

Αντικειμενοστραφής Προγραμματισμός-Εργαστήριο 6

Θέματα προετοιμασίας εξετάσεων

Γεώργιος Τσουμάνης, Άγγελος Δήμητσας, Βασίλειος Νάστος

February 5, 2023

1. Γράψτε ένα πρόγραμμα που παράγει 50 τυχαίους αριθμούς μεταξύ 0 και 200, υπολογίζει τα στατιστικά τους στοιχεία (ελάχιστο, μέγιστο, μέσος όρος και τυπική απόκλιση) και εκτυπώνει τα αποτελέσματα. Το πρόγραμμα θα πρέπει να κάνει χρήση μιας συνάρτησης που ονομάζεται `statistics` που λαμβάνει το διάνυσμα των τυχαίων αριθμών και τέσσερις παραμέτρους εξόδου (`min`, `max`, μέση τιμή και τυπική απόκλιση).

Η συνάρτηση θα πρέπει να υπολογίζει τα στατιστικά στοιχεία και να επιστρέφει τα αποτελέσματα μέσω των παραμέτρων εξόδου. Η κύρια συνάρτηση θα πρέπει να αρχικοποιεί τη γεννήτρια τυχαίων αριθμών, να παράγει τους τυχαίους αριθμούς, να καλεί τη συνάρτηση `statistics` και να εκτυπώνει τα αποτελέσματα.

2. Γράψτε ένα πρόγραμμα που περιέχει μια κλάση `Sphere` που αναπαριστά σφαίρες και τις ιδιότητές τους. Η κλάση έχει τα ακόλουθα χαρακτηριστικά:

- Μέλος δεδομένων `radius` για την αποθήκευση της ακτίνας μιας σφαίρας.
- Κατασκευαστής για την αρχικοποίηση της ακτίνας μιας σφαίρας.
- Destructor για την αποδέσμευση των πόρων που σχετίζονται με ένα αντικείμενο
- `setters/getters`
- Συνάρτηση μέλους `get_volume` για τον υπολογισμό του όγκου μιας σφαίρας
- Φιλική συνάρτηση `show` για την εμφάνιση των ιδιοτήτων μιας σφαίρας

Στην `main` υλοποιήστε τα ακόλουθα:

- Δημιουργείται ένα αντικείμενο σφαίρας `s1` με ακτίνα 4,5 και χρησιμοποιείται η συνάρτηση `show` για την εμφάνιση της ακτίνας και του όγκου του.
 - Με την κλήση του Destructor για το αντικείμενο `s1`, δημιουργείται ένα διάνυσμα αντικειμένων `Sphere`, `spheres`, για την αποθήκευση πολλαπλών αντικειμένων σφαίρας.
 - Χρησιμοποιώντας μια `while` για τη συνεχή εισαγωγή νέων σφαιρών στο διάνυσμα `spheres`. Ο χρήστης καλείται να εισάγει μια σφαίρα (1 για ναι, 0 για όχι), και ο βρόχος συνεχίζεται μέχρι ο χρήστης να αποφασίσει να σταματήσει πληκτρολογώντας 0. Κάθε νέο αντικείμενο σφαίρας που εισάγεται στο διάνυσμα `spheres` δημιουργείται με τυχαία ακτίνα μεταξύ 0 και 100, η οποία παράγεται χρησιμοποιώντας τη συνάρτηση `rand()`.
 - Κλήση της συνάρτησης `statistics` η οποία χρησιμοποιείται για τον υπολογισμό της μέσης ακτίνας και του μέσου όγκου των σφαιρών που είναι αποθηκευμένες στο διάνυσμα `spheres` και θα πρέπει να δηλωθεί νωρίτερα στο κώδικα. Η συνάρτηση `statistics` λαμβάνει δύο δείκτες σε τιμές `double`, `average_radius` και `average_volume`, οι οποίοι χρησιμοποιούνται για την αποθήκευση των υπολογισμένων μέσων τιμών.
 - Τέλος, ο αριθμός των σφαιρών που είναι αποθηκευμένες στο διάνυσμα `spheres`, η μέση ακτίνα και ο μέσος όγκος εμφανίζονται στο χρήστη.
3. Γράψτε ένα πρόγραμμα καλαθιού αγορών σε C++ χρησιμοποιώντας έννοιες αντικειμενοστραφούς προγραμματισμού. Το καλάθι αγορών θα έχει προϊόντα, βιβλία και ηλεκτρονικά είδη. Υλοποιήστε τα ακόλουθα:

- Δημιουργήστε μια βασική κλάση Product με τα ακόλουθα χαρακτηριστικά:
 - name - μια συμβολοσειρά που αντιπροσωπεύει το όνομα του προϊόντος
 - description - μια συμβολοσειρά που αντιπροσωπεύει την περιγραφή του προϊόντος
 - price - ένα double που αντιπροσωπεύει την τιμή του προϊόντος
- Δημιουργήστε δύο παράγωγες κλάσεις Book και Electronics από τη βασική κλάση Product. Οι παράγωγες κλάσεις θα έχουν τα ακόλουθα πρόσθετα χαρακτηριστικά:

Book:

- author - μια συμβολοσειρά που αντιπροσωπεύει τον συγγραφέα του βιβλίου
- copies - ένας ακέραιος αριθμός που αντιπροσωπεύει τον αριθμό των αντιτύπων του βιβλίου

Electronics:

- brand- μια συμβολοσειρά που αντιπροσωπεύει τη μάρκα του ηλεκτρονικού προϊόντος
- sales - ένας ακέραιος αριθμός που αντιπροσωπεύει τον αριθμό των πωλήσεων των ηλεκτρονικών ειδών

- Και για τις τρεις κλάσεις δημιουργήστε setters/getters για τα κατάλληλα πεδία

Στην main:

- Υπερκαλύψτε τις εικονικές μεθόδους value και show στις παράγωγες κλάσεις για να υπολογίσετε την αξία ενός βιβλίου και των ηλεκτρονικών και να εμφανίσετε τα στοιχεία τους, αντίστοιχα. Οι μέθοδοι θα πρέπει να λαμβάνουν υπόψη τα πρόσθετα χαρακτηριστικά των κλάσεων κατά τον υπολογισμό της αξίας και την εμφάνιση των λεπτομερειών
 - Δημιουργήστε ένα διάνυσμα δεικτών Product, με όνομα shopping_cart(vector <Product> shopping_chart), για την αποθήκευση των προϊόντων
 - Γεμίστε το καλάθι αγορών με 10 τυχαία προϊόντα χρησιμοποιώντας έναν βρόχο for και τη συνάρτηση rand. Τα προϊόντα μπορούν να είναι είτε βιβλία είτε ηλεκτρονικά είδη.
 - Υπολογίστε τη συνολική αξία του καλαθιού αγορών προσθέτοντας τις τιμές όλων των προϊόντων.
 - Μετά την κλήση του destructor για τα προϊόντα του διανύσματος, δημιουργήστε ένα αντικείμενο Product με τιμές("Shopping_Chart", "Total Shopping Chart products value", και για το price την συνολική τιμή των προϊόντων του διανύσματος) και εμφανίστε τα δεδομένα του.
4. Γράψτε ένα πρόγραμμα C++ για να χειρίζεστε μαθήματα και τις σχετικές πληροφορίες τους. Το πρόγραμμα περιλαμβάνει μια κλάση Course για την αναπαράσταση ενός μαθήματος με τις ιδιότητές id, name, semester, credits και average_grade. Δημιουργήστε:
- Setters/Getters
 - Μια συνάρτηση μέλος equal_semester που ελέγχει αν δύο μαθήματα ανήκουν στο ίδιο εξάμηνο
 - Μια συνάρτηση μέλος higher_average που συγκρίνει το μέσο βαθμό δύο μαθημάτων και μια συνάρτηση show που εμφανίζει τις πληροφορίες του μαθήματος σε μορφή πίνακα.
 - Μια συνάρτηση μέλος show που εμφανίζει τις πληροφορίες του μαθήματος

Για το πρόγραμμα δίνεται συνάρτηση read_courses(Listing 1) η οποία επιστρέφει ένα διάνυσμα με μαθήματα(vector <Course>), το οποίο περιέχει μαθήματα που διαβάζονται από ένα αρχείο με όνομα dit_courses.txt.

Listing 1: Κώδικας για συνάρτηση read_courses

```
vector <Course> read_courses ()
{
    string line, word;
    vector <Course> courses;
    vector <string> data;
    fstream fp;
    fp.open(".\\datasets\\course_dit.txt");
    while(getline(fp, line))
```

```

{
    data.clear();
    stringstream ss(line);
    while(getline(ss, word, ','))
    {
        data.emplace_back(word);
    }
    courses.push_back(Course(data[0], data[1], stoi(data[2]),
        stoi(data[3]), rand_grade()));
}
fp.close();
return courses;
}

```

Στην main:

- Εμφανίζει τις πληροφορίες όλων των μαθημάτων που βρίσκονται στο ίδιο εξάμηνο με το πρώτο μάθημα.
 - Ταξινομεί τα μαθήματα με βάση το μέσο όρο βαθμολογίας και εμφανίζει τα ταξινομημένα μαθήματα.
5. Δημιουργήστε μια πρότυπη συνάρτηση "matrixMultiplication" που δέχεται δύο πίνακες οποιουδήποτε τύπου δεδομένων και επιστρέφει το γινόμενό τους. Δοκιμάστε τη συνάρτησή σας πολλαπλασιάζοντας δύο πίνακες int και double.
 6. Δημιουργήστε μια πρότυπη συνάρτηση για την εκτέλεση δυαδικής αναζήτησης σε μια λίστα στοιχείων οποιουδήποτε τύπου δεδομένων. Στην main δοκιμάστε την συνάρτηση για ένα πίνακα ακεραίων, ένα πίνακα συμβολοσειρών και ένα πίνακα δεκαδικών τιμών.
 7. Δημιουργήστε μια πρότυπη συνάρτηση covariance(Equation 1) που δέχεται δύο πίνακες οποιουδήποτε τύπου δεδομένων και επιστρέφει τη συνδιακύμανσή τους. Δοκιμάστε τη συνάρτησή σας με πίνακες ακεραίων και διπλών.

$$Cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (1)$$

8. Έστω το τμήμα κώδικα του Listing 2. Αναπτύξτε ένα πρόγραμμα σε C++ χρησιμοποιώντας τον συγκεκριμένο πρότυπο κώδικα που υπολογίζει την ευκλείδεια απόσταση μεταξύ των δοκιμαστικών δειγμάτων και των δειγμάτων εκπαίδευσης.
 - Τα δείγματα εκπαίδευσης αποθηκεύονται σε ένα διάνυσμα 2D που ονομάζεται "data". Η τελευταία στήλη του πίνακα "data" αντιπροσωπεύει την ετικέτα κλάσης κάθε δείγματος (Δίνεται στο listing).
 - Τα δείγματα δοκιμής αποθηκεύονται σε ένα άλλο διάνυσμα 2D που ονομάζεται "test".
 - Η συνάρτηση "nearest_class" λαμβάνει τρεις παραμέτρους εισόδου: τα δείγματα εκπαίδευσης, ένα μόνο δείγμα δοκιμής και δύο παραμέτρους εξόδου (δείκτης συσχετισμένου δείγματος και ελάχιστη ευκλείδεια απόσταση).
 - Η συνάρτηση "nearest_class" υπολογίζει την ευκλείδεια απόσταση μεταξύ του δείγματος δοκιμής και κάθε δείγματος εκπαίδευσης και ενημερώνει τον "δείκτη συσχετιζόμενου δείγματος" και την "ελάχιστη ευκλείδεια απόσταση" με τον δείκτη και την απόσταση του πλησιέστερου δείγματος εκπαίδευσης, αντίστοιχα. **Στην main:**
 - Εκτυπώστε την ετικέτα κλάσης κάθε δείγματος δοκιμής και την αντίστοιχη ελάχιστη ευκλείδεια απόσταση.

Listing 2: Τμήμα κώδικα άσκησης 8

```

#include <iostream>
#include <vector>

```

```

using namespace std;

int main()
{
    vector <vector <double>> data={{0.12,0.26,0.35,0.55,1},
                                     {0.18,0.37,0.47,0.61,1},
                                     {0.15,0.31,0.43,0.62,1},
                                     {0.19,0.39,0.54,0.65,2},
                                     {0.13,0.26,0.33,0.57,1},
                                     {0.21,0.38,0.52,0.63,2},
                                     {0.17,0.32,0.45,0.60,1},
                                     {0.10,0.23,0.29,0.56,1},
                                     {0.20,0.35,0.50,0.64,2},
                                     {0.16,0.30,0.41,0.59,1}
    };

    vector <vector <double>> test={
                                     {0.25,0.40,0.45,0.58},
                                     {0.15,0.28,0.38,0.54},
                                     {0.18,0.32,0.41,0.57},
                                     {0.19,0.34,0.44,0.60}
    };

    // to be completed

    return 0;
}

```

9. Δημιουργήστε μια κλάση με το όνομα "Dish" που έχει τις ακόλουθες ιδιότητες:

- Μια συμβολοσειρά "name" για την αποθήκευση του ονόματος του πιάτου
- Μια συμβολοσειρά "description" για την αποθήκευση της περιγραφής του πιάτου
- Μια float "price" για την αποθήκευση της τιμής του πιάτου.
- Κατάλληλο κατασκευαστή που θα αρχικοποιεί τα μέλη δεδομένα με παραμέτρους
- Setters/Getters

Δημιουργήστε μια κλάση με όνομα "Menu" που έχει τις ακόλουθες ιδιότητες:

- Ένας πίνακας ή ένα διάνυσμα αντικειμένων Dish για την αποθήκευση της συλλογής των πιάτων του μενού
- Κατάλληλο κατασκευαστή
- Setters/Getters
- Μια μέθοδος με όνομα "addDish" για την προσθήκη ενός νέου πιάτου στο μενού
- Μια μέθοδος με όνομα "searchDish" που δέχεται το όνομα ενός πιάτου ως είσοδο και επιστρέφει το αντικείμενο πιάτο αν υπάρχει στο μενού, διαφορετικά επιστρέφει ένα μηδενικό ή κενό αντικείμενο
- Μια μέθοδος με όνομα "viewMenu" που εμφανίζει όλα τα πιάτα του μενού με το όνομα, την περιγραφή και την τιμή τους

Στην main, δημιουργήστε μια περίπτωση της κλάσης Menu, προσθέστε διάφορα πιάτα σε αυτήν, αναζητήστε ένα πιάτο και προβάλτε το μενού.

10. Γράψτε ένα πρόγραμμα C++ που διαβάζει ένα σύνολο δεδομένων θερμοκρασίας από ένα αρχείο κειμένου και εκτελεί τους ακόλουθους υπολογισμούς με κλήση μίας συνάρτησης monitoring:

- Αριθμός εγγραφών
- Μέση θερμοκρασία
- Μέγιστη θερμοκρασία
- Ελάχιστη θερμοκρασία
- Διάμεσος θερμοκρασίας
- **Extra ερώτημα:** Mode temperature (Θερμοκρασία με την υψηλότερη συχνότητα)

Στην κύρια συνάρτηση:

- Διαβάστε (Listing 3) και εμφανίστε τα δεδομένα στην Κονσόλα.
- Καλέστε την συνάρτηση monitoring και εμφανίστε τα αποτελέσματα.
- Εμφανίστε το σύνολο των θερμοκρασιών που ξεπερνούν τον μέσο όρο.

Τα δεδομένα είναι διαθέσιμα στον σύνδεσμο: <https://raw.githubusercontent.com/jbrownlee/Datasets/master/daily-min-temperatures.csv>

Listing 3: Συνάρτηση ανάγνωσης δεδομένων από αρχείο temperature.txt

```
#include <iostream>
#include <vector>
#include <fstream>
#include <sstream>

using namespace std;

vector <double> load_temperatures()
{
    vector <string> data;
    vector <double> temperatures;
    string line, word;
    bool start=true;

    fstream fp;
    fp.open("datasets\\temperatures.txt");
    if(!fp.is_open())
    {
        cerr<<"Error on opening datasets\\temperatures.txt"<<endl;
        exit(EXIT_FAILURE);
    }
    while(getline(fp, line))
    {
        if(start)
        {
            start=false;
            continue;
        }
        data.clear();
        stringstream ss(line);
        while(getline(ss, word))
        {
            data.push_back(word);
        }
        temperatures.push_back(stod(data[1]));
    }
    fp.close();
    return temperatures;
}
```