

2^η Εργασία στον Αντικειμενοστραφή Προγραμματισμό

Περιγραφή

Σκοπός αυτής της εργασίας είναι η δημιουργία εφαρμογής σε Java για τις παραγγελίες ενός «Coffee Shop». Το συγκεκριμένο μαγαζί είναι ανοιχτό από τις 12:00 το μεσημέρι έως τις 24:00 το βράδυ.

Παραγγελίες

Κάθε παραγγελία που παραδίδεται τηλεφωνικά, έχει τα ακόλουθα χαρακτηριστικά:

- Μοναδικό αύξοντα αριθμό
- Ώρα παραγγελίας

Μια παραγγελία μπορεί να αναπαρασταθεί στο πρόγραμμά σας ως εξής:

< num: t_order,t_fullfil, items>

Όπου

- **num:** ο αύξων αριθμός της παραγγελίας,
- **t_order:** αντιστοιχεί στη χρονική στιγμή λήψης της παραγγελίας, εκπεφρασμένη σε αριθμό λεπτών από τις 12:00, οπότε ανοίγει το Coffee Shop, μέχρι τις 24:00 (π.χ. μια παραγγελία που έγινε στις 13:45 θα έχει χρονική στιγμή λήψης t_order = 105).
- **t_fullfil:** ο χρονική στιγμή σε λεπτά (με την παραπάνω θεώρηση), την οποία θα είναι έτοιμη η παραγγελία.
- **items:** Τα προϊόντα που παραγγέλνει ο χρήστης.

Βασικά στοιχεία εφαρμογής

- **Μενού Coffee Shop:** Αρχικά στην πρώτη εκτέλεση του προγράμματος πρέπει να δημιουργήσετε το Μενού του μαγαζιού που θα περιέχει κάθε προϊόν (θα πρέπει να περιέχει **10 προϊόντα, 5 φαγητά και 5 ποτά**), τον τύπο του (π.χ. φαγητό ή ποτό), την τιμή του σε ευρώ και τον χρόνο σε λεπτά που χρειάζεται για να ετοιμαστεί το συγκεκριμένο αντικείμενο. Εγγράψετε το μενού σε σχετικό αρχείο (**menu.txt**) έτσι ώστε σε κάθε νέα εκτέλεση του προγράμματος το μενού να διαβάζεται από αυτό το αρχείο σαν είσοδος **line command argument args[0]** (οπότε το πρόγραμμα να μπορεί να τρέξει ως **java Όνομα Προγράμματος menu.txt**). Το συγκεκριμένο αρχείο σε κάθε γραμμή θα περιέχει τον αύξοντα αριθμό του προϊόντος, το όνομα του το τύπο του (*food or drink*), την τιμή του σε ευρώ και τον χρόνο σε λεπτά που χρειάζεται για να ετοιμαστεί το συγκεκριμένο αντικείμενο) π.χ.

1, coca-cola, drink, 2.5, 3

2, burger, food, 5, 10

Κτλ.

Δηλαδή η «coca-cola» κοστίζει 2,5 ευρώ και χρειάζεται 3 λεπτά να ετοιμαστεί, ενώ το «burger» κοστίζει 5 ευρώ και χρειάζεται 10 λεπτά να ετοιμαστεί.

- **Όνομα Coffee Shop:** Το Coffee Shop θα πρέπει να έχει ένα συγκεκριμένο όνομα.

Αφού διαβάσετε και τα **3 Επίπεδα**, υλοποιήστε μόνο 1 από τα 3 παρακάτω επίπεδα της εργασίας το οποίο έχει και μία ξεχωριστή ανώτατη βαθμολογία. Κάθε ανώτερο επίπεδο θα πρέπει να περιέχει όλες τις λειτουργίες των προηγούμενων επιπέδων.

Επίπεδο 1 (Ανώτατη βαθμολογία: 6):

1. Η εφαρμογή θα πρέπει να συνεχίζει να δέχεται παραγγελίες από πελάτες όσο ο χρήστης δίνει από το πληκτρολόγιο τον ακέραια τιμή 1, ενώ αν δίνει την τιμή 0 θα πρέπει να τερματίζεται το πρόγραμμα.
2. Για κάθε πελάτη, ο χρήστης θα πρέπει να πληκτρολογεί όλα τα αντικείμενα που επιλέγει και στο τέλος να πληκτρολογεί την λέξη «stop» για να δηλώσει το τέλος της παραγγελίας. (**Σημείωση:** Εισάγεται τις επιλογές του πελάτη με χρήση **Stream** χαρακτήρων από το πληκτρολόγιο)
3. Το πρόγραμμα θα πρέπει να αθροίζει το σύνολο των λεπτών που χρειάζεται για να ετοιμαστεί η παραγγελία του κάθε πελάτη καθώς και το συνολικό χρόνο για τον σύνολο των παραγγελιών. Όταν ο χρόνος που χρειάζεται για να ετοιμαστεί κάθε παραγγελία ενός πελάτη περάσει μέσω αντίστοιχης μεθόδου (π.χ. **fulfillOrder**) το πρόγραμμα θα πρέπει να υπολογίζει τον καινούργιο συνολικό χρόνο παραγγελιών. (**Σημείωση:** Κάθε πελάτης εξυπηρετείται σαν μία ουρά **FIFO (First-In-First-Out)**. Επίσης εγγράψετε τις παραγγελίες για κάθε τρέξιμο του προγράμματος σε σχετικό αρχείο (**orders.txt**) σαν **output**, έτσι ώστε να μπορεί να χρησιμοποιηθεί ως γεννήτρια παραγγελιών. Κάθε γραμμή του αρχείου θα περιλαμβάνει κάθε παραγγελία με βάση τον χρόνο της παραγγελίας π.χ.

1,101,111, item1, item2,...

2,110, 120, item1,...

Όπου το 1^ο ψηφίο το **num**, το δεύτερο το **t_order** σε λεπτά, το τρίτο το **t_fullfil** σε λεπτά και τέλος τα **items** που περιέχει κάθε παραγγελία.

Τέλος μετρήστε το χρόνο σε λεπτά, οπότε αν η ώρα είναι **13:40**, η ώρα θα είναι **100 λεπτά**.

4. Για κάθε νέο προϊόν που παραγγέλλει ο πελάτης το πρόγραμμα θα πρέπει να πληροφορεί τον χρήστη αν το συγκεκριμένο προϊόν υπάρχει στο μενού ή όχι.
5. Τέλος, ο χρήστης για κάθε πελάτη θα πρέπει να μπορεί να έχει τη δυνατότητα να ζητήσει και να δει: το συνολικό κόστος της παραγγελίας μέχρι κάθε ένα αντικείμενο που παραγγέλλει, τον χρόνο που χρειάζεται μέχρι να ετοιμαστεί η παραγγελία του, το φθηνότερο προϊόν του μενού, μόνο τα ροφήματα του μενού και μόνο τα φαγητά του μενού.

Επίπεδο 2 (Ανώτατη βαθμολογία: 7):

Δημιουργήστε μία παραλλαγή του παραπάνω προγράμματος του **Επιπέδου 1**, στο οποίο για κάθε πελάτη, ο χρήστης δεν θα μπορεί μόνο να πληκτρολογεί όλα τα αντικείμενα που επιλέγει ο πελάτης αλλά θα χρησιμοποιεί και το αρχείο **orders.txt** σαν γεννήτρια παραγγελιών. Δηλαδή το αρχείο αυτό να μπορεί να χρησιμοποιηθεί σαν 2η **line command argument args[1]** (οπότε το πρόγραμμα να μπορεί να τρέξει και ως **java Όνομα Προγράμματος menu.txt orders.txt**).

Επίπεδο 3:

Μετατρέψτε το παραπάνω πρόγραμμα των προηγούμενων επιπέδων σε παραθυρική εφαρμογή μέσω των βασικών βιβλιοθηκών **swing** ή **awt** της Java. Το γραφικό και οι λειτουργίες της εφαρμογής θα πρέπει να σχεδιαστούν ως εξής:

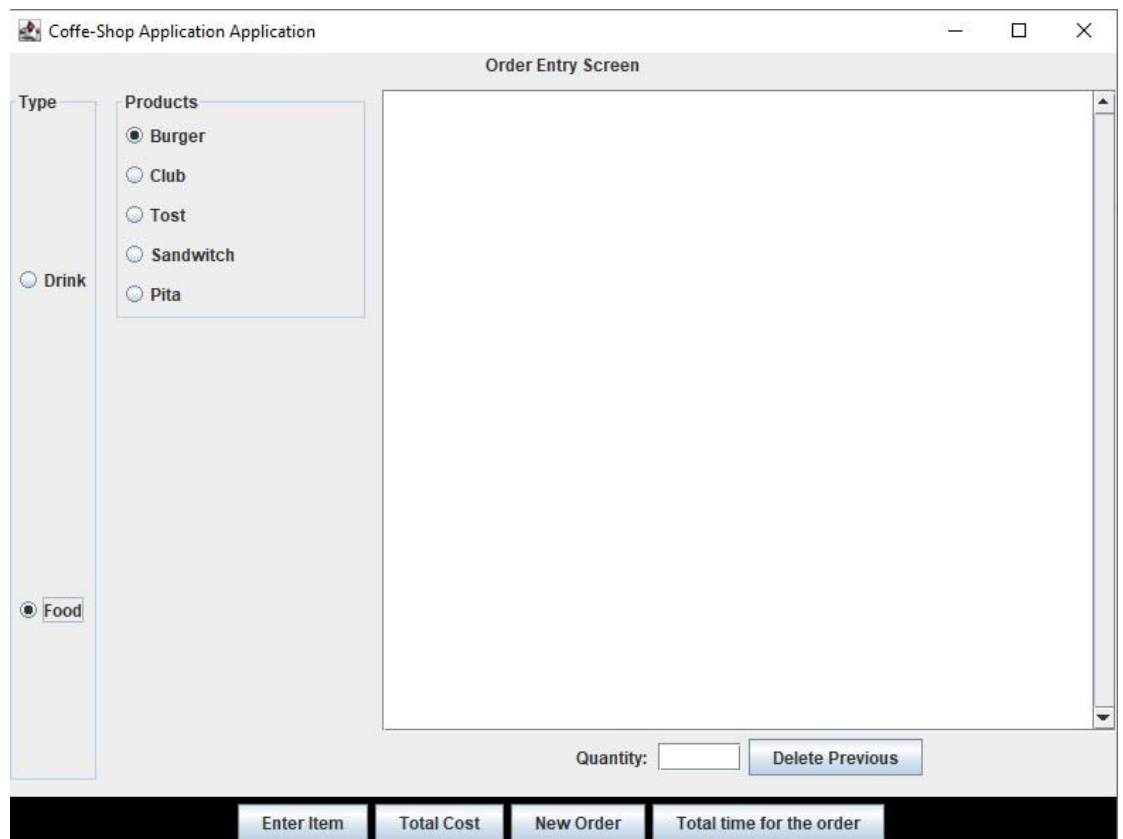
- Το αρχικό παράθυρο της εφαρμογής θα πρέπει να έχει ως τίτλο του Coffee-Shop.
- Το κεντρικό Panel θα πρέπει να είναι στο κέντρο του παραθύρου και να περιέχει τον το label «Order Entry Screen»
- Στο αριστερό μέρος του παραθύρου θα πρέπει να υπάρχει Panel που καθορίζει τον τύπο του προϊόντος της παραγγελία (π.χ. Coffee or Drink). (**Σημείωση:** Το μενού του μαγαζιού θα διαβάζεται από το αρχείο **menu.txt** το οποίο θα δημιουργηθεί με τη 1^η εκτέλεση του προγράμματος).

- Ανάλογα με την επιλογή του χρήστη (π.χ. Coffee or Drink), θα πρέπει να εμφανίζονται τα αντίστοιχα προϊόντα που ανήκουν στη κάθε κατηγορία με αντίστοιχο **Gridlayout**.
- Στα νότια του κεντρικού παραθύρου θα πρέπει να υπάρχει Panel με τα Buttons (**“Enter Item”, “Total Cost”, “New Order”, “Total time for the order”**)
- Στο κέντρο του κεντρικού παραθύρου θα πρέπει να υπάρχει Label **“Quantity”** με **JTextField** στο οποίο ο χρήστης θα συμπληρώνει την ποσότητα του προϊόντος (ακέραιος) που επιλέγει ο πελάτης και επίσης θα πρέπει να υπάρχει Button **“Delete Previous”** με το οποίο ο χρήστης θα μπορεί να διαγράψει το προηγούμενο προϊόν που επέλεξε ο πελάτης. Επίσης σε αυτό το Panel θα πρέπει να υπάρχει **JTextArea** η οποία θα είναι η οθόνη του γραφικού και θα τυπώνει τις αντίστοιχες πληροφορίες ανάλογα με τις επιλογές του χρήστη. Για παράδειγμα, για κάθε φορά που χρήστης επιλέγει την επιλογή **“Enter Item”** θα πρέπει να εκτυπώνεται το όνομα του προϊόντος η τιμή του και ο χρόνος που χρειάζεται για να ετοιμαστεί, αν επιλέξει το **“Total Cost”** θα εκτυπωθεί επίσης το συνολικό κόστος της παραγγελίας, αν επιλέξει **“New Order”** η **JTextArea** θα πρέπει να εκκενωθεί και τέλος με την επιλογή **“Total time for the order”** θα πρέπει να εκτυπωθεί επίσης ο συνολικός χρόνος που χρειάζεται για την παραγγελία. Με την επιλογή **“Delete”** ο χρήστης θα αφαιρεί από το **JTextArea** την τελευταία επιλογή της συγκεκριμένης παραγγελίας κάθε φορά (**Σημείωση: ο χρόνος για κάθε παραγγελία θα πρέπει να λαμβάνει και τον χρόνο για προηγούμενες παραγγελίες εφόσον δεν έχουν ολοκληρωθεί**).
- Με την επιλογή **“New order”** θα πρέπει επίσης να ελέγξετε αν κάποια από τις προηγούμενες παραγγελίες έχει ολοκληρωθεί και αν ναι να καλέστε μέθοδο **fullfilOrder()** όπως και στα προηγούμενα επίπεδα για να υπολογίσετε

τους καινούργιους χρόνους και να εγγράψετε την αντίστοιχη παραγγελία σε σχετικό αρχείο (**orders.txt**) σαν output του προγράμματος, όπως προαναφέρθηκε και σε παραπάνω επίπεδο.

- Μπορείτε να προσθέσετε επιπλέον Buttons στο παραπάνω γραφικό με τα οποίο χρήστης για κάθε πελάτη να μπορεί να επιλέξει αυτόματα το φθηνότερο ποτό ή το φθηνότερο φαγητό του μενού.
- Κάθε **Gui component** που δημιουργήσατε θα πρέπει να συνδέεται με το αντίστοιχο **event object** μέσω της διεπαφής **ActionListener**.

Το γραφικό της εφαρμογής θα πρέπει να μοιάζει με το παρακάτω:



Υποδείξεις:

Χρησιμοποιήστε Stream χαρακτήρων, ArrayList, Πίνακες, καθώς και κλάσεις και μεθόδους του `java.time.*` πακέτου για τον υπολογισμό της ώρας. Μπορείτε να χρησιμοποιήσετε τα πακέτα `java.awt.*`, `javax.swing.*`

για τον σχεδιασμό της παραθυρικής εφαρμογής και ότι άλλο νομίζετε πως χρειάζεστε. Κάντε αποδοτική χρήση κλάσεων και αντικειμένων. Υποστηρίζοντας την αντικειμενοστραφή φύση της Java υλοποιήστε τις παραπάνω λειτουργίες χρησιμοποιώντας όσο το δυνατόν περισσότερα αντικείμενα.

ΑΞΙΟΛΟΓΗΣΗ

Για την αξιολόγηση του προγράμματος σας, θα ληφθούν οι πιο κάτω παράμετροι:

- **Ορθότητα:** ο πρόγραμμά σας πρέπει τρέχει ορθά για οποιαδήποτε είσοδο
- **Κατανοητά σχόλια:** Γράφετε κατανοητά σχόλια που να εξηγούν την λειτουργία της κάθε κλάσης/πεδίου/μεθόδου.
- **Σωστή χρήση προσδιοριστών πρόσβασης**
- **Σωστή χρήση κατασκευαστών**

ΟΔΗΓΙΕΣ

- Στο πάνω μέρος της κάθε κλάσης να γράψετε με σχόλια το ονοματεπώνυμο και τον αριθμό ταυτότητας σας.
- Βεβαιωθείτε ότι τα προγράμματα σας είναι ορθά και τρέχουν.
- Οι άσκηση σας θα πρέπει να παραδοθεί σαν ένα zip αρχείο (που να περιλαμβάνει όλες τις κλάσεις που θα υλοποιήσετε και το έγγραφο τεκμηρίωσης) σε ηλεκτρονική μορφή στο eclass στην ημερομηνία παράδοσης.
- Το zip αρχείο θα είναι πρέπει να γίνει μέσω export από το eclipse σαν archive file.
- Το zip αρχείο θα ονομάζεται με τον εξής τρόπο:
erg2.onoma.epwnymio.omada_x_eripedo_y_αριθμός_μητρώου.zip, όπου x η ομάδα εργαστηρίου στην οποία ανήκετε και y το επίπεδο το οποίο υλοποιείτε.