



Αντικειμενοστραφής Προγραμματισμός

Εργασία Ακαδημαϊκού Έτους 2024-2025

1 Διαδικαστικά

Η εργασία θα πρέπει να παραδοθεί μέχρι τις **19/01/2025 23:59** και μετράει κατά **30%** στον τελικό βαθμό του μαθήματος (3 μονάδες από τις 10). Υπάρχει και ένα *bonus section* το οποίο μπορεί να σας προσφέρει έξτρα 0.5 μονάδες στον τελικό βαθμό.

Η εργασία θα πρέπει να γίνει από **ομάδες των δύο ατόμων το πολύ (μπορεί να είναι κάποιος μόνος του)**, οι οποίες θα δουλέψουν συνεργατικά σε ένα ιδιωτικό αποθετήριο κώδικα στο GitHub.

1.1 Οδηγίες για το GitHub Repository

- Ένα από τα μέλη της ομάδας αρχικά θα πατήσει το ακόλουθο [link](#) και θα δημιουργήσει την ομάδα (από τον τομέα "create a new team"). Πιο συγκεκριμένα, **οφείλετε να ονομάσετε την ομάδα sdiXXXXXXXX-sdiYYYYYYY ή sdiXXXXXXXX** εάν δεν έχετε ομάδα. Έπειτα, πατήστε το κουμπί "accept the assignment". Έτσι, θα σας δοθεί το link του ιδιωτικού repository σας.
- Το 2ο μέλος της ομάδας πατάει το ίδιο [link](#) και διαλέγει την ομάδα που δημιούργησε ο συμφοιτητής του (από τον τομέα "join an existing team"). **Προσοχή:** διαλέξτε τη σωστή ομάδα! Μην μπειτε σε ομάδα άλλων φοιτητών. Μόλις πατήσετε το κουμπί "join", το 2ο μέλος θα αποκτήσει και αυτό πρόσβαση στο ιδιωτικό αποθετήριο της ομάδας.

1.2 Αξιολόγηση

Για την αξιολόγηση των εργασιών θα συνεκτιμηθεί:

- Η συνεισφορά του κάθε μέλους της ομάδας βάσει των commit στο GitHub.
- Η υλοποίηση του λογισμικού (πληρότητα υλοποίησης λειτουργιών, ποιότητα υλοποίησης).
- Ο έλεγχος των αποτελεσμάτων του λογισμικού, μέσω παρεχόμενων unit tests.
- Η πληρότητα τεκμηρίωσης (Ποσότητα σχολίων, τεκμηρίωση κλάσεων, μεθόδων, παραμέτρων και επιστρεφόμενων τιμών μεθόδων, κτλ).
- Οφείλετε να παραδώσετε ένα αρχείο **README** που να αναλύετε τις σχεδιαστικές σας επιλογές και τη λύση σας.

Τέλος, ενδέχεται να υπάρξει προφορική εξέταση της εργασίας.

2 Ζητούμενο

Στο πλαίσιο της εργασίας καλείστε να σχεδιάσετε και να υλοποιήσετε σε **C++** μια **εφαρμογή** που θα προσφέρει λειτουργίες σχετικές με ένα **eshop**. Μέσα από την εφαρμογή, οι διαχειριστές θα μπορούν να εισάγουν διαθέσιμα προϊόντα, π.χ. μακαρόνια, ποτά, κτλ. Οι χρήστες (πελάτες) θα μπορούν να έχουν λογαριασμό ώστε να αποκτήσουν πρόσβαση στην εφαρμογή, να αναζητήσουν και να παραγγείλουν προϊόντα. Στόχος της εργασίας είναι να εξοικειωθείτε:

1. Με τις διαδικασίες σχεδιασμού, υλοποίησης και ελέγχου αντικειμενοστρεφούς λογισμικού χρησιμοποιώντας τεχνικές απόκρυψης πληροφορίας, πολυμορφισμού και επαναχρησιμοποίησης κώδικα.
2. Με τον συνεργατικό προγραμματισμό, δουλεύοντας σε ομάδες και σε αποθετήριο git.

Η εφαρμογή που θα αναπτύξετε θα πρέπει να υποστηρίζει **δύο γενικούς τύπους χρηστών: διαχειριστές και πελάτες**. Οι διαχειριστές θα μπορούν να καταχωρούν τα προϊόντα του eshop στην πλατφόρμα που θα γίνονται διαθέσιμα για αγορά στους πελάτες. Οι πελάτες θα πρέπει να έχουν λογαριασμό ώστε να μπορούν να αναζητούν, να τοποθετούν προϊόντα στο καλάθι και να ολοκληρώνουν τις παραγγελίες τους. Θα χρειαστεί να υλοποιήσετε κάποιον μηχανισμό σύνδεσης (login/authentication) στην εφαρμογή ώστε να εμφανίζονται οι κατάλληλες επιλογές ανάλογα με τον ρόλο του χρήστη.

Το πρόγραμμα θα πρέπει να αρχικοποιείται (μέσω αρχείων) με ικανοποιητικό αριθμό χρηστών και να έχει ικανοποιητικό αριθμό καταχωρημένων προϊόντων με τις απαραίτητες πληροφορίες. Η χρήση αρχείων είναι απαραίτητη για την αποθήκευση των ρόλων και των credentials των χρηστών, των πληροφοριών των προϊόντων και γενικά όλων των δεδομένων και αλλαγών που γίνονται κατά τη διάρκεια χρήσης του προγράμματος, για να είναι διαθέσιμα και μετά τον τερματισμό του. Για παράδειγμα, αν αγοραστεί μια ποσότητα ενός προϊόντος και τερματιστεί το πρόγραμμα, την επόμενη φορά που θα τρέξει ξανά το πρόγραμμα, θα πρέπει να είναι στην ίδια κατάσταση ακριβώς πριν τον τελευταίο τερματισμό, δηλαδή να είναι μειωμένη η ποσότητα του αγορασμένου προϊόντος (αποθέματος).

Το λογισμικό σας θα πρέπει να είναι σχεδιασμένο βάσει των αρχών του αντικειμενοστρεφούς προγραμματισμού που παρουσιάζονται στο μάθημα, ώστε να είναι εύκολη η συντήρηση και η επέκτασή του. Για παράδειγμα στο μέλλον μπορεί να θέλουμε να προστεθούν και άλλες κατηγορίες προϊόντων, χρηστών, κτλ.

3 Βασικές Λειτουργίες

3.1 Διαχειριστές

Οι διαχειριστές θα πρέπει να μπορούν:

1. Να καταχωρούν νέα προϊόντα στην εφαρμογή.
2. Να επεξεργάζονται προϊόντα που έχουν καταχωρήσει οι ίδιοι ή άλλοι διαχειριστές.
3. Να αναζητούν καταχωρημένα προϊόντα.
4. Να βλέπουν κάποια χρήσιμα στατιστικά για τα προϊόντα, όπως:
 - a. Τα μη διαθέσιμα προϊόντα (δηλαδή δεν υπάρχει διαθέσιμη ποσότητα (απόθεμα) για το προϊόν).

- b. Προϊόντα που έχουν εμφανιστεί στις περισσότερες παραγγελίες (ανεξαρτήτως ποσότητας).

3.2 Πελάτες

Οι πελάτες θα πρέπει να μπορούν:

1. Να αναζητούν καταχωρημένα προϊόντα.
2. Να προσθέτουν προϊόντα στο καλάθι τους.
3. Να μπορούν να ενημερώσουν το καλάθι τους, αλλαγή ποσότητας προϊόντος ή διαγραφή προϊόντος.
4. Να ολοκληρώσουν παραγγελίες.
5. Να έχουν πρόσβαση στο ιστορικό παραγγελιών τους.

Θα πρέπει να υποστηρίζονται λογαριασμοί για να μπορεί η εφαρμογή να γνωρίζει ποιος τύπος χρήστη χρησιμοποιεί το πρόγραμμα, ώστε να εμφανίζει κατάλληλες επιλογές. Για παράδειγμα, δεν πρέπει να εμφανίζεται επιλογή σε πελάτες για προσθήκη ενός νέου προϊόντος στο σύστημα.

4 Αναλυτικές Προδιαγραφές Λογισμικού

4.1 Καταχώρηση Προϊόντος

Μόνο οι διαχειριστές θα πρέπει να μπορούν να καταχωρούν προϊόντα στην εφαρμογή. Οι πληροφορίες που μας ενδιαφέρουν για κάθε προϊόν είναι:

1. **Τίτλος.**
2. **Περιγραφή:** ελεύθερο κείμενο με πληροφορίες για το προϊόν.
3. **Κατηγορία προϊόντος:** προκαθορισμένες λίστες με κατηγορίες και αντίστοιχες υποκατηγορίες. Επιλέγοντας μια κατηγορία θα πρέπει να εμφανίζονται μόνο οι σχετικές υποκατηγορίες τις οποίες θα μπορεί να επιλέξει ο διαχειριστής. Για κάθε προϊόν θα πρέπει να επιλέγεται οπωσδήποτε μια υποκατηγορία, όχι μόνο κατηγορία.
4. **Τιμή.**
5. **Διαθέσιμα τεμάχια/κιλά,** ανάλογα με το προϊόν.

Σημειώσεις:

- Όλα τα χαρακτηριστικά είναι **υποχρεωτικά**.
- Ανάλογα με τον τύπο του προϊόντος, θα πρέπει στην φόρμα να είναι εμφανές αν ζητείται να εισάγουμε τεμάχια ή κιλά.

4.2 Προβολή Προϊόντος

Για κάθε προϊόν θα πρέπει να υπάρχει η δυνατότητα προβολής των χαρακτηριστικών του, και να υπάρχουν επιλογές ανάλογα με το ποιος τύπος χρήστη «βλέπει» την καταχώρηση:

1. Για κάθε προϊόν θα πρέπει να εμφανίζεται μια καρτέλα με όλες οι πληροφορίες του.
2. Αν ο χρήστης είναι διαχειριστής, θα πρέπει να υπάρχει επιλογή επεξεργασίας του προϊόντος ώστε να αλλάζουν τα χαρακτηριστικά του προϊόντος, όπως διαθέσιμη ποσότητα στην αποθήκη, τίτλος, κατηγορία, κτλ.
3. Αν ο χρήστης είναι πελάτης, θα πρέπει να υπάρχει επιλογή για προσδιορισμό της ποσότητας που επιθυμεί να αγοράσει και της προσθήκης του προϊόντος στο καλάθι. Ο πελάτης θα πρέπει να ενημερώνεται σε περίπτωση που ζητήσει ποσότητα που δεν είναι διαθέσιμη. Σε αυτή την περίπτωση, θα πρέπει να καταχωρείται στο καλάθι του όλο το απόθεμα που υπάρχει.

4.3 Αναζήτηση Προϊόντος

Το πρόγραμμα θα πρέπει να προσφέρει τη δυνατότητα αναζήτησης προϊόντων, και στους διαχειριστές και στους πελάτες, με τα εξής κριτήρια:

1. **Τίτλος:** απλό κείμενο που θα ταιριάζει στους τίτλους των προϊόντων (ενδεχομένως πολλές λέξεις).
2. **Κατηγορία – υποκατηγορία:** μέσω προκαθορισμένων λιστών. Η αναζήτηση θα μπορεί να γίνει επιλέγοντας και μόνο κατηγορία (χωρίς να επιλέξει υποκατηγορία), οπότε σε αυτήν την περίπτωση θα εξετάζονται όλα τα προϊόντα που ανήκουν στις αντίστοιχες υποκατηγορίες.

Ο χρήστης θα πρέπει να μπορεί να επιλέξει ένα αποτέλεσμα της αναζήτησης και να εμφανίζεται η καρτέλα της προβολής του προϊόντος.

Σημειώσεις:

- Ως κριτήρια αναζήτησης θα μπορούν να χρησιμοποιηθούν είτε το ένα από τα δύο είτε και τα δύο ταυτόχρονα δηλαδή και τίτλος και κατηγορία.
- Αν δεν προσδιοριστεί κάποιο κριτήριο στην αναζήτηση, θα πρέπει να επιστρέφονται όλα τα προϊόντα με όποια σειρά κρίνετε εσείς κατάλληλη.

4.4 Καλάθι Αγορών

Ο πελάτης θα πρέπει να μπορεί να προσθέτει προϊόντα στο καλάθι του με την αντίστοιχη ποσότητα που τον ενδιαφέρει. Θα πρέπει να υπάρχει διαθέσιμη επιλογή εκτύπωσης καρτέλας με πληροφορίες για το ενεργό καλάθι του χρήστη, με το εξής **format**:

```
---CART START---  
Ποσότητα1 Τίτλος1  
Ποσότητα2 Τίτλος2  
...  
ΠοσότηταN ΤίτλοςN  
---CART END---
```

Total Cost: X

Ενδεικτικό παράδειγμα:

---CART START---

10 Apple

2 Can of Orange Juice

6 Bottle of Beer

---CART END---

Total Cost: 40

Θα πρέπει επίσης να υπάρχουν οι εξής επιλογές/δράσεις:

1. Επιλογή εισαγωγής νέου προϊόντος στο καλάθι.
2. Επιλογή να διαγραφεί κάποιο προϊόν από το καλάθι.
3. Δυνατότητα αλλαγής της ποσότητας (τεμάχια/κιλά) αγοράς ενός προϊόντος (με σχετική ενημέρωση αν η νέα ποσότητα δεν είναι διαθέσιμη).

Στην καρτέλα του καλαθιού **θα πρέπει να υπάρχει επιλογή για την ολοκλήρωση της παραγγελίας**. Σε αυτήν την περίπτωση, θα πρέπει να ενημερώνεται η ποσότητα (τεμάχια ή κιλά) των διαθέσιμων προϊόντων. Επίσης μετά την ολοκλήρωση της παραγγελίας, θα πρέπει ο πελάτης να μπορεί να δει στο ιστορικό παραγγελιών του την παραγγελία που μόλις έκανε.

4.5 Ιστορικό Παραγγελιών

Οι πελάτες θα πρέπει να έχουν την δυνατότητα να δουν τις παραγγελίες που έχουν κάνει. Για κάθε μία παραγγελία θα πρέπει να φαίνεται:

1. Τα προϊόντα και οι ποσότητες (τεμάχια ή κιλά) που αγοράστηκαν.
2. Συνολικό κόστος.

Πιο συγκεκριμένα, **πρέπει υποχρεωτικά να ακολουθήσετε το ακόλουθο format αρχείου ιστορικού παραγγελιών**. Αν δεν ακολουθήσετε αυστηρά αυτό το format, δεν θα εκτελούνται επιτυχώς τα unit tests.

Το αναφερόμενο format είναι το εξής:

---CART 1 START---

Ποσότητα1 Τίτλος1

Ποσότητα2 Τίτλος2

...

ΠοσότηταN ΤίτλοςN

---CART 1 END---

Total Cost: X

---CART 2 START---

Ποσότητα3 Τίτλος3

Ποσότητα4 Τίτλος4

...

ΠοσότηταM ΤίτλοςM

---CART 2 END---

Total Cost: Y

Είναι επίσης υποχρεωτικό τα αρχεία αυτά να έχουν όνομα <USERNAME>_history.txt (π.χ. eleni_history.txt), και να βρίσκονται εντός του φακέλου files/order_history/.

Υπάρχουν παραδείγματα στον φάκελο files/order_history/<USERNAME>_history.txt. Όπως παρατηρείτε, πρόκειται για αντίστοιχο format με εκείνο εμφάνισης του ενεργού καλαθιού που προαναφέρθηκε. Μοναδική διαφορά αποτελεί ότι πλέον έχουμε πολλά καλάθια (ιστορικό κάθε ολοκληρωμένης παραγγελίας), και άρα τα αριθμούμε.

4.6 Εγγραφή πελατών

Φόρμα εγγραφής πελατών στην πλατφόρμα (registration):

1. Username
2. Password
3. isAdmin

Σημειώσεις:

- Δεν θα πρέπει να επιτρέπεται η εγγραφή κάποιου χρήστη με username το οποίο ήδη υπάρχει στο σύστημα.
- Όλα τα πεδία είναι υποχρεωτικά.

4.7 Σύνδεση/Αποσύνδεση Χρηστών

Ανάλογα με το username και το password, θα πρέπει η εφαρμογή να καταλαβαίνει το είδος του χρήστη και να προσαρμόζει τη λειτουργικότητα της. Όλοι οι χρήστες (διαχειριστές και πελάτες) θα αποθηκεύονται σε μία **κοινή δομή δεδομένων**. Θα πρέπει να αιτιολογήσετε την επιλογή δομής (ADT) που θα χρησιμοποιήσετε.

4.8 Αρχικοποίηση και Εκτέλεση Προγράμματος

Η αρχικοποίηση θα πρέπει να γίνεται μέσα από αρχεία τα οποία είναι τοποθετημένα στο φάκελο files/ και οι αλλαγές που θα γίνονται θα πρέπει να αποθηκεύονται κατά τον τερματισμό της εφαρμογής στα αρχεία. Τα αρχεία έχουν **συγκεκριμένη** σύνταξη την οποία **πρέπει** να τηρήσετε **υποχρεωτικά**.

- Αρχικά, οι χρήστες που υποστηρίζονται από το πρόγραμμά σας θα πρέπει να υπάρχουν σε ένα αρχείο `users.txt`. Οι χρήστες που θα υπάρχουν αρχικά θα είναι:

Δύο διαχειριστές με:

- Username: admin1, password: pass1
- Username: admin2, password: pass2

Δύο πελάτες με:

- Username: user1, password: pass1
- Username: user2, password: pass2

Η αρχική μορφή αυτού του αρχείου παρέχεται ήδη (`files/users.txt`). Το τρίτο "όρισμα" κάθε γραμμής παίρνει τιμή 0/1 και αφορά το αν ο χρήστης είναι διαχειριστής (1) ή όχι (0).

- Το εκτελέσιμο που θα παράξετε πρέπει να έχει όνομα `oop24` και να εκτελείται με τον ακόλουθο τρόπο:
`./oop24 files/categories.txt files/products.txt files/users.txt`
- Η λίστα με τις κατηγορίες και τις υποκατηγορίες που πρέπει να υποστηρίζει το πρόγραμμά σας θα υπάρχει σε ένα αρχείο `files/categories.txt`. Κάθε γραμμή είναι της μορφής: Κατηγορία (Υποκατηγορία1 @ Υποκατηγορία2...)

Είστε ελεύθεροι να προσθέτε δικές σας κατηγορίες και υποκατηγορίες. Ωστόσο, **απαγορεύεται να διαγράψετε ή να αλλάξετε τις υπάρχουσες κατηγορίες.**

- Τα προϊόντα που πρέπει να υποστηρίζονται από το πρόγραμμά σας θα πρέπει να υπάρχουν σε ένα αρχείο `files/products.txt`. Κάθε γραμμή είναι της μορφής:
 Τίτλος @ Περιγραφή @ Κατηγορία @ Υποκατηγορία @ Τιμή @ Μονάδα μέτρησης @ Ποσότητα
 Η μονάδα μέτρησης είναι Kg για τα προϊόντα τα οποία μετρώνται σε βάρος και Unit για τα προϊόντα τα οποία μετρώνται ανά τεμάχιο.
 Είστε ελεύθεροι να προσθέτε τα δικά σας προϊόντα, ωστόσο **απαγορεύεται να διαγράψετε ή να αλλάξετε τα υπάρχοντα προϊόντα.**

5 Unit Tests

Παράλληλα με τον φάκελο `files/`, σας δίνονται το εκτελέσιμο πρόγραμμα `unit_tests` και ο φάκελος `tests/`. Τα αναφερόμενα αποτελούν απαραίτητες οντότητες για τα αυτοματοποιημένα unit tests. Τα αρχεία που βρίσκονται στον φάκελο `tests/` προσομοιάζουν user inputs και χρησιμοποιούνται από το εκτελέσιμο (`unit_tests`) για την αυτόματη αξιολόγηση των αποτελεσμάτων του προγράμματός σας.

Δεν επιτρέπεται η αλλαγή των περιεχομένων των `.txt` αρχείων. Τονίζεται, επιπλέον, ότι τα unit tests είναι ενδεικτικά και δεν αντικατοπτρίζουν την τελική βαθμολογία. Για να τρέξετε τα unit tests, αρκεί η εντολή `./unit_tests`.

6 Sample Executions

Μερικά ενδεικτικά παραδείγματα λειτουργίας βρίσκονται στον φάκελο `samples/`. Τα inputs του χρήστη που δόθηκαν κατά την εκτέλεση του προγράμματος βρίσκονται εσωκλεισμένα σε `**sample_input**`, ώστε να είναι ευδιάκριτα στα αρχεία.

7 BONUS (+0.5 μονάδες στον τελικό βαθμό)

Καλείστε να υλοποιήσετε ένα **σύστημα εκπτώσεων** το οποίο αναλύει το ιστορικό του κάθε χρήστη και μειώνει την τιμή αγοράς ενός συγκεκριμένου προϊόντος, προσφέροντας έκπτωση. Υπάρχουν τρεις περιπτώσεις κατά τις οποίες ο χρήστης δικαιούται έκπτωση:

1. Εάν ο χρήστης έχει προσθέσει το ίδιο προϊόν σε **3 συνεχόμενες αγορές**, δικαιούται έκπτωση 20% για το ίδιο προϊόν. Η συγκεκριμένη έκπτωση ισχύει **μόνο** για την επόμενη αγορά του χρήστη.
(0.15 μονάδες)
2. Εάν ο χρήστης αγόρασε τουλάχιστον "X ποσότητα" από προϊόντα της ίδιας κατηγορίας, δικαιούται έκπτωση 30% σε ένα τυχαίο προϊόν της ίδιας κατηγορίας, στην επόμενη αγορά του. Η ποσότητα "X" είναι διαφορετική για κάθε κατηγορία και δίνεται στο αρχείο `files/discounts.txt`.
(0.15 μονάδες)
3. Ένας πελάτης θεωρείται **πιστός** εάν έχει κάνει τουλάχιστον 5 αγορές. Για τους πιστούς πελάτες, το σύστημα παρέχει έκπτωση 40% στο **αγαπημένο** τους **αντικείμενο**. Η συγκεκριμένη έκπτωση μπορεί να εξαργυρωθεί **μόνο μία φορά**. Ο αλγόριθμος επιλογής αγαπημένου αντικειμένου αποτελεί δική σας σχεδιαστική επιλογή. Οφείλετε να αναλύσετε την υλοποίησή σας και να εξηγήσετε γιατί αποτελεί σωστή επιλογή.
(0.2 μονάδες)

Εάν ένας χρήστης δικαιούται ταυτόχρονα παραπάνω από μία έκπτωση, το σύστημα επιλέγει τυχαία ποια έκπτωση θα εμφανιστεί για τη συγκεκριμένη αγορά. Ο χρήστης πρέπει να ενημερώνεται για την έκπτωση που δικαιούται πριν ολοκληρώσει την αγορά του. Δεν υπάρχουν unit tests για το κομμάτι του bonus.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!