



1. Γενικές Πληροφορίες

Η εργασία είναι ομαδική (3 φοιτητές/ομάδα). Κάθε ομάδα θα πρέπει να την υποβάλει ηλεκτρονικά μέσω του MOODLE .

Σημειώνεται ότι:

-) Έχετε δυο επιλογές εργασιών (βλ. περιγραφές που ακολουθούν). Η πρώτη επιλογή (ΕΠΙΛΟΓΗ 1) παραδίδεται μέχρι τις 24/05/2019 και προσμετράται με 10% στον τελικό βαθμό. Η δεύτερη επιλογή (ΕΠΙΛΟΓΗ 2) θα πρέπει να παραδοθεί μέχρι τις 12/06/2019 και προσμετράται με 20% στον τελικό βαθμό.
-) Τα προγράμματα θα υποβληθούν σε ένα συμπιεσμένο φάκελος zip ή rar με όνομα AM1_AM2_ AM3 όπου AM οι αριθμοί μητρώου των φοιτητών.
-) Μέσα στο αρχείο θα υπάρχουν τα αρχεία java και class της εφαρμογής του παραδοτέου.
-) Σε περίπτωση που οι ρυθμίσεις του υπολογιστή σας δεν επιτρέπουν τη χρήση ελληνικών ως input ή output, μπορείτε να χρησιμοποιήσετε Greeklish ή Αγγλικά.

2. Περιγραφή Εργασίας

ΕΠΙΛΟΓΗ 1

Περιγραφή Λειτουργιών

Καλείστε να δημιουργήσετε μία εφαρμογή ηλεκτρονικής παραγγελίας φαγητού από καταστήματα εστίασης..

Στην εφαρμογή εξυπηρετούνται 2 τύποι πελατών: ο Εγγεγραμμένος Πελάτης και ο Επισκέπτης Πελάτης.

Όταν το σύστημα εκκινείται δίνεται η δυνατότητα στο χρήστη να κάνει μια από τις παρακάτω ενέργειες (μενού *Σύνδεσης*):

1. Να κάνει εγγραφή ως Εγγεγραμμένος Πελάτης
2. Να κάνει login (με χρήστη email και password) ως Εγγεγραμμένος Πελάτης
3. Να συνεχίσει ως Επισκέπτης Πελάτης

Αν ο χρήστης επιλέξει το **(1)** εμφανίζεται ένας διάλογος που τον καθοδηγεί να εισάγει όλα τα απαραίτητα στοιχεία με σκοπό να δημιουργήσει λογαριασμό στο σύστημα ως Εγγεγραμμένος Πελάτης. Μετά την επιτυχή εγγραφή του νέου Πελάτη εμφανίζεται ξανά το *μενού σύνδεσης* όπου πλέον μπορεί να κάνει login με τα στοιχεία του.

Αν ο χρήστης επιλέξει το **(2)** εμφανίζεται ένας διάλογος που τον καθοδηγεί να εισάγει email και password για να συνδεθεί στο σύστημα. Ο συνδυασμός email και password θα πρέπει να αντιστοιχεί σε κάποιον από τους υπάρχοντες Εγγεγραμμένους Πελάτες για να πραγματοποιηθεί η σύνδεση, σε διαφορετική περίπτωση, εμφανίζει ενημερωτικό μήνυμα ότι ο συνδυασμός δεν είναι ορθός και προτρέπει το χρήστη να δοκιμάσει ξανά. Αν ο χρήστης πατήσει enter key (το πλήκτρο enter) επιστρέφει στο *μενού Σύνδεσης*. Αν ο χρήστης εισάγει σωστό συνδυασμό στοιχείων τότε η εφαρμογή τον καλωσορίζει με το όνομά του και εκτυπώνεται το *Μενού Εγγεγραμμένου Πελάτη*. **(Κάνουμε την υπόθεση ότι κάθε πελάτης χρησιμοποιεί διαφορετικό email κατά την εγγραφή του, άρα ο συνδυασμός είναι μοναδικός)**

Αν ο χρήστης επιλέξει το **(3)** τότε δεν χρειάζεται να κάνει login. Εκτυπώνεται το *Μενού Επισκέπτη Πελάτη*

Μενού Εγγεγραμμένου Πελάτη

Το μενού Εγγεγραμμένου Πελάτη εμφανίζεται σε όσους χρήστες έχουν καταφέρει να συνδεθούν επιτυχώς μέσω του *Μενού Σύνδεσης*. Και προσφέρει τις παρακάτω επιλογές:

1. Εμφάνιση διαθέσιμων Καταστημάτων

Αν ο χρήστης επιλέξει **(2)** τότε ζητείται να εισάγει το όνομα ή τμήμα του ονόματος του/των Καταστήματος/Καταστημάτων) που επιθυμεί να δει τα στοιχεία. Π.χ. αν το σύστημα υπάρχουν τα καταστήματα «Happy Coffee» , «CoffeMania» και «coffeAnytime» και ο χρήστης εισάγει «cof» θα πρέπει να εμφανιστούν στη λίστα αποτελεσμάτων και τα 3 καταστήματα.

Αν ο χρήστης πατήσει το enter key θα πρέπει να τυπωθεί λίστα με το σύνολο των Καταστημάτων χωρίς φίλτρο στην επωνυμία τους.

Η λίστα αποτελεσμάτων εμφανίζει το id, την επωνυμία, τη διεύθυνση και το τηλέφωνο του κάθε καταστήματος.

Μετά την εμφάνιση των αποτελεσμάτων της αναζήτησης (ανεξαρτήτως του τρόπου ή του αριθμού των αποτελεσμάτων) εκτυπώνεται ξανά το *Μενού Εγγεγραμμένου Πελάτη* για να συνεχίσει ο χρήστης την πλοήγηση.

Μενού Επισκέπτη Πελάτη

Στην περίπτωση που ο χρήστης επιλέξει να προηγηθεί ως Επισκέπτης Πελάτης δεν υπάρχει ιστορικό στο οποίο μπορούμε να διατρέξουμε, για αυτό και έχει λιγότερες επιλογές στο μενού του.

1. Εμφάνιση διαθέσιμων Καταστημάτων

Η διαδικασία εμφάνισης διαθέσιμων Καταστημάτων **(1)** είναι όμοια με αυτή του Εγγεγραμμένου Πελάτη.

Περιγραφή Κλάσεων

Το πρόγραμμα θα αποτελείται κατ' ελάχιστον από τις παρακάτω κλάσεις:

-)/ Customer (αφηρημένη κλάση)
-)/ RegisteredCustomer
-)/ VisitorCustomer
-)/ Product
-)/ Shop
-)/ Main (περιλαμβάνει τη main μέθοδο)

Customer

Η κλάση Customer είναι μια αφηρημένη κλάση που περιγράφει τον πελάτη της εφαρμογής.

-)/ Έχει μεταβλητές στιγμιοτύπου ονοματεπώνυμο, τηλέφωνο και δ/ση.
-)/ Κατασκευαστή που αρχικοποιεί βάσει ορισμάτων τις μεταβλητές στιγμιοτύπου και αποθηκεύει τα αντικείμενα που δημιουργούνται σε *λίστα Customer*.
-)/ Τέλος περιέχει την αφηρημένη μέθοδο getMenu. Η getMenu είναι υπεύθυνη για την εκτύπωση του μενού κάθε πελάτη και τη ροή των ενεργειών του.

RegisteredCustomer

Η κλάση RegisteredCustomer περιγράφει τους Εγγεγραμμένους Πελάτες της εφαρμογής, εκείνους δηλαδή που έχουν κάνει επιτυχώς εγγραφή και μπορούν να συνδεθούν σε αυτό μέσω email και password.

-)/ Η RegisteredCustomer επεκτείνει τη κλάση Customer και τα αντικείμενά της περιγράφονται επιπλέον από id (αυτό- αυξανόμενη μεταβλητή), email και password.
-)/ Επίσης είναι υπεύθυνη για την εκτύπωση του μενού του εγγεγραμμένου πελάτη και της ροής που προκύπτει από αυτό.
-)/ Στατική μέθοδο login που λαμβάνει ως όριασμα τα credentials του πελάτη (email και password), κι αναζητά στη λίστα με τους πελάτες αν υπάρχει τέτοιο αντικείμενο. Αν υπάρχει το επιστρέφει, αλλιώς επιστρέφει null.

VisitorCustomer

Η κλάση Customer περιγράφει τους Επισκέπτες Πελάτες της εφαρμογής, εκείνους δηλαδή που επιλέγουν να μην συνδεθούν στο σύστημα με email και password.

- ⌋ Η VisitorCustomer επεκτείνει την Customer.
- ⌋ Επίσης είναι υπεύθυνη για την εκτύπωση του μενού του επισκέπτη πελάτη και της ροής που προκύπτει από αυτό.

Product

Η κλάση Product μοντελοποιεί τα προϊόντα που διατίθενται μέσω της εφαρμογής.

- ⌋ Τα προϊόντα περιγράφονται από id (αυτό-αυξανόμενη μεταβλητή), όνομα (περιγραφή προϊόντος π.χ. «waffle with ice cream») και τιμή ανά τεμάχιο.
- ⌋ Ορίζεται κατασκευαστής που αρχικοποιεί τις μεταβλητές στιγμιότυπου
- ⌋ Ορίζεται μέθοδος toString που επιστρέφει μορφοποιημένο το αντικείμενο προϊόντος

Shop

Η κλάση Shop μοντελοποιεί τα Καταστήματα που συνεργάζονται με την εφαρμογή μας. Κάθε κατάστημα εκτός των χαρακτηριστικών του όπως δ/ση και επωνυμία ορίζει και τα δικά του Προϊόντα προς διάθεση (δείτε κλάση Product). Πιο συγκεκριμένα:

- ⌋ Κάθε Κατάστημα περιγράφονται από id (αυτό-αυξανόμενη μεταβλητή), επωνυμία καταστήματος, διεύθυνση και μια ArrayList από Products. Η ArrayList από προϊόντα είναι απαραίτητη για να ορίσουμε ποια προϊόντα (αντικείμενα προϊόντων) διαθέτει κάθε κατάστημα, καθώς δεν πουλούν όλα τα καταστήματα τα ίδια προϊόντα.
- ⌋ Ορίζεται κατασκευαστής που αρχικοποιεί τις μεταβλητές στιγμιότυπου
- ⌋ Επίσης περιλαμβάνει τις παρακάτω στατικές μεθόδους που σχετίζονται με την αναζήτηση
 - **printAllShops** εκτυπώνει το σύνολο των καταστημάτων της εφαρμογής (συγκεκριμένα για κάθε κατάστημα: id, επωνυμία, διεύθυνση και τηλέφωνο)
 - **filterAndPrintShops** με όρισμα String μεταβλητή: εκτυπώνει το υποσύνολο των καταστημάτων της εφαρμογής που η επωνυμία τους περιέχει το δοθέν String. (συγκεκριμένα για κάθε κατάστημα: id, επωνυμία, διεύθυνση και τηλέφωνο)

Main

Η Main κλάση είναι υπεύθυνη για την εκτέλεση και τη ροή της εφαρμογής. Για να μπορεί να τρέξει η εφαρμογή παράγοντας αποτελέσματα ζητείται να ορισθεί και να κληθεί στατική μέθοδος loadObjects() που σκοπό έχει τη δημιουργία αντικειμένων όλων των κλάσεων ώστε να φαίνεται η λειτουργικότητα.

Στη συνέχεια εκτυπώνεται το Μενού Σύνδεσης κι ανάλογα με τις επιλογές του χρήστη συνεχίζεται η πλοήγηση του ως Εγγεγραμμένος ή Επισκέπτης Πελάτης, με την εκτύπωση των αντίστοιχων μενού

ΕΠΙΛΟΓΗ 2

Η 2^η εκδοχή της εργασίας περιλαμβάνει τις λειτουργίες που ζητούνται στην Επιλογή 1 και την επεκτείνει υλοποιώντας την παραγγελία του πελάτη από συγκεκριμένο κατάστημα. Για αυτό το λόγο θα δημιουργηθεί νέα κλάση «Order» που μοντελοποιεί την παραγγελία του πελάτη από συγκεκριμένο κατάστημα και περιλαμβάνει τα προϊόντα στις ποσότητες που έχει επιλέξει. Το κείμενο με μπλέ χρώμα υποδηλώνει τα σημεία επέκτασης της 2^{ης} επιλογής συγκριτικά με την Επιλογή 1.

Περιγραφή Λειτουργιών

Καλείστε να δημιουργήσετε μία εφαρμογή ηλεκτρονικής παραγγελίας φαγητού από καταστήματα εστίασης..

Στην εφαρμογή εξυπηρετούνται 2 τύποι πελατών: ο Εγγεγραμμένος Πελάτης και ο Επισκέπτης Πελάτης.

Όταν το σύστημα εκκινείται δίνεται η δυνατότητα στο χρήστη να κάνει μια από τις παρακάτω ενέργειες (μενού *Σύνδεσης*):

1. Να κάνει εγγραφή ως Εγγεγραμμένος Πελάτης
2. Να κάνει login (με χρήστη email και password) ως Εγγεγραμμένος Πελάτης
3. Να συνεχίσει ως Επισκέπτης Πελάτης

Αν ο χρήστης επιλέξει το **(1)** εμφανίζεται ένας διάλογος που τον καθοδηγεί να εισάγει όλα τα απαραίτητα στοιχεία με σκοπό να δημιουργήσει λογαριασμό στο σύστημα ως Εγγεγραμμένος Πελάτης. Μετά την επιτυχή εγγραφή του νέου Πελάτη εμφανίζεται ξανά το *μενού σύνδεσης* όπου πλέον μπορεί να κάνει login με τα στοιχεία του.

Αν ο χρήστης επιλέξει το **(2)** εμφανίζεται ένας διάλογος που τον καθοδηγεί να εισάγει email και password για να συνδεθεί στο σύστημα. Ο συνδυασμός email και password θα πρέπει να αντιστοιχεί σε κάποιον από τους υπάρχοντες Εγγεγραμμένους Πελάτες για να πραγματοποιηθεί η σύνδεση, σε διαφορετική περίπτωση, εμφανίζει ενημερωτικό μήνυμα ότι ο συνδυασμός δεν είναι ορθός και προτρέπει το χρήστη να δοκιμάσει ξανά. Αν ο χρήστης πατήσει enter key (το πλήκτρο enter) επιστρέφει στο *μενού Σύνδεσης*. Αν ο χρήστης εισάγει σωστό συνδυασμό στοιχείων τότε η εφαρμογή τον καλωσορίζει με το όνομά του και εκτυπώνεται το *Μενού Εγγεγραμμένου Πελάτη*. **(Κάνουμε την υπόθεση ότι κάθε πελάτης χρησιμοποιεί διαφορετικό email κατά την εγγραφή του, άρα ο συνδυασμός είναι μοναδικός)**

Αν ο χρήστης επιλέξει το **(3)** τότε δεν χρειάζεται να κάνει login. Εκτυπώνεται το *Μενού Επισκέπτη Πελάτη* αλλά κατά την ολοκλήρωση της παραγγελίας θα του ζητηθούν κάποια επιπλέον στοιχεία (όνομα, δ/ση και τηλέφωνο).

Μενού Εγγεγραμμένου Πελάτη

Το μενού Εγγεγραμμένου Πελάτη εμφανίζεται σε όσους χρήστες έχουν καταφέρει να συνδεθούν επιτυχώς μέσω του *Μενού Σύνδεσης*. Και προσφέρει τις παρακάτω επιλογές:

1. Εμφάνιση προηγούμενων παραγγελιών

2. Εμφάνιση διαθέσιμων Καταστημάτων
3. Προβολή Καταλόγου και Παραγγελία από κατάστημα

Αν ο χρήστης επιλέξει **(1)** τότε τυπώνονται σε αριθμημένη λίστα τα στοιχεία των προηγούμενων παραγγελιών του (αν υπάρχουν) : ημερομηνία& ώρα παραγγελίας, επωνυμία καταστήματος, συνολικό κόστος

Αν ο χρήστης επιλέξει **(2)** τότε ζητείται να εισάγει το όνομα ή τμήμα του ονόματος του/των Καταστήματος/Καταστημάτων) που επιθυμεί να δει τα στοιχεία. Π.χ. αν το σύστημα υπάρχουν τα καταστήματα «Happy Coffee» , «CoffeMania» και «coffeAnytime» και ο χρήστης εισάγει «cof» θα πρέπει να εμφανιστούν στη λίστα αποτελεσμάτων και τα 3 καταστήματα.

Αν ο χρήστης πατήσει το enter key θα πρέπει να τυπωθεί λίστα με το σύνολο των Καταστημάτων χωρίς φίλτρο στην επωνυμία τους.

Η λίστα αποτελεσμάτων εμφανίζει το id, την επωνυμία, τη διεύθυνση και το τηλέφωνο του κάθε καταστήματος.

Μετά την εμφάνιση των αποτελεσμάτων της αναζήτησης (ανεξαρτήτως του τρόπου ή του αριθμού των αποτελεσμάτων) εκτυπώνεται ξανά το *Μενού Εγγεγραμμένου Πελάτη* για να συνεχίσει ο χρήστης την πλοήγηση.

Αν ο χρήστης επιλέξει **(3)** τότε ζητείται να εισάγει το id του καταστήματος από το οποίο επιθυμεί να παραγγείλει. Αν ο χρήστης εισάγει id που δεν αντιστοιχεί σε κάποιο κατάστημα τότε ενημερώνεται για αυτό και ζητείται η εκ νέου εισαγωγή id. Αν ο χρήστης πατήσει enter key επιστρέφει στο *Μενού Εγγεγραμμένου Πελάτη*.

Όταν ο χρήστης εισάγει id που αντιστοιχεί σε καταχωρημένο (υπάρχον) κατάστημα, τότε τυπώνεται ο κατάλογος προϊόντων του συγκεκριμένου καταστήματος : id, όνομα (περιγραφή) προϊόντος και τιμή ανά τεμάχιο προϊόντος.

Στη συνέχεια ζητείται από το χρήστη να εισάγει το id του προϊόντος που επιθυμεί να παραγγείλει και μετά την ποσότητα. Ο χρήστης μπορεί να εισάγει προϊόντα και ποσότητες επαναληπτικά, μέχρι να πατήσει enter key στην εισαγωγή id προϊόντος. Μετά από κάθε εισαγωγή id προϊόντος πραγματοποιείται έλεγχος αν αντιστοιχεί σε κάποιο από τα ids των προϊόντων του συγκεκριμένου καταστήματος (δηλαδή αν το προϊόν αποτελεί ένα από τα προϊόντα που έχει δηλώσει το κατάστημα ότι διαθέτει). Αν αντιστοιχεί, τότε ζητείται από το πελάτη να εισάγει την ποσότητα που επιθυμεί, διαφορετικά εκτυπώνεται κατάλληλο ενημερωτικό μήνυμα και δίνεται η δυνατότητα εκ νέου εισαγωγής id προϊόντος. Επίσης και για την τιμή της ποσότητας πραγματοποιείται έλεγχος, επιτρεπτές τιμές θεωρούνται οι ακέραιοι που ανήκουν στο διάστημα [1,20]. Αν η ποσότητα που εισάγει ο χρήστης είναι εκτός των δοθέντων ορίων, τυπώνεται κατάλληλο ενημερωτικό μήνυμα και δίδεται εκ νέου δυνατότητα εισαγωγής ποσότητας.

Όταν ο χρήστης πατήσει το enter key (κατά την εισαγωγή id προϊόντος), σταματά η εισαγωγή προϊόντων και τιμών και τυπώνονται οι επιλογές του αναλυτικά (κατάστημα αγοράς, προϊόντα και ποσότητες που επιλέχθηκαν καθώς και η συνολική τιμή) αν και ακόμα δεν έχει καταχωρηθεί η παραγγελία. Σε αυτό το σημείο, ο χρήστης ερωτάται αν θέλει να την ολοκληρώσει. Αν πατήσει «N» (από το ναι) τότε αυτή αποθηκεύεται , και τυπώνεται στην οθόνη στην *τελική της μορφή* (δείτε περιγραφή κλάσης Order) και πλέον είναι διαθέσιμη προς προβολή ανά πάσα στιγμή μέσω του *Μενού Εγγεγραμμένου Πελάτη*).

Σε κάθε περίπτωση, είτε ολοκληρώνοντας επιτυχώς την παραγγελία είτε διακόπτοντας τη διαδικασία σε όποιο βήμα, ο χρήστης επιστρέφει στο *Μενού Εγγεγραμμένου Πελάτη*.

Μενού Επισκέπτη Πελάτη

Στην περίπτωση που ο χρήστης επιλέξει να προηγηθεί ως Επισκέπτης Πελάτης δεν υπάρχει ιστορικό στο οποίο μπορούμε να διατρέξουμε, για αυτό και έχει λιγότερες επιλογές στο μενού του.

1. Εμφάνιση διαθέσιμων Καταστημάτων
2. Αναζήτηση Καταλόγου Προϊόντων Συγκεκριμένου Καταστήματος

Η διαδικασία εμφάνισης διαθέσιμων Καταστημάτων **(1)** είναι όμοια με αυτή του Εγγεγραμμένου Πελάτη.

Η διαδικασία Αναζήτησης και επιλογής προϊόντων **(2)** είναι ίδια με αυτή του Εγγεγραμμένου Πελάτη με τη διαφορά ότι αφού συμφωνήσει για την ολοκλήρωση της Παραγγελίας θα πρέπει να εισάγει όνομα, δ/ση, τηλέφωνο ώστε να είναι εφικτή η παράδοση. Μόνο μετά την επιτυχή εισαγωγή αυτών των στοιχείων η παραγγελία καταχωρείται (αποθηκεύεται).

Σε κάθε περίπτωση, είτε ολοκληρώνοντας την παραγγελία είτε διακόπτοντας τη διαδικασία σε όποιο βήμα, ο χρήστης επιστρέφει στο *Μενού Επισκέπτη Πελάτη*.

Περιγραφή Κλάσεων

Το πρόγραμμα θα αποτελείται κατ' ελάχιστον από τις παρακάτω κλάσεις:

-] Customer (αφηρημένη κλάση)
-] RegisteredCustomer
-] VisitorCustomer
-] Product
-] Shop
-] Order
-] Main (περιλαμβάνει τη main μέθοδο)

Customer

Η κλάση Customer είναι μια αφηρημένη κλάση που περιγράφει τον πελάτη της εφαρμογής.

-] Έχει μεταβλητές στιγμιοτύπου ονοματεπώνυμο, τηλέφωνο και δ/ση.
-] Κατασκευαστή που αρχικοποιεί βάσει ορισμάτων τις μεταβλητές στιγμιοτύπου και αποθηκεύει τα αντικείμενα που δημιουργούνται σε *λίστα Customer*.
-] Τέλος περιέχει την αφηρημένη μέθοδο getMenu. Η getMenu είναι υπεύθυνη για την εκτύπωση του μενού κάθε πελάτη και τη ροή των ενεργειών του.

RegisteredCustomer

Η κλάση RegisteredCustomer περιγράφει τους Εγγεγραμμένους Πελάτες της εφαρμογής, εκείνους δηλαδή που έχουν κάνει επιτυχώς εγγραφή και μπορούν να συνδεθούν σε αυτό μέσω email και password.

-)] Η RegisteredCustomer επεκτείνει τη κλάση Customer και τα αντικείμενά της περιγράφονται επιπλέον από id (αυτό- αυξανόμενη μεταβλητή), email και password.
-)] Επίσης είναι υπεύθυνη για την εκτύπωση του μενού του εγγεγραμμένου πελάτη και της ροής που προκύπτει από αυτό.
-)] Στατική μέθοδο login που λαμβάνει ως όρισμα τα credentials του πελάτη (email και password), κι αναζητά στη λίστα με τους πελάτες αν υπάρχει τέτοιο αντικείμενο. Αν υπάρχει το επιστρέφει, αλλιώς επιστρέφει null.

VisitorCustomer

Η κλάση Customer περιγράφει τους Επισκέπτες Πελάτες της εφαρμογής, εκείνους δηλαδή που επιλέγουν να μην συνδεθούν στο σύστημα με email και password.

-)] Η VisitorCustomer επεκτείνει την Customer.
-)] Επίσης είναι υπεύθυνη για την εκτύπωση του μενού του επισκέπτη πελάτη και της ροής που προκύπτει από αυτό.

Product

Η κλάση Product μοντελοποιεί τα προϊόντα που διατίθενται μέσω της εφαρμογής.

-)] Τα προϊόντα περιγράφονται από id (αυτό-αυξανόμενη μεταβλητή), όνομα (περιγραφή προϊόντος π.χ. «waffle with ice cream») και τιμή ανά τεμάχιο.
-)] Ορίζεται κατασκευαστής που αρχικοποιεί τις μεταβλητές στιγμιότυπου
-)] Ορίζεται μέθοδο toString που επιστρέφει μορφοποιημένο το αντικείμενο προϊόντος

Shop

Η κλάση Shop μοντελοποιεί τα Καταστήματα που συνεργάζονται με την εφαρμογή μας. Κάθε κατάστημα εκτός των χαρακτηριστικών του όπως δ/ση και επωνυμία ορίζει και τα δικά του Προϊόντα προς διάθεση (δείτε κλάση Product). Πιο συγκεκριμένα:

-)] Κάθε Κατάστημα περιγράφονται από id (αυτό-αυξανόμενη μεταβλητή), επωνυμία καταστήματος, διεύθυνση και μια ArrayList από Products. Η ArrayList από προϊόντα είναι απαραίτητη για να ορίσουμε ποια προϊόντα (αντικείμενα προϊόντων) διαθέτει κάθε κατάστημα, καθώς δεν πουλούν όλα τα καταστήματα τα ίδια προϊόντα.
-)] Ορίζεται κατασκευαστής που αρχικοποιεί τις μεταβλητές στιγμιότυπου
-)] Μέθοδος productIdExists (μη στατική) που λαμβάνει ως όρισμα μια ακέραια μεταβλητή (id) κι ελέγχει αν αντιστοιχεί σε κάποιο από τα Προϊόντα (Products) της ArrayList του καταστήματος.
-)] Μέθοδος printProductName (μη στατική) που λαμβάνει ως όρισμα μια ακέραια μεταβλητή (id) κι επιστρέφει το όνομα του προϊόντος με αυτό το id.

-) Μέθοδος `printProductPrice` (μη στατική) που λαμβάνει ως όρισμα μια ακέραια μεταβλητή (`id`) κι επιστρέφει την τιμή του προϊόντος με αυτό το `id`.
-) Μέθοδο `printCatalogue` (μη στατική) που εκτυπώνει τον κατάλογο προϊόντων του καταστήματος : `id`, όνομα (περιγραφή) προϊόντος και τιμή ανά τεμάχιο (προτείνεται να καλέσετε τις `printProductName` και `printProductPrice`)
-) Επίσης περιλαμβάνει τις παρακάτω στατικές μεθόδους που σχετίζονται με την αναζήτηση
 - ο **`searchById`** με όρισμα ακέραια μεταβλητή: διατρέχει τη λίστα των Καταστημάτων για να βρει αν υπάρχει κατάστημα με το δοθέν `id`. Αν υπάρχει, επιστρέφει το αντικείμενο που βρέθηκε, σε διαφορετική περίπτωση επιστρέφει `null`
 - ο **`printAllShops`** εκτυπώνει το σύνολο των καταστημάτων της εφαρμογής (συγκεκριμένα για κάθε κατάστημα: `id`, επωνυμία, διεύθυνση και τηλέφωνο)
 - ο **`filterAndPrintShops`** με όρισμα `String` μεταβλητή: εκτυπώνει το υποσύνολο των καταστημάτων της εφαρμογής που η επωνυμία τους περιέχει το δοθέν `String`. (συγκεκριμένα για κάθε κατάστημα: `id`, επωνυμία, διεύθυνση και τηλέφωνο)

Order

Η κλάση `Order` μοντελοποιεί την παραγγελία που κάνει ένας πελάτης (Εγγεγραμμένος ή Επισκέπτης) από ένα Κατάστημα αφού έχει κάνει επιλογή συγκεκριμένων Προϊόντων.

-) Η Παραγγελία περιγράφεται από τον Πελάτη που την πραγματοποίησε (αντικείμενο Πελάτη), το Κατάστημα που αφορά (αντικείμενο Καταστήματος) και, `ArrayList` των `product_id` – ποσοτήτων, το συνολικό κόστος της παραγγελίας και την ημερομηνία & ώρα παραγγελίας – `String` μεταβλητή, δείτε Μέρος 4 «Χρήσιμες Μέθοδοι»-πραγματοποίησής της.
-) Η `ArrayList` «`product_id` – ποσοτήτων» είναι μία `ArrayList` δισδιάστατων πινάκων ακεραίων τιμών.
π.χ. : `ArrayList<Integer[][]> idsQuantities = new ArrayList<Integer[][]>`
Όπου κάθε `item` της λίστας είναι ένας δισδιάστος πίνακας ακεραίων. Για παράδειγμα οι εντολές:

```

    Int [][] tempArray = {{1,4}};
    idsQuantities.add (tempArray);
  
```

 υποδηλώνουν ότι στον πίνακα με το σώμα της παραγγελίας προστέθηκε ο κωδικός προϊόντος 1 σε ποσότητα 4.
 Στη συνέχεια θα μπορούσε να επιλεγθεί ο κωδικός προϊόντος 3 σε ποσότητα 1 :

```

    tempArray = {{3,1}};
    idsQuantities.add (tempArray);
  
```
-) Εμπεριέχει μέθοδο `toString` που επιστρέφει μορφοποιημένη την παραγγελία στην τελική μορφή της (συμπεριλαμβάνοντας το `datetime` που πραγματοποιήθηκε, το όνομα του πελάτη – αν υπάρχει-, το όνομα του καταστήματος και τα ονόματα των προϊόντων με τις ποσότητες που αγοράστηκαν καθώς και το συνολικό κόστος)
-) Εμπεριέχεται στατική μέθοδος `printOrder` που αναζητά στο ιστορικό (λίστα Παραγγελιών) κι εκτυπώνει τις παραγγελιών κάθε Εγγεγραμμένου Πελάτη (ημερομηνία& ώρα παραγγελίας – `String` μεταβλητή, δείτε Μέρος 4 «Χρήσιμες Μέθοδοι»-, επωνυμία καταστήματος, συνολικό κόστος).

Main

Η Main κλάση είναι υπεύθυνη για την εκτέλεση και τη ροή της εφαρμογής. Για να μπορεί να τρέξει η εφαρμογή παράγοντας αποτελέσματα ζητείται να ορισθεί και να κληθεί στατική μέθοδος `loadObjects()` που σκοπό έχει τη δημιουργία αντικειμένων όλων των κλάσεων ώστε να φαίνεται η λειτουργικότητα.

Στη συνέχεια εκτυπώνεται το Μενού Σύνδεσης κι ανάλογα με τις επιλογές του χρήστη συνεχίζεται η πλοήγηση του ως Εγγεγραμμένος ή Επισκέπτης Πελάτης, με την εκτύπωση των αντίστοιχων μενού

3. Σημειώσεις

-) Δημιουργείστε αρκετά αντικείμενα στη `loadObjects()` ώστε να μπορούν όλα τα ερωτήματα να τρέξουν
-) Μην επιτρέψετε η εφαρμογή να διακόψει τη ροή της λόγω παραγωγής εξαιρέσεων
-) Οι διάλογοι και τα μηνύματα επικοινωνίας με το χρήστη μεριμνήστε να είναι όσο πιο κατανοητά γίνεται
-) Επίσης δώστε έμφαση στην «περιποιημένη» εμφάνιση των αποτελεσμάτων και διαλόγων στο cmd
-) Μπορείτε να δημιουργήσετε όσες επιπλέον μεθόδους κρίνετε ότι χρειάζεται
-) Αντί για λίστες (`ArrayLists`) μπορείτε να χρησιμοποιήσετε πίνακες. Το μέγεθος των οποίων είναι στη δική σας διάθεση να ορίσετε.

4. Χρήσιμες Μέθοδοι

-) Για να βρούμε αν ένα `String` περιέχει ένα `substring` μπορούμε να χρησιμοποιήσουμε τη `Boolean` μέθοδο `contains`
[https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#contains\(java.lang.CharSequence\)](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#contains(java.lang.CharSequence))
-) Μπορούμε να μετρέψουμε σε κεφαλαία ή μικρά όλα τα γράμματα μιας συμβολοσειράς (`String`) με τη χρήση των μεθόδων `toUpperCase()` και `toLowerCase`
[https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#toUpperCase\(\)](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#toUpperCase())
[https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#toLowerCase\(\)](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#toLowerCase())
-) Μετατροπή `Integer` σε primitive `int` type:
[https://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html#parseInt\(java.lang.String\)](https://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html#parseInt(java.lang.String))

```
Integer myNumber = 8;  
int number = Integer.parseInt(myNumber.toString());
```
-) Αποθήκευση σε `String` μεταβλητή και στη συνέχεια εκτύπωση του τρέχοντος `datetime`

```
1 package ergasia;
2
3 import java.text.DateFormat;
4 import java.text.SimpleDateFormat;
5 import java.util.Date;
6
7 public class Test {
8
9     public static void main(String[] args) {
10         DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
11         Date date = new Date();
12         String datetime = dateFormat.format(date);
13         System.out.println(datetime);
14     }
15 }
16
17 }
```

Καλή επιτυχία!