

Αντικειμενοστρεφής Προγραμματισμός Ι

8ο Φυλλάδιο εργαστηρίου

15/05/2023

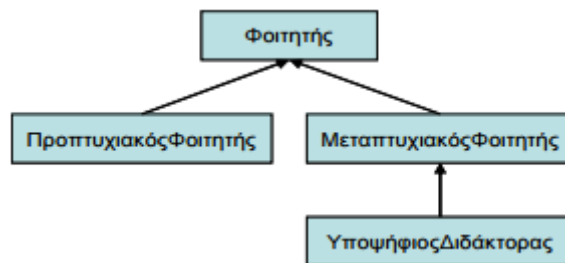
Σκοπός: Προγραμματίζοντας με απλή και πολλαπλή κληρονομικότητα. Χρήση πολυμορφισμού, virtual & pure virtual συναρτήσεων και αφηρημένων (abstract) κλάσεων.

Άσκηση 1η

Τροποποιήστε τον κώδικα της τελευταίας άσκησης του 7ου φυλλαδίου, ορίζοντας στην main συνάρτηση πίνακα αντικειμένων στον οποίο θα αποθηκεύσετε ζώακια τύπου **Dog** και **Turtle**. Για κάθε ένα από αυτά τα αντικείμενα καλέστε τη συνάρτηση display για να εμφανίσετε όλα τα στοιχεία τους.

Άσκηση 2η

Δίνεται η παρακάτω ιεραρχία κλάσεων:



Σύμφωνα με την παραπάνω ιεραρχία κλάσεων υλοποιήστε εφαρμογή σε C++ που διαχειρίζεται τους διαφορετικούς τύπους φοιτητών/τριών ενός πανεπιστημίου. Τα βασικά στοιχεία που πρέπει να διατηρούνται για κάθε φοιτητή/τρια είναι το όνομα, το επίθετο, το μοναδικό προσδιοριστικό του και ο αριθμός των υποχρεωτικών μαθημάτων που έχει περάσει. Επιπλέον για τους προπτυχιακούς πρέπει να καταγραφεί και ο αριθμός των εργαστηρίων που έχουν περάσει. Ομοίως για τους μεταπτυχιακούς φοιτητές θα πρέπει να γνωρίζουμε και το θέμα της διπλωματικής εργασίας που εκπονούν. Τέλος, για τους υποψήφιους διδάκτορες που, όπως φαίνεται και από το διάγραμμα είναι μια κατηγορία μεταπτυχιακών φοιτητών, θα πρέπει να γνωρίζουμε πόσα χρόνια έρευνας έχουν εκπονήσει. Υλοποιήστε τους κατάλληλους δομητές (constructors).

Επίσης γράψτε λειτουργίες για τα ακόλουθα:

- Αύξηση αριθμού μαθημάτων που έχει περάσει ο φοιτητής.
- Αύξηση αριθμού εργαστηρίων που έχει περάσει ο προπτυχιακός φοιτητής.
- Έλεγχος εάν ο φοιτητής πληροί τις προϋποθέσεις για την αποφοίτηση του (ο προπτυχιακός οφείλει 40 μαθήματα και 15 εργαστήρια, ο μεταπτυχιακός οφείλει 10 μαθήματα και ο διδακτορικός θα πρέπει να έχει περάσει 3 μαθήματα και να έχει διεξάγει 4 χρόνια έρευνας).
- Εμφάνιση όλων των στοιχείων του φοιτητή.

Στο κυρίως πρόγραμμα (συνάρτηση main) ορίστε αντικείμενα και από τις 3 κατηγορίες φοιτητών. Δοκιμάστε να καλέσετε τις συναρτήσεις που υλοποιήσατε. Θα πρέπει να εκμεταλλευτείτε στην υλοποίησή σας τον μηχανισμό της κληρονομικότητας.

Άσκηση 3η

Τι αποτέλεσμα θα έχει το παρακάτω πρόγραμμα; Δικαιολογήστε την απάντησή σας.

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

class sxima {
```

```

    string color;
public:
    void set_col(string c) {color = c;}
    void show() {cout << "Sxima xrwmatos " << color << endl;}
};

class kyklos: public sxima {
    float aktina;
public:
    void set_r(float r) {aktina = r;}
    void show() {
        cout << "Kyklos aktinas " << aktina << endl;
        sxima::show();
    }
};

class tetragono : public sxima {
    float plevra;
public:
    void set_a(float a) {plevra = a;}
    void show() {
        cout << "Tetragwno plevras " << plevra << endl;
        sxima::show();
    }
};

int main() {
    kyklos k1;
    tetragono t1;
    sxima *ptr1 = &t1;

    t1.set_a(10);
    t1.set_col("Kokkino");
    t1.show();

    k1.set_r(5);
    k1.set_col("Mple");
    k1.show();

    cout << endl << "Call using pointer" << endl << endl;
    ptr1->show();
    ptr1 = &k1;
    ptr1->show();
    return 0;
}

```

Τι αλλαγές θα πρέπει να γίνουν στον παραπάνω κώδικα ώστε, όταν καλείται η συνάρτηση show() μέσω του δείκτη ptr1, να καλείται η συνάρτηση-μέλος show() της κλάσης των αντικειμένων στα οποία δείχνει ο δείκτης.

Άσκηση 4

Συμπληρώστε τον ακόλουθο κώδικα ώστε να εκτελείται σωστά το κυρίως πρόγραμμα. Πρέπει να ορίσετε την κλάση που λείπει καθώς και να τροποποιήσετε τον υπάρχοντα κώδικα στην κλάση Shape.

```

#include <iostream>
using namespace std;

class Shape {
public:
    virtual void draw () = 0;
    void print () { cout << "I am a shape" << endl; }
};

// Συμπληρώστε κατάλληλο κώδικα ...

```

```
int main () {
    Shape *a = new Box();
    a->print(); // πρέπει να τυπώνει "I am a box"
}
```

Άσκηση 5

Δίνεται ο ορισμός της κλάσης Employee. Ποια από τα παρακάτω είναι έγκυρα;

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

class Employee {
private:
    string name;
    string id;
public:
    Employee() {}
    Employee(string name, string id) {
        this->name = name;
        this->id = id;
    }

    string getName() const { return name; }
    string getId() const { return id; }

    void setName(string name) { this->name = name; }
    void setId(string id) { this->id = id; }
    virtual void printCheck() const = 0;
};
```

α)

```
int main() {
    Employee george;
}
```

β)

```
class HourlyEmployee: public Employee {
private:
    double wageRate;
    double hours;

public:
    HourlyEmployee() {}
    void printCheck() const {
        cout << "Check Amount: " << wageRate * hours;
    }
};
```

γ)

```
int main() {
    Employee *george = new HourlyEmployee();
}
```

δ)

```
bool isBossOf(const Employee& emp1, const Employee& emp2) {
    ...
}
```

Άσκηση 6

Σχολιάστε τον ακόλουθο κώδικα. Έχει συντακτικά λάθη; Τι θα εμφανίσει κάθε μια από τις κλήσεις των λειτουργιών print().

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

class Pet {
// Ορίσαμε τις μεταβλητές μέλη δημόσιες για να
// κρατήσουμε το παράδειγμα πιο απλό
public:
    string name;
    virtual void print() const;
};

class Dog : public Pet {
public:
    string breed;
    virtual void print() const;
};

void Pet::print() const {
    cout << "Name: " << name << endl << endl;
}

void Dog::print() const {
    cout << "Name: " << name << endl;
    cout << "Breed: " << breed << endl << endl;
}

int main() {
    Dog vdog;
    Pet vpet;

    vdog.name = "Orfeas";
    vdog.breed = "Maltez";
    vpet = vdog;
    vpet.print();
    cout << vpet.breed;

    Pet *ppet;
    ppet = new Pet;
    Dog *pdog;
    pdog = new Dog;

    pdog->name = "Orfeas";
    pdog->breed = "Maltez";
    ppet = pdog;

    ppet->print();
    pdog->print();

    return 0;
}
```

Άσκηση 7

Στο ακόλουθο πρόγραμμα σας δίνεται η main συνάρτηση και θα πρέπει να συμπληρώσετε τις υπόλοιπες κλάσεις. Στα σχόλια μπορείτε να δείτε ενδεικτική εκτέλεση του προγράμματος.

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

class Animal {
```

```

...
};

class Pig: public Animal {
...
};

class Cow: public Animal {
...
};

class Farm {
// Η φάρμα μπορεί να φιλοξενήσει το πολύ 100 ζώακια
...
};

void show(const Farm &f) {
    for ( int i = 0; i < f.getNumOfAnimals(); i++ )
        cout << f.getAnimal(i)->getName() << " ("
        << f.getAnimal(i)->getType() << ") "
        << f.getAnimal(i)->getWeight() << endl;
}

int main() {
    Farm farm(2); // Αρχικοποίηση φάρμας με 2 ζώα

    // Αρχικοποίηση 1ου ζώου (γουρούνι με όνομα Piggie 30.5 κιλών)
    farm.setAnimal(0, new Pig("Piggie",30.5));

    // Αρχικοποίηση 2ου ζώου (αγελάδα με όνομα Clara 315.10 κιλών)
    farm.setAnimal(1, new Cow("Clara",315.10));
    show(farm);      // Εκτυπώνει: Piggie (gourounaki), 30.5
                    // Clara (ageladitsa), 315.10
    farm.feed();     // Τάισμα ζώων της φάρμας - διαφορετική αύξηση βάρους/ζώο
    show(farm);      // Εκτυπώνει: Piggie (gourounaki), 30.8
                    // Clara (ageladitsa), 315.7
}

```

Άσκηση 8

Με δεδομένο τις παρακάτω κλάσεις να βρεθούν τα λάθη στο παρακάτω πρόγραμμα:

```

#include <iostream>
class BaseA {
    int iMemA;
public:
    void FunctionA(int );
};

class BaseB {
    int iMemB;
public:
    void FunctionB(int );
};

class Derived : public BaseA, public BaseB {
    int iMem;
};

void main() {
    Derived* pD = new Derived();

    BaseA* pA = pD;
    pA->FunctionA(1);
    pA->FunctionB(2);
}

```

```

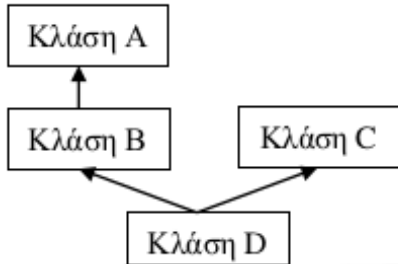
BaseB* pB = pD;
pB->FunctionA(2);
pB->FunctionB(1);

pD->FunctionA(4);
}

```

Άσκηση 9

Γράψτε κώδικα ο οποίος θα υλοποιεί το παρακάτω διάγραμμα κληρονομικότητας:



Σε κάθε κλάση να συμπεριλάβετε μια δημόσια συνάρτηση-μέλος η οποία θα εμφανίζει στην οθόνη το όνομα της κλάσης (στις κλάσεις A, B και D το όνομα της συνάρτησης θα πρέπει να είναι show ενώ στην κλάση C ονομάστε την αντίστοιχη συνάρτηση ως show_C). Απαντήστε στα παρακάτω ερωτήματα:

- Υποθέτουμε ότι δημιουργούμε ένα αντικείμενο της κλάσης D ως εξής: D obj; Η παρακάτω πρόταση obj.show(); ποια συνάρτηση θα καλέσει;
- Είναι δυνατό να καλέσουμε τη συνάρτηση-μέλος show() του στιγμιότυπου της κλάσης B καθώς και της κλάσης A του αντικειμένου obj;
- Θα υπήρχε καμία διαφορά στην περίπτωση που η συνάρτηση show() είχε δηλωθεί σαν εικονική σε κάποια από τις κλάσεις βάσης;

Άσκηση 10

Υλοποιήστε πρόγραμμα σε C++ που να αναπαριστά έναν καθηγητή πανεπιστημίου. Ένας καθηγητής πανεπιστημίου όπως όλοι οι άνθρωποι έχει ένα ονοματεπώνυμο. Όμως έχει και άλλες 2 βασικές ιδιότητες. Την ιδιότητα του διδάσκοντα και την ιδιότητα του ερευνητή. Για κάθε διδάσκοντα είναι χρήσιμο να καταχωρούνται τα μαθήματα που διδάσκει (το πολύ 2) και η βαθμίδα στην οποία ανήκει. Για έναν ερευνητή θα πρέπει να γνωρίζουμε το πεδίο έρευνας του και τον αριθμό των δημοσιεύσεων που έχει. Σχεδιάστε και υλοποιήστε την κατάλληλη ιεραρχία των κλάσεων. Σε κάθε κλάση υλοποιήστε μέθοδο που θα εμφανίζει τα στοιχεία της. Η αρχικοποίηση όλων των ιδιοτήτων θα πρέπει να γίνει με δομητές (constructors).