# Επανάληψη εργαστηρίων 1,2

Γεώργιος Τσουμάνης, Άγγελος Δήμητσας, Βασίλειος Νάστος

November 10, 2022

## 1 Θέματα προετοιμασίας

1. Γράψτε πρόγραμμα που να δεσμεύει θέσεις μνήμης για μια ακέραια, μια πραγματική (double) και μια τιμή χαρακτήρα, να αναθέτει μια τιμή σε κάθε δεσμευμένη θέση μνήμης και να απελευθερώνει τη μνήμη.

2. Γράψτε μια συνάρτηση με όνομα circle που να δέχεται μια παράμετρο που να αντιστοιχεί στην ακτίνα ενός κύκλου και να επιστρέφει το εμβαδόν του κύκλου και την περίμετρο του κύκλου.

$$A = \pi r^2 \tag{1}$$

$$P = 2\pi r \tag{2}$$

3. Γράψτε πρόγραμμα που να δημιουργεί έναν πίνακα 5 θέσεων με τις τιμές 2,7,1,3,6. Να δέχεται από τον χρήστη μια ακέραια τιμή x και να αντιγράφει τον πίνακα x φορές σε έναν δυναμικό πίνακα τον οποίο στην συνέχεια να εμφανίζει στην οθόνη. Να αποδεσμεύει τον πίνακα πριν τον τερματισμό του προγράμματος

4. Δημιουργήστε μία κλαση Car που θα περιέχει 4 ιδιωτικά μέλη δεδομένα:

   - CarModel:string
   - CarName:string
   - CarPrice:double
   - CarOwner:string

   Στην κλάση δημιουργήστε getters και setters για όλα τα μέλη δεδομένα και μία συνάρτηση display που θα εμφανίζει όλα τα μέλη δεδομένα.

   Extra ερώτημα:Στην κύρια συνάρτηση δημιουργήστε ένα πίνακα με 10 αντικείμενα και ταξινομίστε τα με βάση την τιμή. Στην συνέχεια εμφανίστε όλα τα αντικείμενα στην οθόνη.

5. Δημιουργήστε κλάση Student η οποία θα έχει τα εξής μέλη δεδομένα:

   - name:string
   - semester:int
   - number_of_grades:int
   - grades:pointer to double

   Η κλάση θα έχει τις ακόλουθες συναρτήσεις μέλη:

   - Constructor χωρίς ορίσματα που θα αρχικοποιεί τα μέλη δεδομένα με τις ακόλουθες τιμές ("noname",-1,0,nullptr)
   - Constructor με 3 ορίσματα(string,int,int) που θα αρχικοποιεί τα μέλη δεδομένα. Το τρίτο όρισμα θα αντιστοιχεί στον αριθμό τον βαθμολογιών που μπορούν να προσθεθούν για ένα φοιτητή.
   - getters και setters για όλα τα μέλη δεδομένα
   - Συνάρτηση μέλος add_grade που θα προσθέτει στον πίνακα grades μία βαθμολογία. Σε περίπτωση που δεν μπορεί να προσθεθεί άλλος βαθμός να εμφανίζεται κατάλληλο μήνυμα.
   - Συνάρτηση average που θα υπολογίζει τον μέσο όρο ενός φοιτητή.

   Στην κύρια συνάρτηση:

   - Να δημιουργηθούν 10 φοιτητές.
   - Να βρεθεί ο φοιτητής με τον μέγιστο μέσο όρο και ο φοιτητής με τον μικρότερο μέσο όρο.

6. Να κατασκευαστεί πρόγραμμα σε C++ που θα πραγματοποιεί τα ακόλουθα:

   - Να ορίζεται η κλάση account (λογαριασμός τράπεζας). Η κλάση account θα πρέπει να περιέχει:
     - τα ιδιωτικά πεδία name (όνομα πελάτη) και balance (υπόλοιπο λογαριασμού)
     - Έναν constructor που θα αρχικοποιεί τα μέλη δεδομένα με τιμές ("",0.0).
     - Έναν constructor που θα αρχικοποιεί τα δεδομένα με παραμέτρους.
     - getters και setters για τα πεδία name και balance.
     - Μεθόδους για deposit (κατάθεση) ενός ποσού στον λογαριασμό και withdraw (ανάληψη) ενός ποσού από τον λογαριασμό. Η μέθοδος withdraw θα πρέπει να ελέγχει έτσι ώστε να μην επιτρέπει την ανάληψη μεγαλύτερου ποσού από το διαθέσιμο και σε περίπτωση που επιχειρηθεί κάτι τέτοιο να εμφανίζει σχετικό μήνυμα.
   - Να ορίζεται η κλάση bank (τράπεζα) που να περιέχει πίνακα με 10 λογαριασμούς. Συμπληρώστε τις κατάλληλες μεθόδους έτσι ώστε:
     - Να εμφανίζονται όλοι οι λογαριασμοί. Μέθοδος: print_accounts()

- Να πραγματοποιείται κατάθεση ενός ποσού σε έναν λογαριασμό δεδομένου του ονόματος του πελάτη. Μέθοδος: deposit(string, double)
- Να πραγματοποιείται ανάληψη ενός ποσού από έναν λογαριασμό δεδομένου του ονόματος του πελάτη. Μέθοδος withdraw(string, double)
- Να αποδίδεται τόκος βάσει ενός επιτοκίου και να προστίθεται στα υπόλοιπα όλων των λογαριασμών της τράπεζας. Μέθοδος add_interest(double). Για παράδειγμα αν το επιτόκιο είναι 1,5% τότε η μέθοδος θα καλείται ως bank_object.add_interest(1.5);

Στην κύρια συνάρτηση να δημιουργήσετε ένα αντικείμενο τύπου Bank. Εισάγεται τα στοιχεία για τους δέκα λογαριασμούς και πραγματοποιήστε προσθήκη 300 ευρώ για ένα πελάτη καθώς και αφαίρεση ενός για ένα πελάτη της επιλογής σας.

7. Να κατασκευαστεί κλάση Point η οποία θα έχει ως μέλη δεδομένα:

   - x_point:int
   - y_point:int

   Για την κλάση να δημιουργηθούν:

   - Setters και getters για τα δύο μέλη δεδομένα καθώς και ένας κατασκευαστής χωρίς ορίσματα ο οποίος θα αρχικοποιεί τα δύο μέλη δεδομένα με την τιμή 0.
   - Μία συνάρτηση distance η οποία θα δέχεται ως όρισμα ένα αντικείμενο τύπου Point και θα υπολογίζει την απόσταση μεταξύ δύο σημείων. Η απόσταση υπολογίζεται από την εξίσωση 3. Δύο αντικείμενα Point(5,5), Point(1,5) θα έχουν απόσταση $\sqrt{(5-1)^2+(5-5)^2}=4$.

$$d\left(p,q\right)=\sqrt{\sum_{i=1}^{n}\left(q_i-p_i\right)^2} \tag{3}$$

   Στην κύρια συνάρτηση να πραγματοποιηθούν τα εξής:

   - Να δημιουργηθούν 30 τυχαία σημεία στο εύρος $[0, 20]$.
   - Για κάθε σημείο να πραγματοποιηθεί εμφάνιση των συντεταγμένων τους.
   - Για κάθε σημείο να βρεθουν τα 5 κοντινότερα σε αυτό σημεία και να πραγματοποιηθεί εμφάνιση τους.

8. Για τη δευτεροβάθμια εξίσωση: $ax^2+bx+c=0$ Γράψτε μια συνάρτηση που να υπολογίζει τη διακρίνουσα. Γράψτε μια συνάρτηση που να επιστρέφει

τις ρίζες της δευτεροβάθμιας εξίσωσης καθώς και μια τιμή (1 ή 0) που να υποδηλώνει το εάν υπάρχουν πραγματικές ρίζες ή όχι.

$$D = b^2–4ac \tag{4}$$

$$x_{1,2} = \begin{cases} \dfrac{-b \pm \sqrt{D}}{2a}, & \text{if } D > 0 \\ 0, & \text{if } D = 0 \\ \dfrac{-b}{2a} & \text{otherwise} \end{cases}$$

9. Η ορμή(momentum) ορίζεται ως το προϊόν της μάζας ενός αντικειμένου και της ταχύτητας(velocity) του. Η μάζα είναι μια κλιμακωτή τιμή, ενώ Η ταχύτητα εκφράζεται γενικά ως πίνακας με τρεια στοιχεία. Η ορμή υπολογίζεται πολλαπλασιάζοντας την ταχύτητα επί την μαζα για κάθε στοιχείο. Γράψτε μια συνάρτηση με όνομα momentum και παραμέτρους:

   - Ένα δείκτη σε δεκαδικές τιμές(double) και
   - την τιμή της μάζα (double)

   και επιστρέψτε ένα δείκτη σε δεκαδικές τιμές που αντιστοιχει στην ορμή. Γράψτε ένα πρόγραμμα για να καθορίσετε τη μέση ορμή μιας συλλογής δεδομένων με τυχαίες ταχύτητες και μάζες.

   - Κατασκευάστε μια συνάρτηση με όνομα Randvec, χωρίς παραμέτρους η οποία θα επιστρέψει ένα δείκτη σε δεκαδικές τιμές που θα αντιστοιχεί στην ταχύτητα ενός δεδομένου. Κάθε στοιχείο στον πίνακα πρέπει να είναι μια τυχαία παραγόμενη τιμή στο διάστημα $[-100.0, 100.0]$.
   - Χρησιμοποιώντας το RandVec και τη λειτουργία momentum, δημιουργήστε 1000 εγγραφές, οι οποίες θα έχουν τυχαία ταχύτητα (όπως περιγράφεται παραπάνω) και μια τυχαία παραγόμενη Μάζα στο διάστημα $[1.0, 10.0]$. Αποθηκεύστε τις εγγραφές χρησιμοποιώντας κατάλληλη δομή της επιλογής σας.
   - Προσδιορίστε και εμφανίστε το μέσο ορμή των εγγραφών χρησιμοποιώντας ένα για βρόχο. [Tip: Ο μέσος όρος πρέπει να γίνει ανά στοιχείο.]

# 2 Αναφορές


- https://chgogos.github.io/oop/recitation/recitation.html
- https://www.studocu.com/row/document/comsats-university-islamabad/object-oriented-programming/oop-exercises-practice-problems-of-object-oriented-programming-in-java/13486070
- https://www.programiz.com/cpp-programming/memory-management
- https://gist.github.com/simonrenger/d1da2a10d11f8a971fc6f1b574ab3e99