

Πανεπιστήμιο Δυτικής Μακεδονίας  
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών  
Αντικειμενοστραφής Προγραμματισμός Ι  
Εργαστήριο 2

Στο εργαστήριο θα ασχοληθούμε την δημιουργία κλάσεων και την χρήση αντικειμένων

Στόχοι εργαστηρίου:

- Δήλωση κλάσης
- Συναρτήσεις δόμησης
- Συναρτήσεις

Στο παράδειγμα που ακολουθεί θέλουμε να δηλώσουμε μια κλάση με το όνομα Person.

Η κλάση Person περιέχει δύο στοιχεία δεδομένων και δύο βασικές συναρτήσεις-μέλη. Αυτές οι δύο συναρτήσεις παρέχουν μοναδική πρόσβαση στα στοιχεία δεδομένων έξω από την κλάση. Η πρώτη συνάρτηση-μέλος – readData() - μας δίνει τη δυνατότητα να πληκτρολογήσουμε δύο τιμές και να τις αποδώσουμε άμεσα στα στοιχεία δεδομένων και η δεύτερη συνάρτηση-μέλος – printData() - εμφανίζει αυτές τις τιμές.

Η τοποθέτηση των δεδομένων και των συναρτήσεων μαζί, σε μία ενότητα, είναι η κεντρική ιδέα του αντικειμενοστραφούς προγραμματισμού.

Επιπρόσθετα, η κλάση περιέχει δύο συναρτήσεις-μέλη, που έχουν το ίδιο όνομα με την κλάση. Οι συναρτήσεις αυτές ονομάζονται συναρτήσεις εγκατάστασης, δόμησης (constructors) και έχουν σαν σκοπό να αρχικοποιούν τα δεδομένα ενός αντικειμένου όταν αυτό δηλώνεται. Όταν κατά τη δήλωση ενός αντικειμένου δε μεταβιβάζονται κάποιες τιμές, τότε εκτελείται ο constructor χωρίς ορίσματα, που μπορεί να μη κάνει τίποτα ή π.χ. να μηδενίσει τα δεδομένα. Όταν όμως μεταβιβάζονται τιμές στη δήλωση κάποιου αντικειμένου, τότε οι τιμές αυτές αποδίδονται αυτόματα στα δεδομένα-μέλη του αντικειμένου.

Για να δημιουργήσουμε το πρόγραμμα μας αρχικά δηλώνουμε την κλάση Person σε ένα header file. Αρχικά δημιουργείτε ένα φάκελο στον οποίο θα σώζετε όλα τα αρχεία σας (π.χ Lab2). Τα βήματα που ακολουθούμε είναι:

File->New Project->Console Application (Σώζετε στο φάκελο που δημιουργήσατε)

File->New Source File (Σώζετε σαν αρχείο επικεφαλίδας π.χ Person.h)

File->New Source File (Σώζετε σαν αρχείο c++ π.χ Person.cpp)

File->New Source File (Σώζετε σαν αρχείο c++ π.χ main.cpp)

Στο Person.h έχετε την δήλωση της κλάσης Person όπως φαίνεται στην εικόνα.

```
class Person
{
private:
    char name[30];
    int age;
public:
    Person();
    Person(char name1[], int age1);
    void readData();
    void printData();
}; // τέλος κλάσης
```

Στο Person.cpp έχετε τον ορισμό των συναρτήσεων της κλάσης Person όπως φαίνεται στην εικόνα.

```
#include <iostream>
#include "Person.h"
using namespace std;

Person::Person()
{
    strcpy(name, " ");
    age = 0;
}

Person::Person(char name1[], int age1)
{
    strcpy(name, name1);
    age = age1;
}

void Person::readData()
{
    cout << " Enter name:";
    cin >> name;
    cout << " Enter age:";
    cin >> age;
}

void Person::printData()
{
    cout << " The name of the person is " << name << endl;
    cout << " The age of the person is " << age << endl;
}
```

---

Στο main.cpp έχετε την κλήση της main όπως φαίνεται στην εικόνα.

```
#include <cstdlib>
#include <iostream>
#include "Person.h"
using namespace std;

int main(int argc, char *argv[])
{
    Person p1; // δήλωση αντικειμένου, χρήση constructor χωρίς ορίσματα
    Person p2("GEORGIU", 35); // δήλωση αντικειμένου, χρήση constructor με
    // ορίσματα
    cout << "Input data for object p1:" << endl;
    p1.readData(); // κλήση συνάρτησης-μέλους για ορισμό δεδομένων

    cout << "The data for object p1 are:" << endl;
    p1.printData(); // κλήση συνάρτησης μέλους για εμφάνιση δεδομένων
    cout << "The data for object p2 are:" << endl;
    p2.printData(); // κλήση
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

## Άσκηση Εργαστηρίου

Να αναπτυχθεί πρόγραμμα όπου θα δηλώνεται μία κλάση με το όνομα **vector** με δύο μέλη-δεδομένα (private) **int x**, **int y** και τις συναρτήσεις μέλη:

- **void readData()**
- **void setData(int xi, int yi)**
- **void printData(void)**

Στη συνέχεια:

- Να γίνει συνάρτηση εγκατάστασης (**constructor**) χωρίς ορίσματα.
- Να γίνει δεύτερη συνάρτηση εγκατάστασης (υπερφόρτωση) με ορίσματα (**int xi, int yi**).
- Να γίνει συνάρτηση μέλος **metro** χωρίς παραμέτρους και με επιστρεφόμενη τιμή τύπου **float**, η οποία να επιστρέφει το μήκος του διανύσματος

Στη **main** να δηλώσετε τρία αντικείμενα **vector** το **v1** το οποίο θα το διαβάσετε από το πληκτρολόγιο μέσω της συνάρτησης **readData()** και το **v2** το οποίο θα αρχικοποιήσετε μέσω του **constructor** με ορίσματα στις τιμές (1,4) και το **v3** το οποίο θα αρχικοποιήσετε μέσω της **setData(int xi, int yi)**, στις τιμές (5,9). Να υπολογίσετε και να εμφανίσετε στην οθόνη το μήκος των διανυσμάτων **v2** και **v3** καθώς και να τυπώσετε ένα μήνυμα που δείχνει πιο από τα 2 διανύσματα έχει μεγαλύτερο μήκος.

## Οδηγίες

- Πρέπει συνολικά να γίνουν όλες οι παραπάνω συναρτήσεις και το αποτέλεσμα υπολογισμού του μήκους να είναι σωστό. Για είσοδο (8,10) το αποτέλεσμα είναι 12,8.
- Η συνάρτηση για το μήκος του διανύσματος είναι η ακόλουθη:

$$| \overset{P}{A} | = \sqrt{x^2 + y^2}$$

- Πρέπει να χρησιμοποιήσετε τη συνάρτηση **sqrt** και να κάνετε **import** τη βιβλιοθήκη **cmath** (**#include <cmath>**). Για παράδειγμα **a= sqrt(y)**
- Θα πρέπει όλα τα αποτελέσματα της εκτέλεσης του προγράμματος είναι στοιχισμένα και να υπάρχουν τα κατάλληλα μηνύματα καθοδήγησης στον χρήστη.