

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
Κ10: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ**

Εξετάσεις 26 Ιανουαρίου 2018

1. (α') Τι εννοούμε με τον όρο "εμβέλεια" (scope) στον προγραμματισμό; Γιατί είναι κρίσιμη για τον αντικειμενοστραφή προγραμματισμό;
- (β') Δίδεται το παρακάτω πρόγραμμα Java:

```
public class MainClass {
public static void main(String[] args) { A a;
    a = new C(); a.print(); a = new B(); a.print();
    a = new A(); a.print(); } }

class A{ private int data1;
    {System.out.println("The initial value of A data is: " + data1);}
    A() {this(5);}
    A(int i) { data1 = i;
        System.out.println("The given value of A data is: " + data1);}
    void print() {System.out.println("The current value of A data is: " + data1);} }

class B extends A{
    B() { super(10);}
    B(int i ){super(i);} }

class C extends B{ private int data2;
    C(){this(100);}
    C(int i){super(i); data2 = i;
        System.out.println("The given value of C data is: " + data2);}
    void print() {super.print();
        System.out.println("The current value of C data is: " + data2);} }
```

Επίσης, δίδεται και το παρακάτω πρόγραμμα C++:

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

class A{ int data1;
public:
    A():data1(5) {cout << "The given value of A data is: " << data1 << endl;}
    A(int i):data1(i) { cout << "The given value of A data is: " << data1 << endl;}
    void print() {cout << "The current value of A data is: " << data1 << endl;} };

class B : public A{
public:
    B():A(10) {}
    B(int i ):A(i){} };

class C : public B{ int data2;
public:
    C():data2(100),B(100){cout << "The given value of C data is: " << data2 << endl;}
    C(int i):data2(i),B(i){cout << "The given value of C data is: " << data2 << endl;}
    void print() {B::print();
        cout << "The current value of C data is: " << data2 << endl;} };
```

```
int main(){ A* pa;
    pa = new C(); pa->print(); pa = new B(); pa->print();
    pa = new A(); pa->print(); }
```

Δώστε τα αποτελέσματα των δύο προγραμμάτων, αιτιολογώντας την απάντησή σας. Τα αποτελέσματα των δύο προγραμμάτων διαφέρουν. Αιτιολογήστε γιατί. Τροποποιήστε το πρόγραμμα C++ ώστε να δίνει το ίδιο αποτέλεσμα με το Java πρόγραμμα.

2. Αιτιολογώντας την απάντησή σας, δώστε το αποτέλεσμα της εκτέλεσης του παρακάτω προγράμματος C++:

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

class A{ int data;
public:
    A(int i=10):data(i) {cout << "An A just created" << endl;}
    A(const A& a):data(a.data) {cout << "An A just created with CP" << endl;}
    ~A(){cout << "An A to be destroyed with data" << data << endl;}
    void increase(int i) {data*=i;} };

class A1{A data1; A& data2;
public:
    A1(A& a):data1(a),data2(a)
        {cout << "An A1 just created" << endl;}
    ~A1(){cout << "An A1 to be destroyed" << endl;}
    void process() {data1.increase(2); data2.increase(2);} };

class A2{A data1; A* data2;
public:
    A2(const A& a):data1(a) {data2 = new A(a);
        cout << "An A2 just created" << endl;}
    A2(const A2& a2):data1(a2.data1) {data2 = new A(*(a2.data2));
        cout << "An A2 just created with CP" << endl;}
    ~A2(){cout << "An A2 to be destroyed" << endl; delete data2;}
    void process() {data1.increase(10); data2->increase(10);} };

class B{A1& data1; A2 data2;
public:
    B(A1& a1, A2& a2): data1(a1),data2(a2)
        {cout << "A B just created" << endl;}
    ~B(){cout << "A B to be destroyed" << endl;}
    void process(){data1.process(); data2.process();} };

int main(){ A a; A1 a1(a); A2 a2(a);
    B b1(a1,a2); B b2(b1);
    b1.process(); b2.process(); a1.process(); a2.process(); }
```

3. Υλοποιήστε σε C++ την πληροφορία μιας απλουστευμένης βάσης φιλομογραφίας (examK10mdb). Η βάση περιέχει πληροφορίες για (κινηματογραφικές) ταινίες (films), ηθοποιούς (actors) και σκηνοθέτες (directors).

Οι χρήστες μπορούν να ψηφίζουν τους αγαπημένους τους ηθοποιούς και τις αγαπημένες τους ταινίες. Ο ηθοποιός που θα συγκεντρώσει τις περισσότερες ψήφους χαρακτηρίζεται ως ο *πλέον δημοφιλής ηθοποιός*. Ο σκηνοθέτης ο οποίος έχει σκηνοθετήσει ταινίες που συνολικά έχουν συγκεντρώσει τις περισσότερες ψήφους χαρακτηρίζεται ως ο *πλέον δημοφιλής σκηνοθέτης*.

Τόσο οι ηθοποιοί όσο και οι σκηνοθέτες έχουν ονοματεπώνυμο, ημερομηνία γέννησης κι ένα σύνολο ταινιών στις οποίες έχουν συμμετάσχει. Ένας ηθοποιός έχει καταχωρημένο και το πλήθος ψήφων που έχει λάβει.

Τόσο οι ηθοποιοί όσο και οι σκηνοθέτες αρχικοποιούνται με τα στοιχεία του ονοματεπωνύμου τους και της ημερομηνίας γέννησής τους. Αρχικά, θεωρείται ότι δεν έχουν συμμετάσχει σε καμία ταινία. Οι ηθοποιοί, αρχικά, δεν έχουν καμιά ψήφο.

Οι ηθοποιοί και οι σκηνοθέτες συμμετέχουν σε μια ταινία (*participates*), προσθέτοντας την ταινία αυτή σε αυτές που έχουν συμμετάσχει.

Οι ηθοποιοί λαμβάνουν ψήφο (*get_voted*) αυξάνοντας το πλήθος των ψήφων τους κατά 1. Το πλήθος αυτό των ψήφων τους μπορεί να ανακτηθεί (*get_votes*).

Οι σκηνοθέτες συλλέγουν ψήφους (*get_votes*) αιθροίζοντας τις ψήφους όλων των ταινιών τις οποίες έχουν σκηνοθετήσει.

Οι ταινίες έχουν έναν τίτλο, ένα έτος παραγωγής, έναν σκηνοθέτη, ένα σύνολο ηθοποιών που συμμετέχουν και το πλήθος ψήφων που έχουν λάβει.

Αρχικά, στις ταινίες καταχωρούνται ο τίτλος τους, το έτος παραγωγής τους, οι ηθοποιοί τους και ο σκηνοθέτης τους. Αρχικά, οι ψήφοι των ταινιών είναι μηδέν.

Μια ταινία λαμβάνει ψήφους (*get_voted*) προσθέτοντας μια μονάδα στις ψήφους της. Το πλήθος αυτό των ψήφων τους μπορεί να ανακτηθεί (*get_votes*).

Τόσο οι ηθοποιοί και οι σκηνοθέτες όσο και οι ταινίες παρέχουν τη δυνατότητα εκτύπωσης των στοιχείων τους (*print*).

Η βάση της φιλομογραφίας περιέχει ηθοποιούς, σκηνοθέτες και ταινίες. Αρχικοποιείται με κάποιους ηθοποιούς, σκηνοθέτες και ταινίες.

Στη βάση, μπορούμε:

- Να ψηφίσουμε έναν ηθοποιό, ψηφίζοντας τον ηθοποιό (*vote*)
- Να ψηφίσουμε μια ταινία, ψηφίζοντας την ταινία (*vote*)
- Να ανακτήσουμε τον “πλέον δημοφιλή ηθοποιό” (*get_favorite_actor*)
- Να ανακτήσουμε τον “πλέον δημοφιλή σκηνοθέτη” (*get_favorite_director*)

Υλοποιήστε την παραπάνω πληροφορία σε C++, υλοποιώντας κατάλληλες κλάσεις, ορίζοντας τα μέλη-δεδομένα που χρειάζονται και συναρτήσεις-μέλη που υλοποιούν την παραπάνω συμπεριφορά. Θεωρήστε δεδομένο έναν τύπο λίστας, δίδοντας, όμως, τις προδιαγραφές του. (*Σημείωση: Δεν χρειάζεται υλοποίηση συνάρτησης main*).