4η Προγραμματιστική άσκηση για το μάθημα
343 Εισαγωγή στον Προγραμματισμό
Μιχάλης Μπέκος / Κωνσταντίνα Τζουβάρα

Χειμερινό εξάμηνο 2024-2025
Πανεπιστήμιο Ιωαννίνων
29/10/2024

4η Προγραμματιστική Εργασία

Στόχοι: Εξοικείωση με τη δήλωση και υλοποίηση συναρτήσεων, χρήση κλάσεων ως παραμέτρους και ως τύπους επιστροφής, τοπικές μεταβλητές.

Άσκηση 1 (παλαιότερο θέμα εξετάσεων)

Ένα έτος είναι δίσεκτο είτε αν διαιρείται με το 400 είτε αν διαιρείται με το 4 και δεν διαιρείται με το 100.

- Να γραφεί η δήλωση και η υλοποίηση της συνάρτησης isLeapYear, η οποία δέχεται ως παράμετρο έναν ακέραιο αριθμό και επιστρέφει true εάν ο αριθμός αυτός αντιστοιχεί σε δίσεκτο έτος, και false διαφορετικά.
- Να υλοποιηθεί η συνάρτηση-οδηγός main(), η οποία ζητά από το χρήστη να δώσει στο τερματικό έναν ακέραιο και εκτυπώνει το αποτέλεσμα της παραπάνω συνάρτησης.

Άσκηση 2

Στη διάλεξη είδατε πως μπορεί κάποιος να γράψει συναρτήσεις, οι οποίες έχουν ως παραμέτρους αντικείμενα. Το παράδειγμα που χρησιμοποιήθηκε ήταν αυτό της συνάρτησης `add()`, η οποία παίρνει σαν παραμέτρους δύο αντικείμενα της κλάσης `Fraction` και επιστρέφει ένα νέο αντικείμενο που αντιστοιχεί στο άθροισμα τους.

Κατεβάστε από το e-course τα αρχεία Vector.hpp, Vector.cpp και main.cpp, τα οποία θα βρείτε στο συνοδευτικό υλικό του πρώτου μέρους των διαλέξεων και τροποποιήστε κατάλληλα και επεκτείνετε την υλοποίηση με τις παρακάτω συναρτήσεις:

`Vector add(const Vector &v1, const Vector &v2)`
Επιστρέφει ένα νέο αντικείμενο της κλάσης Vector το οποίο αντιστοιχεί στο άθροισμα v1`+`v2.
`Vector subtract(const Vector &v1, const Vector &v2)`
Επιστρέφει ένα νέο αντικείμενο της κλάσης Vector το οποίο αντιστοιχεί στη διαφορά v1`-`v2.
`Vector scale(double factor, const Vector &v)`
Επιστρέφει το διάνυσμα `factor * v`.
`bool equal(const Vector &v1, const Vector &v2)`
Επιστρέφει true αν τα διανύσματα v1 και v2 έχουν το ίδιο μέτρο.
`bool isGreater(const Vector &v1, const Vector &v2)`
Επιστρέφει true αν το μέτρο του v1 είναι (αυστηρώς) μεγαλύτερο από αυτό του v2.
`bool isSmaller(const Vector &v1, const Vector &v2)`
Επιστρέφει true αν το μέτρο του v1 είναι (αυστηρώς) μικρότερο από αυτό του v2.

Οι δηλώσεις των παραπάνω συναρτήσεων να γίνουν στο αρχείο Vector.hpp, ενώ οι αντίστοιχες υλοποιήσεις στο αρχείο Vector.cpp. Επεκτείνετε την υλοποίηση της συνάρτησης main() έτσι ώστε να ελέγξετε την ορθότητα όλων των παραπάνω συναρτήσεων.

Παρακάτω περιγράφεται ένας τρόπος υπερφόρτωσης του τελεστή + (αποτελεί ουσιαστικά απλή μετονομασία της συνάρτησης `add`), ο οποίος σας επιτρέπει να μπορείτε να προσθέσετε δύο αντικείμενα της κλάσης `Vector` με τον ίδιο ακριβώς τρόπο με τον οποίο θα προσθέτατε δύο βασικούς τύπους δεδομένων, π.χ. `int`.

```
Vector operator+(const Vector &v1, const Vector &v2) {
  return add(v1, v2); }
```

Εισάγετε τον παραπάνω κώδικα στην υλοποίηση σας και προσπαθήστε να καταλάβετε τι κάνει. Επίσης, επεκτείνετε την υλοποίηση της συνάρτησης `main()` έτσι ώστε να υπολογίσετε το άθροισμα των αντικειμένων v1 και v2 χρησιμοποιώντας την ακόλουθη γραμμή κώδικα:

```
Vector v = v1 + v2;
```

Αφού κατανοήσετε τον τρόπο λειτουργίας της συνάρτησης `operator+` η οποία σας επιτρέπει να προσθέσετε δύο αντικείμενα τύπου Vector χρησιμοποιώντας τον τελεστή +, να υλοποιήσετε με αντίστοιχο τρόπο τις παρακάτω συναρτήσεις:

`Vector operator-(const Vector &v1, const Vector &v2)`
Επιστρέφει ένα νέο αντικείμενο της κλάσης Vector το οποίο αντιστοιχεί στη διαφορά `v1-v2`.
`Vector operator*(double factor, const Vector &v1)`
Επιστρέφει το διάνυσμα `factor * v1`.
`bool operator==(const Vector &v1, const Vector &v2)`
Επιστρέφει true αν τα διανύσματα v1 και v2 έχουν το ίδιο μέτρο.
`bool operator>(const Vector &v1, const Vector &v2)`
Επιστρέφει true αν το μέτρο του v1 είναι (αυστηρώς) μεγαλύτερο από αυτό του v2.
`bool operator<(const Vector &v1, const Vector &v2)`
Επιστρέφει true αν το μέτρο του v1 είναι (αυστηρώς) μικρότερο από αυτό του v2.

Επεκτείνετε περαιτέρω την υλοποίηση της συνάρτησης main() έτσι ώστε να ελέγξετε την ορθότητα όλων των παραπάνω τελεστών. Τέλος, εμπλουτίστε τον κώδικα σας με σχόλια.