

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ**



**Πολυτεχνική Σχολή**  
**Τμήμα Μηχανικών Η/Υ και Πληροφορικής**  
**Τεχνολογία Λογισμικού**

**CIRCUIT HEAVEN** 

**Project-code-v1.0**

Ιούνιος 2023

## Μέλη

Αβραμόπουλος Γεώργιος - AM 1070772 - 5<sup>ο</sup> έτος [up1070772@upnet.gr](mailto:up1070772@upnet.gr)

Παναγιώτης Σταθόπουλος - AM 1070928 – 5<sup>ο</sup> έτος [up1070928@upnet.gr](mailto:up1070928@upnet.gr)

Βασιλική Πούπουζα - AM 1080449 - 4<sup>ο</sup> έτος [up1080449@upnet.gr](mailto:up1080449@upnet.gr)

Αναστασία Νάτση - AM 1080702 - 4<sup>ο</sup> έτος [up1080702@upnet.gr](mailto:up1080702@upnet.gr)

## Github

- Κλάση **main**:

`Standardbuyers()`: ορίζουμε εξ αρχής κάποιος αγοραστές και τον ιδιοκτήτη του καταστήματος

`login()`

`register()`

- κλάση **ShoppingCart**:

`viewcart()`: ελέγχει τα προϊόντα που έχεις στο καλάθι

`showcart()`: εμφανίζει τα προϊόντα που έχεις στο καλάθι

`CalculateArrivingTime()`: υπολογίζει τον χρόνο παράδοσης

`checkout()`: αγορά προϊόντος

`calculatenet()`: υπολογίζει το συνολικό κόστος μαζί με Κούριερ εκπτώσεις

`couriercost()`: υπολογίζει το κόστος του courier

`category()`: κοιτάει την κατηγορία του αγοραστή

`clearcart()`: αδειάζει το καλάθι

- κλάση **Pencil, Pen, Notebook, Paper**

Pencil(), Pen(), Notebook(), Paper() : χαρακτηριστικά

getDetails(): τα συνολικά χαρακτηριστικά

- Κλάση **Owner**:

Owner(): επιλογές που μπορεί να κάνει ο ιδιοκτήτης

- Κλάση **eshop**:

addNewItem(): προσθήκη προϊόντος

removeItem(): αφαίρεση προϊόντος

showItem(): εμφάνιση προϊόντος

addbuyer(): προσθήκη καινούριου αγοραστή

removebuyer(): αφαίρεση αγοραστή

updateitemstock(): ανανεώνει το απόθεμα των προϊόντων

checkstatus(): παίρνει την μέθοδο status() από την κλάση Buyer.

- Κλάση **Buyer**:

status(): εμφανίζει την κατάσταση των χρηστών(όνομα χρήστη, κατηγορία, ποσότητα διαφορετικών αντικειμένων).

choice(): Μια Switch case για το τι θέλει να κάνει ο αγοραστής

store(): δείχνει τα προϊόντα όλα μαζί

bonusSetCategory(): ορίζει την κατηγορία του αγοραστή βάσει το συνολικό κόστος των αγορών του.

- Κλάση **Service**:

sendQuestionnaire(): ερωτήσεις σχετικά με το προϊόν

askDeviceReplacement(): ερώτηση για αντικατάσταση προϊόντος

fillServiceForm(): ενημέρωση φόρμας

updateRepairStatus(): ενημέρωση την κατάσταση επισκευής

setNextOrderDiscountCode(): κωδικός έκπτωσης

printServiceInfo(): αναλυτική παρουσίαση

- κλάση **TrackDelivery**:

goTo():επιλέγει κατηγορία

find():εύρεση των παραγγελιών

show():εμφάνιση αναζητούμενου

pickDelivery():επιλέγει παραγγελία

returnItem():επιστροφή αποτελέσματος αναζήτησης

track():παρακολούθηση παραγγελίας

checkUpdates():ελέγχει για αλλαγές

isDelivered():πραγματοποίηση αποστολής παραγγελίας

save():αποθηκεύει την αξιολόγηση της παραγγελίας

- κλάση **ArrivingProduct**

ArrivingProduct(String estimatedArrival): constructor της κλάσης

displayPrompt(): εμφανίζει το estimatedArrival

- Κλάση **AuctionPage**

displayPrompt(List<Description> productList): κύρια μέθοδος List<Description>: επιτρέπει στον χρήστη να δώσει input για διαφορετικές επιλογές (Customer, Seller, Invalid choice/input)

ProductPage(productList).displayPrompt();: στην επιλογή Customer δημιουργεί ένα νέο αντικείμενο της κλάσης ProductPage και καλεί την displayPrompt() με παραμετρο constructor το productList στο αντικείμενο της κλάσης ProductPage

Form form = new Form();: στην επιλογή Seller δημιουργεί ένα νέο αντικείμενο της κλάσης Form

Description productDescription = form.fillForm();: καλείται η μέθοδος fillForm() της κλάσης Form, επιστρέφει ένα αντικείμενο Description το οποίο αντιστοιχίζεται στην μεταβλητή productDescription

Products products = new Products();: δημιουργούμε ένα νέο αντικείμενο της κλάσης Products

products.addProduct(productDescription): καλούμε την addProduct() μέθοδο από την κλάση Products και περνάμε την μεταβλητή productDescription ως ορισμό

- κλάση **Description**

Description(int productId, String productName, String description, double startingBid): είναι ο constructor της κλάσης, κάνει initialize τα πεδία που αναφέρονται

getProductId(), public String getProductName(), public String getDescription(), public double getStartingBid(), public double getCurrentBid(): επιστρέφουν τα αντιστοιχικά στοιχεία από το αντικείμενο Description

setCurrentBid(double bid): θέτει το currentBid του αντικειμένου Description ως την τιμή που δηλώνει. (Δεν το έχω καταφέρει να δουλέψει)

- Κλάση **Form**

public Description fillForm(): δίνει prompt στον χρήστη για να συμπληρώσει την φόρμα

- Κλάση **HomePage**

do { choice = sc.nextLine(); } while (!choice.equals("AuctionPage"));: την χρησιμοποιούμε για να σιγουρευτούμε ότι ο χρήστης θα μεταβεί στην AuctionPage

Products products = new Products(): δημιουργούμε ένα νέο αντικείμενο της κλάσης Products και χρησιμοποιούμε την μεταβλητή products για να αναφερθούμε σε αυτό το αντικείμενο

List<Description> productList = products.getProductList(): καλούμε την getProductList() από την κλάση Products για να εμφανίσουμε μια λίστα με αντικείμενα

AuctionPage.displayPrompt(productList): εμφανίζει την AuctionPage, μεταβιβάζοντας και την productList

- Κλάση **ProductPage**

ProductPage(List<Description> productList): constructor της κλάσης με παραμετρο List<Description>. Κάνει initialize την productList και δημιουργεί νέο instance της κλάσης HashMap για να κάνει initialize το bidMap

displayPrompt(): εμφανίζει το prompt της ProductPage και χειρίζεται το user interaction

ArrivingProduct arrivingProduct = new ArrivingProduct(estimatedArrival): δημιουργούμε ένα νέο αντικείμενο της κλάσης ArrivingProduct με ορίσμο το estimatedArrival

arrivingProduct.displayPrompt(): καλούμε την αντιστοιχη μεθοδο του αντικειμενου  
arrivingProductbidMap.put(chosenProduct.getProductid(), bid): θεωρητικά προσθετει ενα entry στο bidMap με το productid ως key και το bid ως το value. Δεν το εχω καταφερει να αποθηκευει το bid ακομα

- κλάση **Products**

`Products()`: constructor της κλάσης. Κάνει initialize `productList` ως ένα νέο `ArrayList` και καλεί την μέθοδο `loadProducts()`

`loadProducts()`: με την βοήθεια της μεθόδου `FileHelper.readFileLines()` διαβάζει το αντίστοιχο αρχείο και κάνει load τα προϊόντα

`private Description parseDescription(String line)`: αναλύει μια γραμμή εισόδου και δημιουργεί αντικείμενο `Description`. Μετατρέπει τα πειδα στους καταλλήλους τυπούς δεδομένων

`public List<Description> getProductList()`: επιστρέφει την `productList` που είναι μια λίστα από αντικείμενα της `Description`

`addProduct(Description product)`: προσθέτουμε ένα αντικείμενο της `Description` στο `productList` και καλούμε την `saveProducts()` μέθοδο

`saveProducts()`: κάνει save την `productList` στο αντίστοιχο αρχείο/φάκελο. (χρησιμοποιεί `BufferedWriter`, `FileWriter` για να γράψει το αντίστοιχο data στο αρχείο/φάκελο)

`private String formatProduct(Description product)`: κάνει format το data που αναφεράμε προηγουμένως