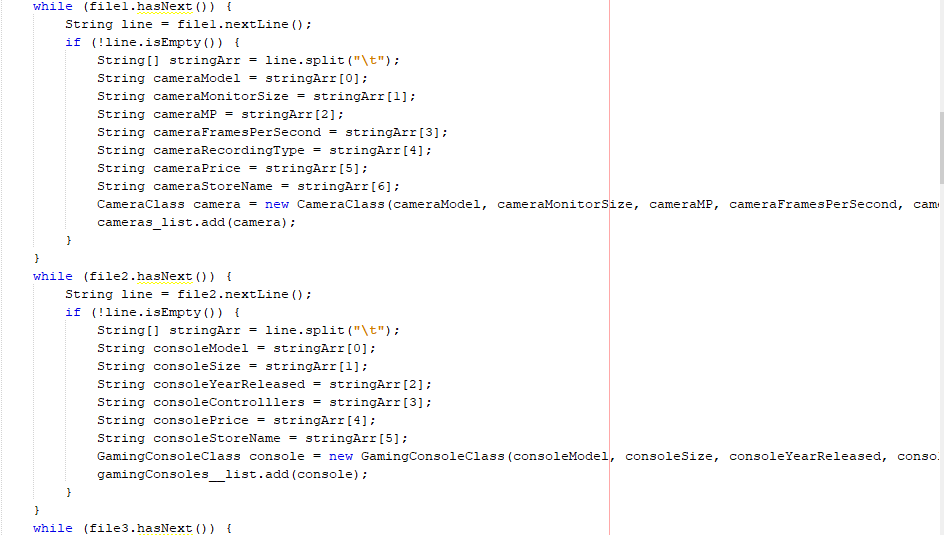


Figure 28: Φόρτωση αρχείων για εκτύπωση(κομμάτι κώδικα)

Δημιουργούμε JFrame και JTextArea που θα ενσωματωθεί στο frame. Στη συνέχεια το JScrollPane θα ενσωματωθεί στο frame ενώ κάθε φορά θα διαβάζουμε τα αντικείμενα απο κάθε αρχείο και με κατάλληλη μορφοποίηση θα εμφανιστούν στην οθόνη( **Figure: 29)**

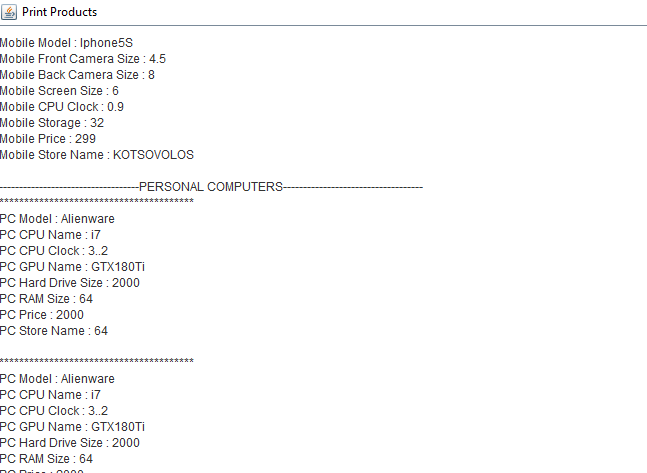


Figure 29: Εκτύπωση συνολικών προιόντων

Ενώ ο κώδικας φαίνεται παρακάτω(**Figure:30**)

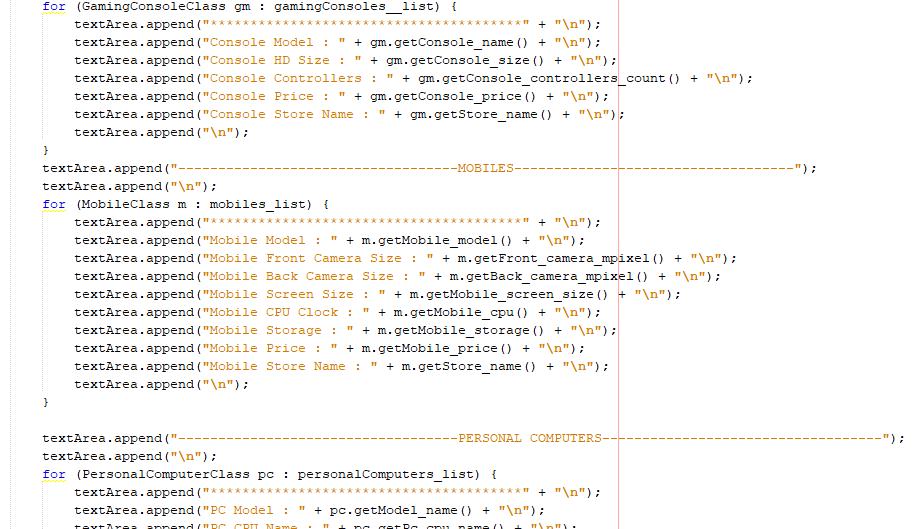


Figure 30: Εκτύπωση προιόντων (κομματι κώδικα)

### 4.2.1.3 DELETE PRODUCTS

Για να σβήσουμε ενα προιόν πρέπει καταρχάς να ξέρουμε κάποια χαρακτηριστικά για το συγκεκριμένο προιόν, όπως ειναι για παράδειγμα το είδος του. Ετσι, δημιουργήσαμε την κλάση DeleteProducts.java με σκοπό να μπορέσουμε να μεταβούμε στην κατάλληλη διαγραφή προιόντος. Μέρος του κώδικα της κλάσης φαίνεται παρακάτω(**Figure: 31**)



Figure 31: Κώδικας μετάβασης για διαγραφή κατάλληλου προιόντος

O κώδικας ειναι παρόμοιος σε κάθε περίπτωση, οπότε θα επεξηγήσουμε μία απο τις παραπάνω διαγραφές προιόντων. Για παράδειγμα έστω οτι θέλουμε να διαγράψουμε εναν ηλεκτρονικό υπολογιστή απο το αρχείο personalComputers.txt . Αρχικά μεταβαίνουμε στη κλάση πατώντας το κατάλληλο κουμπί απο το frame της κλάσης DeleteProducts.java( **Figure: 32**)



Figure 32: Διαγραφή προιόντος(κεντρικό πανελ)

Αρχικά θα μεταβούμε στην κλάση RemovePC. Θα δημιουργήσουμε ενα κατάλληλο JFrame και θα δώσουμε τα κατάλληλα στοιχεια για την διαγραφή(**Figure: 33**)

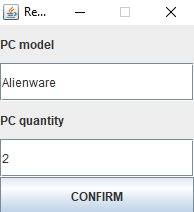


Figure 33: Διαγραφή ενός υπολογιστή απο το αρχειο

Ουσιαστικά μέσω του confirm ο actionListener παιρνει τις παραπάνω τιμές και τις στέλνει στη κλάση reconstructArrayList. Μεσω αυτής της κλάσης, ανακατασκευάσουμε το arrayList με τους υπολογιστές απο το αρχείο(**Figure: 34**)

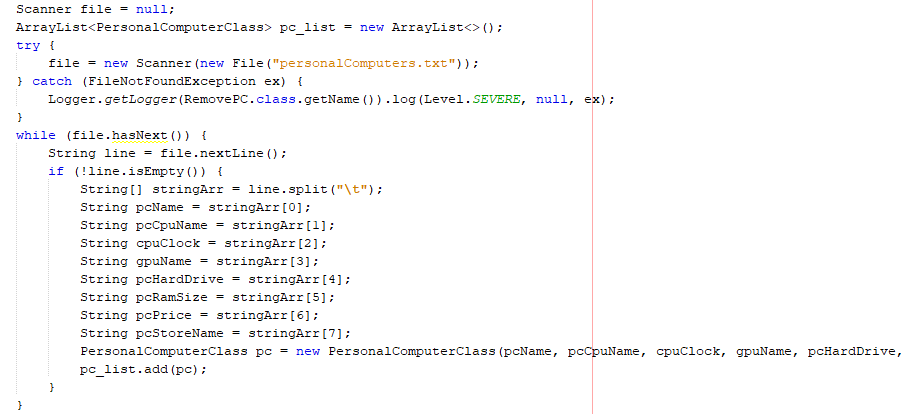


Figure 34: Ανακατασκευή του arrayList pcs\_list

Κατόπιν χρησιμοποιώντας μια κλάση που καλείται μέσα στη προηγούμενη κλάση που ονομάζεται removeUsingIterator προσπαθούμε, έχοντας το μοντέλο αλλα και την ποσότητα(τα έδωσε ο admin στην φόρμα) έχοντας ώς μέγιστο όριο τον αριθμό/ποσότητα που δώσαμε, να προσπαθεί μέσω ενός iterator κάθε φορά να αφαιρεί απο τη arrayList ενα προιόν που έχει όνομα μοντέλου ιδιο με αυτό που έδωσε ο admin.

Οταν ο αριθμός προιόντων που δώσαμε ισούται με τον αριθμό που κλήθηκε και βρήκε/διέγραψε η κλάση, τότε η διαγραφή σταματά και βγαίνει ώς έξοδο στο console, ενα μήνυμα επιτυχίας(**Figure: 35**)

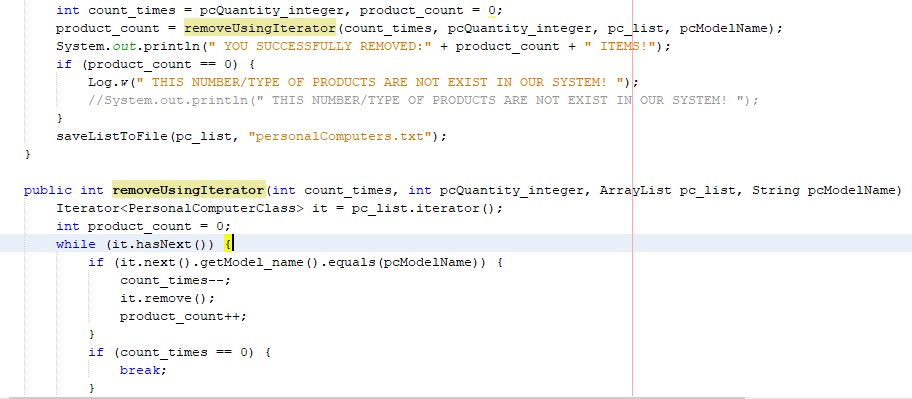


Figure 35: Διαγραφή ποσότητας προιόντων απο arrayList

Τέλος πρέπει να αποθηκευτεί η νέα και ανανεωμένη arrayList των ηλεκτρονικών υπολογιστών στο αρχείο με τη χρήση της κλάσης saveListToFile(**Figure:36**)

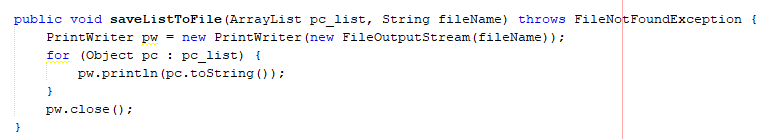


Figure 36: Αποθήκευση της νέας arrayList στο αρχειο

Ομως ας δούμε μια διαγραφή(το πώς ηταν και πως αλλαξε το αρχειο). Αρχικά το αρχείο περιείχε τα εξής προιόντα(**Figure: 37**)

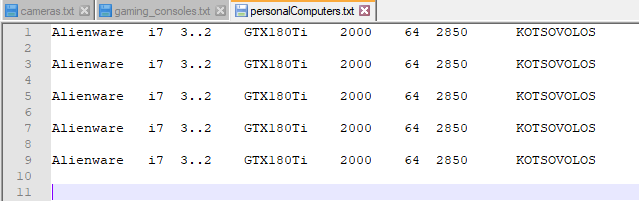


Figure 37: Αρχειο personalComputers πρίν τη διαγραφή

Τελικά το αρχείο αλλαξε(**Figure:38**)

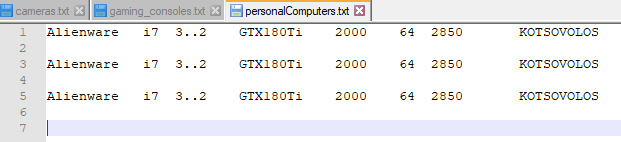


Figure 38: Αρχειο personalComputers μετα την διαγραφή

### 4.2.1.4 PRINT CUSTOMERS

Με παρόμοιο τρόπο που εκτυπώσαμε τα προιόντα μπορούμε απλά να εκτυπώσουμε τους πελάτες του καταστήματος. Αρχικά φορτωση του αρχειου των πελατών, ανακατασκευή της arrayList και στη συνέχεια εκτύπωση με JTextArea και JScrollPane(**Figure: 39**)

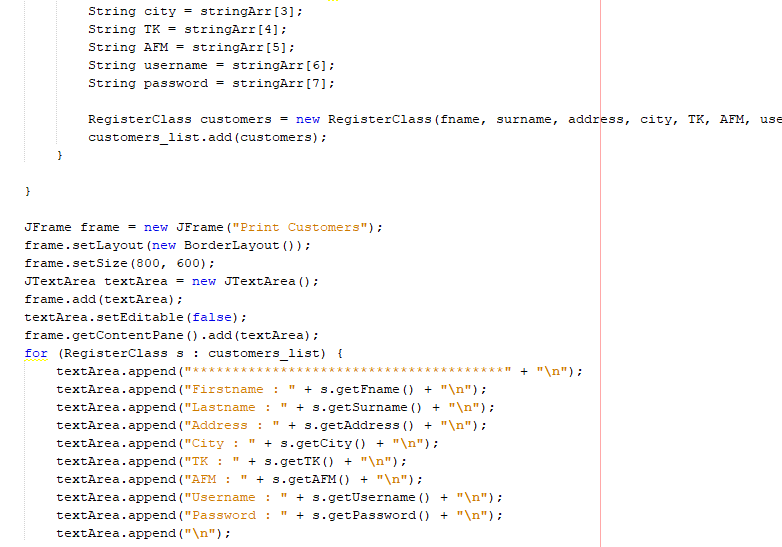


Figure 39: Εκτύπωση πελατών(κομμάτι κώδικα)

Οσο αφορά την εκτύπωση, πατώντας το κουμπί PRINT CUSTOMERS που βρίσκεται στο κεντρικό πανελ του admin μπορουμε να δούμε τους πελάτες του καταστήματος(**Figure: 40**)

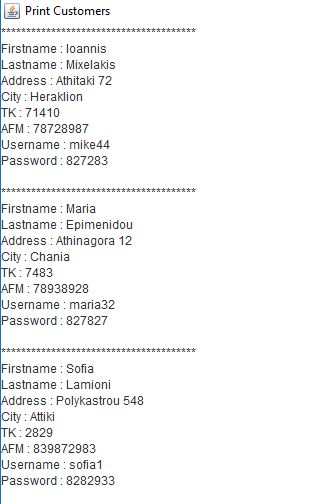


Figure 40: Εκτύπωση πελατών

### 4.2.1.5 ADD COMPANY

Η προσθήκη μιας εταιρείας γίνεται όπως ακριβώς η προσθήκη προιόντων. Απλά αντί να ανοίξουμε 5 αρχεια, ανοίγουμε μόνο το αρχείο των εταιρειών ανακατασκευάζουμε το arrayList των εταιρείων, προσθέτουμε μια εταιρεία, και κάνουμε με κατάλληλη κλάση(save) στο αρχειο. Πιο συγκεκριμένα, το JFrame προσθήκης μιας εταιρείας παρουσιάζεται παρακάτω(**Figure: 41**)

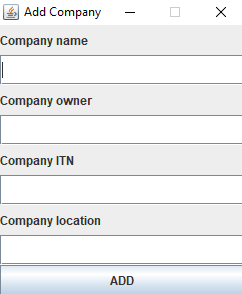


Figure 41: JFrame προσθήκης εταιρείας στο κατάστημα

Αρχικά ανακατασκευή της arrayList των εταιρείων(**Figure: 42**)

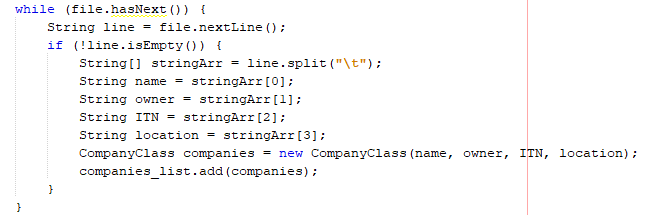


Figure 42: Ανακατασκευή της arrayList των εταιρειων

Ενα σημείο που πρέπει να προσέξουμε εδώ ειναι οτι **ελέγχουμε για το εαν υπάρχει ήδη μια εταιρεία στο σύστημα μας** με τον ίδιο αριθμό φορολογικού μητρώπου(I.T.N). Αυτό γινεται με το εαν μετρήσουμε πόσες εταιρείες υπάρχουν με ιδιο ITN σύμφωνα με τον αριθμό ITN που δώσαμε για την προσθήκη νεας εταιρειας.

Εαν δεν υπαρχει αλλη εταιρεια τοτε καλουμε τη κλάση writeCompany αλλιως βγάζουμε ενα μήνυμα προειδοποίησης και δεν γινεται προσθήκη της εταιρειας (**Figure: 43**)

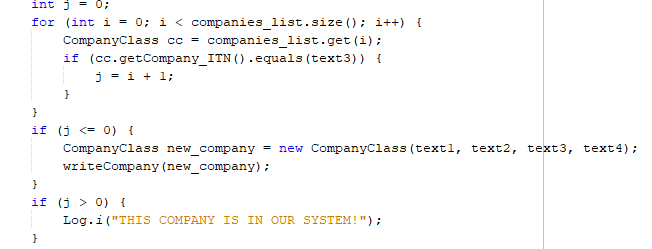


Figure 43: Ελεγχος για υπαρξη όμοιας εταιρείας στο σύστημα

Τελος γράφουμε με τον παραδοσιακό τρόπο στο σύστημα μας την εταιρεία(**Figure:44**)

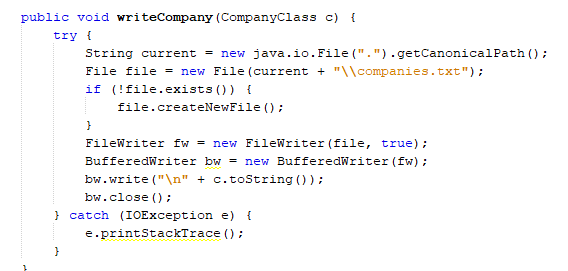


Figure 44: Προσθήκη εταιρείας στο σύστημα

Τώρα θα προσθέσουμε μια εταιρεία στο σύστημα. Το αρχικο αρχείο έχει την εξής μορφή(**Figure: 45**)

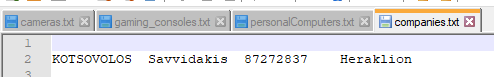


Figure 45: Αρχειο εταιρείων πρίν την προσθήκη νεας εταιρείας

Εαν προσθέσουμε την παρακάτω εταιρεία(**Figure: 46**)

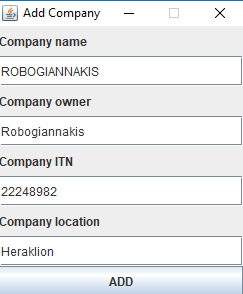


Figure 46: Προσθήκη εταιρείας στο αρχειο

Το αρχείο θα αλλάξει(**Figure: 47**)

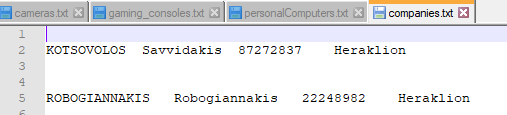


Figure 47: Αρχειο μετά την προσθήκη εταιρείας

### 4.2.1.6 REMOVE COMPANY & PRINT COMPANIES

Παρομοια με τις προηγούμενες διαγραφές, γίνεται και για την διαγραφή μιας εταιρείας. Οπως πάντα πρέπει να ανακατασκευασουμε τη λίστα και στη συνέχεια με τη χρήση του ονόματος της να την αφαιρέσουμε απο τη λίστα, και στη συνέχεια να αποθηκευσουμε ξανά το αρχειο.

Η εκτύπωση ειναι ακριβώς ίδια οπως με την εκτύπωση των προιόντων και των πελατών απλά αλλαξαμε τα αρχεία αλλα και τους τρόπους ανακατασκευάσματος των arrayLists που στη συνέχεια ως γνωστον θα εκτυπωθούν σε ενα JTextArea με χρηση JScrollPane

### 4.2.1.7 FINISH CUSTOMER ORDER

Ο κώδικας εδώ ειναι **αρκετά πιο περίπλοκος** και αξίζει να τον περιγράψουμε όσο πιο αναλυτικά γίνεται. Αρχικά δημιουργούμε ενα JFrame , JPanel αλλα και ενα κουμπί το CONFIRM. Επίσης ενα JTextField(με το κατάλληλο JLabel) το οποίο θα περιέχει το username του πελάτη. Έστω οτι ο πελάτης έχει κανει μια παραγγελια.

O actionListener θα πάρει την τιμή του username και θα το περάσει στη κλάση **reconstructTemporaryAndNoTemporaryFilesArrayListsPlusCalculateOrderCost** η οποία σκοπό έχει να κανει ανακατασκευασμό των temporary files.

*Τα temporary μας βοηθούν ως καλάθι αγορών του πελάτη καθώς οταν ο πελάτης κανει μια παραγγελια τα προιόντα γράφονται στα εκάστωτε temporary αρχεια προιόντων τα οποία φυσικά αδειάζουν οταν ο admin οριστικοποιήσει την παραγγελια ωστε ο πελάτης μελλοντικά να μπορέσει να ξανα-αγοράσει προιόντα.*

Η παραπάνω κλάση αφού κανει οτι πρέπει, καλεί την κλάση **calculateOrderCost** με σκοπό να υπολογίσει το τελικό κόστος των παραγγελιων και να το επιστρέψει στο σημείο που έγινε το event του actionListener. Πολύ απλά, θεωρούμε οτι το αρχικό κόστος της παραγγελιας ειναι μηδέν αλλα κάθε φορά που βλέπουμε σε κάθε αρχείο ενα προιόν με το συγκεκριμένο username το προσθέτουμε στο συνολικό κόστος. Κατόπιν καλούμε τη κλάση **reconstructBasicProductFiles**.

Μεσα σε αυτήν καλούμε μια κλάση που θα διαγράψει τα προιόντα απο τα βασικά αρχεία καλώντας διαδοχικά τις συναρτήσεις:

* deleteCamerasFromNonTemporary
* deleteConsolesFromNonTemporary
* deleteMobilesFromNonTemporary
* deletePCSFromNonTemporary
* deleteTVSFromNonTemporary

Σε κάθε μια απο αυτές τις συναρτήσεις καλουνται με την σειρά οι συναρτήσεις αποθήκευσης των νεων arrayLists στα αρχεία(βασικά αρχεία που περιέχουν τα προιόντα). Δηλαδή θέλουμε εκτός απο το να σβηστούν τα προιόντα (στη συνεχεια θα εξηγηθει) απο το καλάθι αγορών αλλα να σβηστούν και απο το κατάστημα με σκοπό ο admin να προσθέσει μελλοντικά νέα προιόντα:

* saveCamerasListToFile
* saveConsolesListToFile
* saveMobilesListToFile
* savePCSListToFile
* saveTVSListToFile

**Η διαγραφή απο τα temporary files**: Οταν ολες οι παραπάνω συναρτήσεις που περιγράψαμε τελειώσουν την εργασία τους, γυρνάμε πίσω στο event του actionListener και και εχοντας το username του πελάτη απλά αδειάζουμε τα αρχεια χρησιμοποιώντας το **fi**l**e.delete();** και **file.createNewFile();** Οπου file ειναι το όνομα αρχείου που θέλουμε να διαγράψουμε και να δημιουργήσουμε ξανά(κενό αρχείο).

Για να μπορέσουμε να εκτυπώσουμε μια παραγγελία πρέπει αρχικά πρώτα να την οριστικοποιήσουμε. Οποτε δημιουργήσαμε τη κλάση **createAnOrder** που παίρνει ως ορίσματα το username και το τελικό κόστος που υπολογίσαμε με την κατάλληλη αλληλουχία συναρτήσεων ανοίγουμε το customers.txt για να πάρουμε τα βασικά στοιχεία του πελάτη που έχει το συγκεκριμένο username( ειναι μοναδικό για κάθε χρήστη) και στη συνέχεια δημιουργούμε μια arrayList με τα στοιχεία του πελάτη και το κόστος και τα αποθηκευουμε στο αρχειο customer\_orders.txt(η αποθηκευση γινεται με την βοήθεια της κλάσης **writeCustomerOrder**)

Για περισσότερη αληθοφάνεια δημιουργήσαμε τη κλάση **getTimeZone** που απλά βασιζόμενοι σε μια γεννήτρια τυχαίων αριθμών επιλέγουμε ως ενα string μια ζώνη ώρας

Φυσικά κάθε παραγγελία έχει **διαφορετικό** order\_id που επιλέγεται και αυτό **τυχαία.** Ο κώδικας ειναι αρκετά μεγάλος για να μπορέσει να παρουσιαστεί με εικόνες(κοιτάξτε το αρχείο **FinishCustomerOrder.java**)

### 4.2.1.8 PRINT FINISHED ORDERS

Η εκτύπωση παραγγελιων είναι παρόμοια με την εκτύπωση προιόντων, πελατών αλλα και εταιρείων. Ανοιγμα του αρχείου και στη συνέχεια ανακτασκευή της λίστας και εκτύπωση με χρήση JTextArea και JScrollPane. Μέρος του κώδικα φαίνεται παρακάτω(**Figure: 48**)

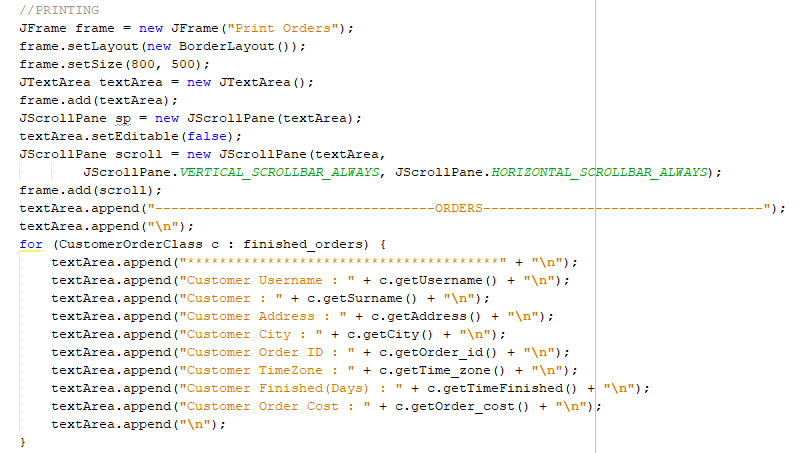


Figure 48: Εκτύπωση παραγγελιων

### 4.2.1.9 LOGOUT

Έξοδος απο το σύστημα κανοντας αυτόματα dispose το frame της κλάσης

## 4.3 Η ΠΛΕΥΡΑ ΤΟΥ ΠΕΛΑΤΗ ( CLIENT SIDE )

### 4.3.1 PRINT PRODUCTS

Για να εκτυπώσουμε προιόντα σε αυτή τη περίπτωση επιλέξαμε είτε να εκτυπωθούν συνολικά σε ενα JtextArea ειτε να εκτυπωθούν ανα είδος. Ουσιαστικά στον πελάτη εμφανίζεται το εξής frame εκτυπώσεων(**Figure: 49** )

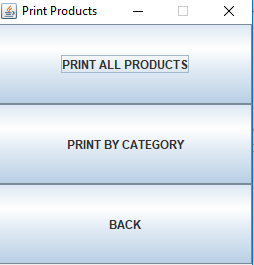


Figure 49: Δυο τρόποι εκτυπώσεων για τον πελάτη

Απλά δημιουργήσαμε ενα JFrame μεσα στο οποίο αφού προσθέσαμε τα κατάλληλα JButtons(**Figure: 50**)

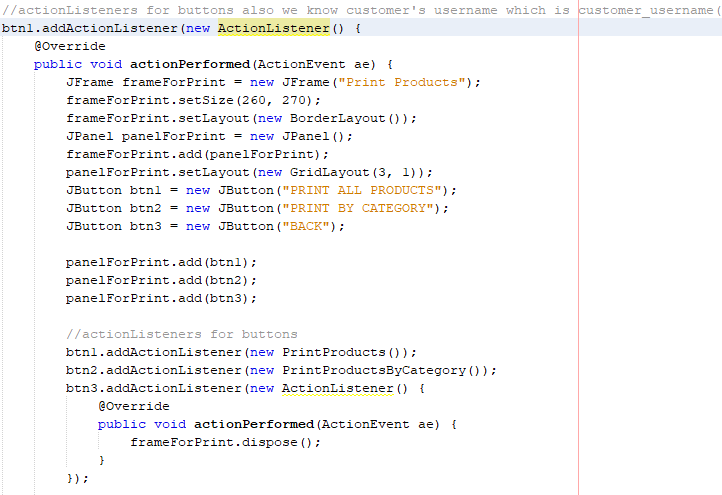


Figure 50: Κώδικας υλοποίησης JFrame εκτυπώσεων

Η κλάση PrintProducts ανοιγει ολα τα αρχεία, ανακατασκευάζει τις κατάλληλες arrayList και μεσω της χρήσης JTextArea και JScrollPane εκτυπώνει στην οθόνη. Οπως δηλαδή καναμε κα στις εκτυπώσεις ολων των προιόντων στον admin. Ομως ο πελάτης πιθανόν να θέλει να δεί τα προιόντα ξεχωριστά ανα είδος. Οποτε μόλις μεταβούμε στη κλάση PrintProductsByCategory μας ανοίγει το παρακάτω frame(**Figure: 51**).

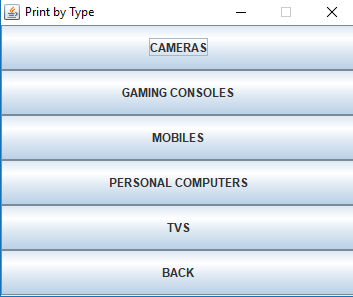


Figure 51: Εκτυπωση προιόντων ανα ειδος

Ο κώδικας φαίνεται παρακάτω(**Figure: 52**)

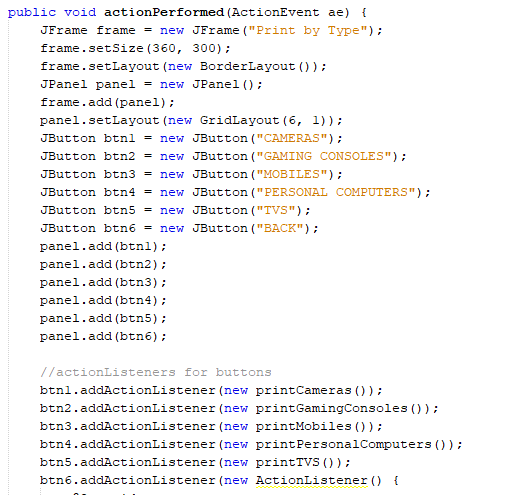


Figure 52: Κλήση συναρτήσεων για εκτυπωση διαφορετικών προιόντων

Για παράδειγμα εαν θέλουμε να εκτυπώσουμε τηλεοράσεις τότε μεσω του κατάλληλου κουμπιού μεταβαινουμε στη κλάση **printTVS** . Αρχικά ανοίγουμε το αρχειο tvs.txt , κατόπιν δημιουργούμε ξανά την arrayList με τις τηλεοράσεις. Επιπρόσθετα δημουργούμε ενα JFrame, ενα JTexτArea και ενα JScrollPane. Τελος χρησιμοποιώντας την arrayList και ενα αντικειμενο τύπου TVClass μπορουμε με κατάλληλη μορφοποίηση να παρουσιάσουμε τις τηλεοράσεις στην οθόνη(**Figure: 53**)

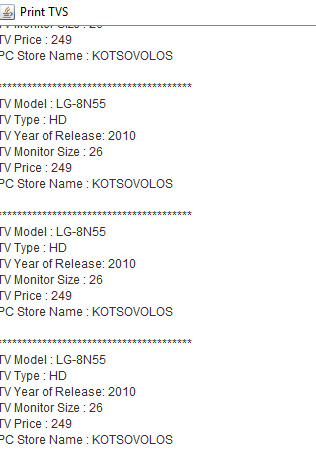


Figure 53: Εκτύπωση τηλεοράσεων

Ενώ ο κώδικας φαινεται παρακάτω(**Figure: 54**)

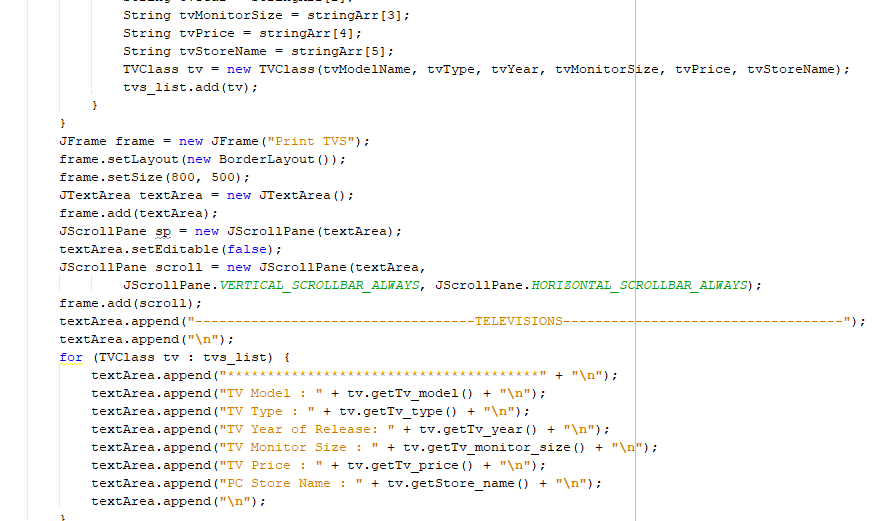


Figure : Κώδικας εκτύπωσης τηλεοράσεων

### 4.3.2 SHOPPING BASKET

Οσο αφορά το καλάθι αγορών αρχικά μεταβαίνουμε στη κλάση ShoppingBasket η οποία παίρνει ώς όρισμα το username του πελάτη που έχει εισέλθει στο σύστημα. Η κλαση αυτή υλοποιεί ενα JFrame που περιέχει 3 κουμπιά. Ενα που που υλοποιεί την λειτουργία προσθήκη στο καλάθι, ενα αλλο που εκτυπώνει το καλάθι αγορών και ενα τελευταίο που απλά ειναι το dispose του frame αυτου. Το JFrame φαινεται παρακάτω(**Figure: 55**)



Figure 55: JFrame για το καλάθι αγορών

Ενώ το frame της προσθήκης στο καλάθι αγορών(**Figure: 56**)

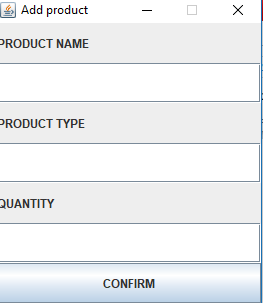


Figure 56: Προσθήκη προιόντος στο καλάθι αγορών

Εδώ, εχοντας λαβει το μοντελο του προιόντος, το είδος και την ποσότητα κάθε φορα με ενα κατάλληλο if μεταβαινουμε στις συναρτήσεις:

* shoppingBasketProccessingCamerasProducts
* shoppingBasketProccessingConsolesProducts
* shoppingBasketProccessingMobilesProducts
* shoppingBasketProccessingPCSProducts
* shoppingBasketProccessingTVSProducts

Για παράδειγμα η συνάρτηση shoppingBasketProccessingPCSProducts ανοιγει το αρχείο personalComputers.txt , δημιουργεί ξανά τη arrayList απο το αρχειο και κάνοντας τον κατάλληλο ελεγχο για το εαν το προιόν υπάρχει στο κατάστημα ή και οχι . Τελικα μέσω αυτων των συναρτήσεων καλουνται οι παραάτω συναρτήσεις για τελική αποθηκευση του προιόντος που δημιουργήθηκε:

* writeCamerasToFileUsingCustomerUsername
* writeConsolesToFileUsingCustomerUsername
* writeMobilesToFileUsingCustomerUsername
* writePCSToFileUsingCustomerUsername
* writeTVSToFileUsingCustomerUsername

### 4.3.3 LOGOUT

Έξοδος απο το σύστημα κάνοντας dispose το κεντρικό JFrame του customer panel.

# 5. ΜΕΛΛΟΝΤΙΚΟ ΕΡΓΟ( FUTURE WORK)

## 5.1 Η ΧΡΗΣΗ ΤΗΣ SQL ΔΙΝΕΙ ΕΥΚΟΛΟΤΕΡΗ ΥΛΟΠΟΙΗΣΗ

Η εφαρμογή μας **στηρίχτηκε καθαρά στην χρήση .txt αρχείων** για την αποθήκευση των δεδομένων των χρηστών αλλα και των προιόντων τ ου υποκαταστήματος. Θα μπορούσαμε να ειχαμε χρησιμοποιήσει βάσεις δεδομένων όπως ειναι για παράδειγμα η MySQL ή η PostgreSQL. Χρησιμοποιώντας μια σχεσιακή βάση δεδομένων τα πράγματα ειναι πολύ πιο απλά και ώς προς το κομμάτι του κώδικα αλλα και πρός την υλοποίηση του απο τον προγραμματιστή(MySQL)

Για παράδειγμα θα μπορούσαμε αντί για ενα .txt χρηστών( παρόμοιως για όλα τα .txt της εφαρμογής μας) να είχαμε δημιουργήσει μια βάση δεδομένων είτε χειροκίνητα κατεβάζοντας την εφαρμοή MySQL Community είτε μέσω της MySQL Workbench πολύ απλά γράφοντας την παρακάτω εντολή στη μορφή MySQL :

CREATE DATABASE users;

Οπου το users ειναι το όνομα του πίνακα που θέλουμε να δημιουργήσουμε ενώ το CREATE DATABASE ειναι δεσμευμένες εντολές της SQL για δημιουργία μιας βάσης. Για να την χρησιμοποιήσουμε πολύ απλά γράφουμε την εντολή(MySQL):

USE users;

Μεσα στη βάση μπορούμε για παράδειγμα να δημιουργήσουμε διάφορα tables. Για παράδειγμα ενας user εχει ονοματεπώνυμο, αριθμό τηλεφώνο, password και username. Αυτό για παράδειγμα θα μπορούσε να γίνει ώς εξής: (MySQL)

CREATE TABLE user

(

id INT unsigned NOT NULL AUTO\_INCREMENT, # Unique ID for the record

fname VARCHAR(150) NOT NULL, # First name of the user

surname VARCHAR(150) NOT NULL, # Surname of the user

telephone VARCHAR(10),NOT NULL # Birthday of the cat

password VARCHAR(30),NOT NULL # Password of the user

PRIMARY KEY (id) # Make the id the primary key

);

Το INT χρησιμοποιείται για να δηλώσουμε ακεραίους, το varchar συμβολοσειρές ενώ σε κάθε table χρειαζόμαστε ενα μοναδικό κλειδί οπου εδώ ειναι το id του χρήστη. Με παρόμοιο τρόπο θα μπορούσαμε να φτιάξουμε βάσεις και κατάλληλα tables σε κάθε βάση για κάθε .txt αρχείο μας όπου με τη χρήση της SQL αλλα και της Java να δημιουργούσαμε μια πιό σύγχρονη και εύκολη προς υλοποίηση εφαρμογή(MySQL).

## 5.2 Η ΣΥΝΔΕΣΗ ΤΗΣ SQL ΜΕ ΤΗΝ JAVA

Αρχικά πρεπει να κατεβασουμε τον κατάλληλο driver για να γινει η συνδεση μας πχ θα μπορούσαμε να κατεβάσουμε τον JDBC Driver της SQL απο το παρακάτω link:

<https://www.microsoft.com/en-us/download/details.aspx?id=11774>

Στη συνέχεια να **τον κάνουμε add** στις βιβλιοθήκες μας πχ στο Netbeans. Κατόπιν με τις παρακάτω εντολές έχουμε σύνδεση στη βάση μας :

public static void main (String[] args) throws Exception{

Connection conn=null;

String url="jdbc:jtds:sqlserver://YourServerIp:1433/dbName";

String username="some\_username";

String password=”some\_password";

String driver="net.sourceforge.jtds.jdbc.Driver";

Class.forName(driver);

conn= DriverManager.getConnection(url, username, password);

}

Οπου dbName ειναι το ονομα της βάσης μας ενώ username, password ειναι τα κατάλληλα username και password που χρειάζονται για την είσοδο μας στη βάση μας. Κάνουμε την σύνδεση κάθε φορά που τη χρειαζόμαστε (πχ σε ανάλογη κλάση) και στη συνέχεια γράφουμε τα κατάλληλα υπερωτήματα( queries) για να πάρουμε ή να στείλουμε μια πληροφορία στη βάση μας, σε συγκεκριμένο table , όπου αυτό χρειάζεται.

## 5.3 SQL – JAVA ΥΠΕΡΩΤΗΜΑΤΑ(QUERIES)

Για να κάνουμε ενα κατάλληλο query πρέπει να κάνουμε τα παρακάτω βήματα(Alvin, 2018):

* Αρχικά δημιουργούμε ενα JDBC ResultSet αντικείμενο
* Εκτελούμε το SQL query SELECT
* Λαμβάνουμε τα κατάλλληλα αποτελέσματα

Εαν για παράδειγμα θέλουμε να βρούμε τα προιόντα που είναι τύπου υπολογιστή και έχουν τύπο μοντέλου Alienware και βρίσκονται στην εταρεία Κωτσόβολος κάνουμε τα εξής παρακάτω(Alvin, 2018):

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT Alienware FROM personalComputers WHERE company=KOTSOVLOS" )

Με παρόμοιο τρόπου μπορούμε θέσουμε ερωτήματα στη βάση εφόσον φυσικά είναι συνδεμένη με την εφαρμογή μας και να πάιρνουμε τις κατάλληλες απαντήσεις. Μέσω αυτών των κατάλληλων απαντήσεων και πράξεων μπορούμε να στείλουμε τις αλλαγμένες τιμές ή νεες τιμές πίσω στη βάση.

Η διαδικασία αποστολής δεδομένων πίσω στη βάση χρειάζεται και 2 βήματα(Alvin, 2018):

* Δημιουργούμε ενα Java statement αντικείμενο
* Εκτελούμε ενα SQL query τύπου INSERT διαμέσου του JDBC αντικειμένου

Οποτε για το παραπάνω παράδειγμα η εισαγωγή ενός Alienware στο κατάλληλο table της βάσης μας γίνεται ώς εξής(Alvin, 2018):

Statement = conn.createStatement();

statement.executeUpdate("INSERT INTO personalComputers " + "VALUES (Alienware, i7, 3.2, GTX180Ti, 2000, 64, 2850, KOTSOVOLOS)");

# ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη σημερινή εποχή ειναι απαραίτητο να μπορούμε οσο το δυνατόν να γλιτώνουμε περιττό χρόνο μιας καθώς ζούμε σε πολύ γρήγορους ρυθμούς ζωής. Οπότε η χρησιμότητα εφαρμογών που σκοπευούν στην αγοραπωλησία προιόντων και υπηρεσίων εξ αποστάσεως είναι μια καλή ιδέα ώστε ο μέσος άνθρωπος να μπορέσει έστω να γλιτώσει την ταλαιπωρία επίσκεψης του σε κάποιο κατάστημα. Επομένως, η αγορά απο το διαδίκτυο είναι γρήγορη, έυκολη και αρκετά αξιοπιστη.

Με το πάτημα ενός κουμπιού μπορεί η παραγγελία σου να έρθει στο σπίτι, στο γραφείο σου χωρίς περιττές βόλτες και το βαναυσο χάσιμο χρόνου. Ενα όμως θέμα σοβαρό που θίγεται οσο αφορά τα ηλεκτρονικά καταστήματα είναι η ασφάλεια τους. Δίνοντας τα προσωπικά σου στοιχεία και πιθανόν τον αριθμό της χρεωστικής/πιστωτικής σου κάρτας, δεν μπορείς ποτέ να εισαι 100% σίγουρος οτι θα είσαι ασφαλής. Οπότε, κύριο μέλημα των επιστημόνων πληροφορικής που ασχολούνται με τέτοια θέματα είναι να μπορέσουν να αυξήσουν την ασφάλεια στα ηλεκτρονικά καταστήματα.

# ΑΝΑΦΟΡΕΣ

* Chisholm, H. (1911). Commerce- Encyclopædia Britannica. 6 (11th). Cambridge University Press
* Khurana, A.(2017).Advantages of E-commerce Over Traditional Retail. Retrieved from [here](https://www.thebalancesmb.com/advantages-of-ecommerce-1141610) [accessed 22 April 2018]
* Khurana, A.(2018).Disadvantages of E-commerce. Retrieved from [here](https://www.thebalancesmb.com/disadvantages-of-e-commerce-1141571) [accessed 22 April 2018]
* Manthena, P.(2013). Connecting a Java application to an SQL database with Eclipse, Stackoverflow. Retrieved from [here](https://stackoverflow.com/questions/16961342/connecting-a-java-application-to-an-sql-database-with-eclipse) [accessed 18 June 2018]
* MySQL, Creating and Selecting a Database. Retrieved from [here](https://dev.mysql.com/doc/refman/8.0/en/creating-database.html) [ accessed 18 June 2018]
* MySQL, Some Basic Operations with MySQL. Retrieved from [here](https://dev.mysql.com/doc/mysql-getting-started/en/#mysql-getting-started-basic-ops) [ accessed 18 June 2018]
* Wienclaw, Ruth A. (2013).E-Commerce. Research Starters.
* Wills, B.(2014).18 Major Benefits of e-Commerce Business for Retailers & Customers 2018. Retrieved from [here](https://medium.com/@briannawillsss/18-major-benefits-of-e-commerce-business-for-retailers-customers-in-2015-63c5fc87f679) [accessed 22 April 2018]
* Alvin, A.(2018). Java JDBC: A SQL SELECT query example. Retrieved from [here](https://alvinalexander.com/java/edu/pj/jdbc/jdbc0003) [ accessed 22 June 2018]
* Alvin, A.(2018). Java JDBC Insert Example: How to insert data into a SQL table. Retrieved from [here](https://alvinalexander.com/java/edu/pj/jdbc/jdbc0002) [accessed 22 June 2018]