

Αντικειμενοστρεφής Προγραμματισμός I - C++

5ο Φυλλάδιο εργαστηρίου

Τετάρτη 30/03/2022

Σκοπός: Εξοικείωση με static μεταβλητές, static μέλη κλάσεων και με αναδρομικές συναρτήσεις. Υλοποίηση παραδειγμάτων που απαιτούν σύνθεση κλάσεων.

Άσκηση 1η

α) Ποιο θα είναι το αποτέλεσμα της εκτέλεσης του παρακάτω προγράμματος. Δικαιολογήστε την απάντησή σας.

```
#include <iostream>
using namespace std;

void SumNums(int);

int main() {
    int count=1;

    while (count <= 3) {
        SumNums(count);
        count++;
    }

    return 0;
}

void SumNums(int count) {
    static int sum=0;

    sum = sum + count;
    cout << "H timi tis metavlitis sum einai : " << sum << endl;
}
```

β) Τροποποιήστε το πρόγραμμα έτσι ώστε να προκύπτει το ίδιο αποτέλεσμα αλλά με την χρήση καθολικών μεταβλητών.

Απάντηση

α) Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι :

```
H timi tis metavlitis sum einai: 1
H timi tis metavlitis sum einai: 3
H timi tis metavlitis sum einai: 6
```

Αυτό οφείλεται στο γεγονός ότι η μεταβλητή sum είναι στατική οπότε διατηρεί την τιμή της πάντα μετά την εκτέλεση της συνάρτησης.

β)

```
#include <iostream>
using namespace std;

int sum=0;
```

```

void SumNums(int);

int main(){
int count=1;

while (count <= 3) {
    SumNums(count);
    count++;
}

return 0;
}

void SumNums(int count) {
sum = sum + count;
cout << " Η timi tis metavlitis sum einai : " << sum << endl;
}

```

Άσκηση 2η

Υλοποιήστε πρόγραμμα το οποίο θα ορίζει κλάση που αναπαριστά ένα ορθογώνιο παραλληλόγραμμο. Η κλάση θα έχει ως μέλη τις δύο πλευρές του ορθογωνίου και κατάλληλες συναρτήσεις για υπολογισμό εμβαδού και εμφάνισης των στοιχείων του. Επιπροσθέτως η κλάση πρέπει να διατηρεί σε κατάλληλες μεταβλητές το πλήθος των αντικειμένων-ορθογωνίων παραλληλογράμμων που έχουμε δημιουργήσει αλλά και τη συνολική τους επιφάνεια.

Απάντηση

```

#include <iostream>
#include <cmath>

using namespace std;

class Rectangle {
private:
    float sideA;
    float sideB;
public:

// Δήλωση των στατικών-μελών της κλάσης. Οι μεταβλητές αυτές θα δηλωθούν
// και εκτός της κλάσης και μόνο τότε δεσμεύεται η μνήμη που χρησιμοποιούν.

// Οι μεταβλητές αυτές καταλαμβάνουν χώρο στη μνήμη μόνο μια φορά για ολόκληρη τη
// κλάση, και τις «μοιράζονται» όλα τα αντικείμενα της κλάσης.
    static int countRectangle;
    static float totalSurface;

    Rectangle(float a, float b);
    ~Rectangle();

    float area() {return sideA * sideB;}
    void setSides(float a, float b) {sideA = a; sideB = b;}
    void print() {
        cout << endl << "Side A: " << sideA << endl;
        cout << "Side B: " << sideB << endl;
        cout << "Rectangle Area: " << area() << endl << endl;
    }
}

```

```

};

// Αν δεν αρχικοποιήσουμε εμείς τις μεταβλητές εξ ορισμού αρχικοποιούνται με 0.
int Rectangle::countRectangle=0;
float Rectangle::totalSurface=0;

Rectangle::Rectangle(float a, float b) {
    sideA = a;
    sideB = b;

    // Στη συνάρτηση δόμησης κάθε φορά που δημιουργείται ένα αντικείμενο, αυξάνεται η
    // στατική μεταβλητή που μετρά τα αντικείμενα και προστίθεται και το εμβαδόν του
    // νέου αντικειμένου
    countRectangle++;
    totalSurface = totalSurface + area();
}

Rectangle::~Rectangle() {
    // Στη συνάρτηση αποδόμησης κάθε φορά που διαγράφεται ένα αντικείμενο από τη
    // μνήμη, μειώνεται κατά 1 η στατική μεταβλητή που μετρά τα αντικείμενα και
    // αφαιρείται και το εμβαδόν του αντικειμένου που διαγράφηκε
    countRectangle--;
    totalSurface = totalSurface - area();
}

int main() {
    Rectangle rect1(6, 9), rect2(5, 8);
    rect1.print();
    rect2.print();

    // Πρόσβαση στα στατικά μέλη της κλάσης.
    cout << "Number of objects: " << Rectangle::countRectangle << endl;
    cout << "Total Surface: " << Rectangle::totalSurface << endl;

    {
        Rectangle rect3(4, 6);
        cout << "Number of objects: " << Rectangle::countRectangle << endl;
        cout << "Total Surface: " << Rectangle::totalSurface << endl;
    }

    cout << "Number of objects: " << Rectangle::countRectangle << endl;
    cout << "Total Surface: " << Rectangle::totalSurface << endl;

    return 0;
}

```

Άσκηση 3η

Δημιουργήστε μια κλάση η οποία αναπαριστά ένα μαθητή (**Student**). Ο μαθητής περιγράφεται από το ονοματεπώνυμο του, τον αριθμό μητρώου του και από 2 μαθήματα που παρακολούθησε στο τρέχον εξάμηνο.

Για να ορίσετε το μάθημα θα πρέπει να ορίσετε και να χρησιμοποιήσετε μια κλάση (**Course**) η οποία αναπαριστά ένα μάθημα. Ένα μάθημα περιγράφεται από τον κωδικό του, τον βαθμό εργαστηρίου και τον βαθμό της τελικής γραπτής εξέτασης.

Η κλάση Course να περιέχει συνάρτηση η οποία να υπολογίζει και να επιστρέφει το τελικό βαθμό

του μαθήματος (40% βαθμός εργαστηρίου + 60% βαθμός της τελικής γραπτής εξέτασης).

Η κλάση Student θα πρέπει να περιέχει δομητή (constructor) για την αρχικοποίηση του ονόματος και του αριθμού μητρώου. Επιπλέον να περιέχει δύο συναρτήσεις μία για την εισαγωγή των στοιχείων του για κάθε ένα από τα μαθήματα (κωδικός και βαθμολογίες) και μία για την εμφάνιση όλων των στοιχείων του μαθητή καθώς και του τελικού βαθμού για κάθε μάθημα.

Στην main συνάρτηση ορίστε 2 μαθητές και εμφανίστε τα στοιχεία τους.

Απάντηση

```
#include <iostream>
using namespace std;

class Course {
private:
    string courseId;
    float labScore;
    float examScore;

    float finalCourse() {
        return 0.40*labScore + 0.60*examScore;
    }

public:
    void insertElements() {
        cout << "Eisagete ton kwdiko: " << endl;
        cin >> courseId;
        cout << "Eisagete tin vathmologia tou ergastiriou: " << endl;
        cin >> labScore;
        cout << "Eisagete tin vathmologia tis exetasis: " << endl;
        cin >> examScore;
    }

    void displayElements() {
        cout << "Kwdikos tou mathimatos: " << courseId << endl;
        cout << "Vathomologia ergastiriou: " << labScore << endl;
        cout << "Vathomologia exetasis: " << examScore << endl;
        cout << "Teliki Vathmologia: " << finalCourse() << endl;
        cout << endl;
    }
};

class Student {
private:
    string studentName;
    string studentId;
    Course course1;
    Course course2;

public:
    Student(string pstudentName, string pstudentId) {
        studentName = pstudentName;
        studentId = pstudentId;
    }

    void insertCourses() {
        cout << "Eisagete ta stoixeia tou " << "lou mathimatos: " << endl;
        course1.insertElements();
```

```

        cout << "Eisagete ta stoixeia tou " << "2ou mathimatos: " << endl;
        course2.insertElements();
    }

    void displayStudent() {
        cout << "To onoma tou foititi einai: " << studentName << endl;
        cout << "o kwdikos tou: " << studentId << endl;
        cout << "Ta stoixeia twm mathimatwn tou einai" << endl << endl;

        course1.displayElements();
        course2.displayElements();
    }
};

int main() {
    Student std1("Anna Iwannou", "321/2021001");
    std1.insertCourses();
    std1.displayStudent();

    Student std2("Kwstas Vlachos", "321/2021010");
    std2.insertCourses();
    std2.displayStudent();
    return 0;
}

```

Άσκηση 4η

Ένας άνθρωπος χαρακτηρίζεται από τα στοιχεία: όνομα, επώνυμο, έτος γέννησης και τη διεύθυνση κατοικίας. Μια διεύθυνση αποτελεί μια σύνθετη δομή και περιλαμβάνει την οδό, τον αριθμό, την πόλη και τον ταχυδρομικό κώδικα.

Υλοποιήσετε πρόγραμμα σε C++, ορίζοντας τις κατάλληλες κλάσεις, το οποίο να δημιουργεί δυο μέλη της οικογένειας σας και ακολούθως να εμφανίζει τα στοιχεία τους.

Ακολουθεί ενδεικτική εμφάνιση από την εκτέλεση της εφαρμογής:

*Nikos Andreou is 37 years old and stays at: Pythagora 25, Karlovasi, 83200.
Lilly Andreou is 33 years old and stays at: Aristarchou 10, Karlovasi, 83200.*

Απάντηση

```

#include <iostream>
using namespace std;

/*
 * Class Address
 */
class Address {
    string street;
    int number;
    string city;
    int poCode;

public:
    // setters
    void set_street(string p_street) {
        street = p_street;
    }
}

```

```

    void set_number(int p_number) {
        number = p_number;
    }

    void set_poCode(int p_poCode) {
        poCode = p_poCode;
    }

    void set_city(string p_city) {
        city = p_city;
    }

    void print() const {
        cout << street << " " << number << ", " << city << ", " << poCode << endl;
    }
};

/*
 * Class Person
 */
class Person {
    string name;
    string surname;
    int birthYear;
    Address pAddress;

public:
    // Constructor
    Person(string p_surname, string p_name, int p_birthYear, string p_street, int
p_number, int p_poCode, string p_city) {
        surname = p_surname;
        name = p_name;
        birthYear = p_birthYear;
        pAddress.set_street(p_street);
        pAddress.set_number(p_number);
        pAddress.set_poCode(p_poCode);
        pAddress.set_city(p_city);
    }

    void print() {
        cout << name << " " << surname << " is " << 2022-birthYear << " years old
and stays at: ";
        pAddress.print();
    }
};

int main() {
    // Δημιουργούμε 2 άτομα και εμφανίζουμε τα στοιχεία τους και τη
    // διεύθυνσή τους αντίστοιχα
    Person p1("Andreou", "Nikos", 1985, "Pythagora", 25, 83200, "Karlovasi");
    p1.print();
    Person p2("Andreou", "Lilly", 1989, "Aristarchou", 10, 83200, "Karlovasi");
    p2.print();
    return 0;
}

```

Άσκηση 5η

Έστω η ακόλουθη κλάση σε C++:

```
class RandomTest {
    int number;
public:
    void display() { cout << number << endl; }
};
```

Συμπληρώστε κατάλληλα την κλάση έτσι ώστε όταν δημιουργείται ένα νέο αντικείμενο, να καταχωρείται στο μέλος number ένας τυχαίος αριθμός στο διάστημα [0, 200] και να εμφανίζεται το μήνυμα "Δημιουργήθηκε αντικείμενο με τιμή xxx" όπου xxx ο τυχαίος αριθμός. Όταν καταστρέφεται ένα αντικείμενο να εμφανίζεται το μήνυμα "Καταστράφηκε αντικείμενο με τιμή xxx". Δημιουργείστε κατάλληλη main() συνάρτηση για να ελέγξετε την υλοποίησή σας.

Απάντηση

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

class RandomTest {
    int number;
public:
    RandomTest();
    ~RandomTest();
    void display() {
        cout << number << endl;
    }
};

RandomTest::RandomTest() {
    number = rand() % 201;
    cout << "Object created with value: " << number << endl;
}

RandomTest::~~RandomTest () {
    cout << "Object deleted with value: " << number << endl;
}

int main() {
    srand(time(NULL));

    RandomTest t;
    // Δοκιμή εμβέλειας
    {
        RandomTest r;
    }
    RandomTest s;
    return 0;
}
```

Άσκηση 6η

Να εντοπίσετε το λάθος στην επόμενη αναδρομική συνάρτηση:

```
int rec(int n) {  
    int s;  
  
    s = n + rec(n-1);  
    return s;  
}
```

Απάντηση

Το πρόβλημα στην συγκεκριμένη αναδρομική συνάρτηση είναι ότι δεν υπάρχει στον κώδικα μία τουλάχιστον συνθήκη τερματισμού, δηλαδή περίπτωση όπου η συνάρτηση δεν θα καλέσει τον εαυτό της.

Άσκηση 7η

Υλοποιήστε *αναδρομική* συνάρτηση με όνομα *printVertical* η οποία να δέχεται ως παράμετρο έναν θετικό ακέραιο αριθμό και να τον εμφανίζει στην οθόνη με τα ψηφία να εμφανίζονται το ένα κάτω από το άλλο, δηλαδή ένα ψηφίο σε κάθε γραμμή. Για παράδειγμα η κλήση *printVertical(1942)* θα εμφανίσει:

```
1  
9  
4  
2
```

Καλέστε την συνάρτηση με διαφορετικές τιμές παραμέτρου για να επιδείξετε την λειτουργία της.

Απάντηση

```
#include <iostream>  
using namespace std;  
  
void printVertical(long);  
  
int main() {  
    int number;  
  
    cout << "Enter a positive number: ";  
    cin >> number;  
  
    printVertical(number);  
  
    return 0;  
}  
  
void printVertical(long n) {  
    if (n < 10)  
        cout << n << endl;  
    else {  
        printVertical(n/10);  
        cout << n % 10 << endl;  
    }  
}
```


Άσκηση 8η

Τι θα εμφανιστεί κατά την εκτέλεση του παρακάτω προγράμματος;

```
#include <iostream>
using namespace std;

void tryMe();

main() {
    tryMe();
    return 0;
}

void tryMe() {
    static int times=0;

    if(times < 10) {
        cout << "Function Call: " << times + 1 << endl;
        times++;
        tryMe();
    }
}
```

Απάντηση

Θα εμφανίσει:

```
Function Call: 1
Function Call: 2
Function Call: 3
Function Call: 4
Function Call: 5
Function Call: 6
Function Call: 7
Function Call: 8
Function Call: 9
Function Call: 10
```