

Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών
Αντικειμενοστραφής Προγραμματισμός Ι
Εργαστήριο 3

Στο εργαστήριο θα ασχοληθούμε την δημιουργία κλάσεων και την χρήση αντικειμένων

Στόχοι εργαστηρίου:

- Δήλωση κλάσης που έχει δεδομένο πίνακα
- Συναρτήσεις που δέχονται σαν όρισμα αντικείμενα
- Αντικείμενα σε πίνακες

Στο παράδειγμα που ακολουθεί θέλουμε να δηλώσουμε μια κλάση με το όνομα Runner (Δρομέας). Η κλάση Runner περιέχει δύο στοιχεία δεδομένων :

- `double time[5];` //πίνακας που περιέχει τους 5 καλύτερους χρόνους του δρομέα για μια συγκεκριμένη απόσταση
- `int distance;` // η απόσταση στην οποία αναφέρονται οι 5 καλύτεροι χρόνοι.

Η παραπάνω κλάση έχει και τις ακόλουθες συναρτήσεις:

- `void set_distance(int d);` // αρχικοποιεί τη μεταβλητή distance
- `void set_times(int i, double t);` // αρχικοποιεί τη θέση i του πίνακα time
- `void display();` // τυπώνει μήνυμα το οποίο αναφέρει τους 5 καλύτερους χρόνους του δρομέα για συγκεκριμένη απόσταση
- `double FindBest ();` // βρίσκει και επιστρέφει τον καλύτερο από τους 5 χρόνους
- `bool FindQuicker(Runner r);` // συγκρίνει το δρομέα του τρέχοντος αντικειμένου με τον δρομέα r (που παίρνει η συνάρτηση σαν όρισμα) και επιστρέφει true αν ο πρώτος δρομέας είναι γρηγορότερος από τον δεύτερο.

Στη main δηλώνουμε και αρχικοποιούμε 2 αντικείμενα Runner και μετά συγκρίνουμε τα 2 αντικείμενα και τυπώνουμε σχετικό μήνυμα για το ποιος από τους 2 δρομείς είναι γρηγορότερος.

Για να δημιουργήσουμε το πρόγραμμα μας αρχικά δηλώνουμε την κλάση Runner σε ένα header file. Αρχικά δημιουργείτε ένα φάκελο στον οποίο θα σώζετε όλα τα αρχεία σας (π.χ Lab3). 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 Runner.h)

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

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

Στο Runner.h έχετε την δήλωση της κλάσης Runner όπως φαίνεται.

```

class Runner
{

    private:

        double time[5]; //array of time in second
        int distance; //distance in meter

    public:
        void set_distance(int d);
        void set_times(int i, double t);
        void display();
        double FindBest();
        bool FindQuicker(Runner r);

};

```

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

```

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

void Runner::set_distance(int d)
{
    distance = d;
}

void Runner::set_times(int i, double t)
{
    time[i] = t;
}

void Runner::display()
{
    cout << "Here is your best 5 times for \n";
    cout << distance << " meter \n";

    for(int i = 0; i < 5; i++)
    {
        cout << time[i] << endl;
    }
}

double Runner::FindBest()
{
    double min=time[0];
    for (int i=1;i<5;i++)
        if (time[i]<min)

```

```

    min=time[i];
    return min;
}

bool Runner:: FindQuicker(Runner r)
{
    if (this->FindBest()>r.FindBest())
        return true;
    else
        return false;
}

```

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

```

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

int main(int argc, char** argv) {
    double n;
    int i;
    Runner r1,r2;
    r1.set_distance(6);
    r2.set_distance(6);
    for(i=0;i<5;i++)
    {
        cout<<"Give number for runner 1"<<i+1;
        cin>>n;
        r1.set_times(i,n);
    }

    for(i=0;i<5;i++)
    {
        cout<<"Give number for runner 2"<<i+1;
        cin>>n;
        r2.set_times(i,n);
    }
    if(r1.FindQuicker(r2))
        cout<<"Runner 1 is quicker";
    else
        cout<<"Runner 2 is quicker";

    return 0;
}

```

Άσκηση Εργαστηρίου

Να αναπτυχθεί πρόγραμμα όπου θα δηλώνεται μία κλάση με το όνομα **House** με δύο μέλη-δεδομένα (private)

int amount, το οποίο αντιπροσωπεύει τη χρηματική αξία ενός σπιτιού

double emvaddon[3], πίνακας που περιλαμβάνει το εμβαδόν καθενός από τα τρία δωμάτια τα οποία περιλαμβάνει το σπίτι.

και τις συναρτήσεις μέλη:

- **void readData()** διαβάζει από το πληκτρολόγιο τα δεδομένα ενός σπιτιού δηλαδή τις μεταβλητές amount και emvaddon[]
- **void printData()** τυπώνει τα στοιχεία ενός σπιτιού
- **double TotalArea()** υπολογίζει και επιστρέφει το συνολικό εμβαδόν ενός σπιτιού
- **bool findBigger(House h)** επιστρέφει true αν το σπίτι (αντικείμενο) που καλεί τη συνάρτηση είναι μεγαλύτερο από το αντικείμενο h που είναι όρισμα της συνάρτησης.
- **double findValueperMeter** υπολογίζει και επιστρέφει το κόστος που έχει το τετραγωνικό μέτρο του σπιτιού.

Στη συνέχεια:

- Να δηλώσετε και να αρχικοποιήσετε 2 αντικείμενα House.
- Να τυπώσετε τα στοιχεία τους και το εμβαδόν τους.
- Να γίνει σύγκριση του εμβαδού των 2 σπιτιών και να τυπωθεί πιο είναι μεγαλύτερο.
- Να γίνει σύγκριση του μέσου κόστους τετραγωνικού των 2 σπιτιών και να τυπωθεί πιο από τα 2 είναι μεγαλύτερη ευκαιρία καθώς και το κόστος του τετραγωνικού μέτρου των 2 σπιτιών.

Οδηγίες

- Πρέπει συνολικά να γίνουν όλες οι παραπάνω συναρτήσεις και τα αποτελέσματα των υπολογισμών να είναι σωστά.
- Θα πρέπει όλα τα αποτελέσματα της εκτέλεσης του προγράμματος είναι στοιχισμένα και να υπάρχουν τα κατάλληλα μηνύματα καθοδήγησης στον χρήστη.