

ΑΛΛΕΣ ΧΡΗΣΕΙΣ ΑΝΤΙΚΕΙΜΕΝΩΝ

Παράδειγμα 4: ΧΡΗΣΗ ΑΝΤΙΚΕΙΜΕΝΟΥ ΣΑΝ ΟΡΙΣΜΑ ΜΕΘΟΔΟΥ

Πρέπει να κατασκευάσουμε ένα πρόγραμμα στο οποίο θα **εισάγουμε** και θα **επεξεργαζόμαστε** πληροφορίες για τους αθλητές κατά την διάρκεια διεξαγωγής ενός αθλήματος. Η εισαγωγή θα γίνεται από τον **χρήστη** και οι πληροφορίες θα είναι το **όνομα** και η **επίδοση** κάθε αθλητή. Η εισαγωγή των πληροφοριών για τον κάθε αθλητή θα **επαναλαμβάνεται** έως ότου ο χρήστης επιλέξει να μην συνεχίσει. Το πρόγραμμα θα **συγκρίνει** τις επιδόσεις των αθλητών και θα **τυπώνει** στην οθόνη τα στοιχεία του αθλητή με την **καλύτερη επίδοση**.

Μέσα στην κλάση κατασκευάζουμε την μέθοδο `is_Better_Than()` η οποία δέχεται ως όρισμα ένα αντικείμενο της κλάσης και συγκρίνει το σκορ του αντικειμένου που την καλεί με το αντικείμενο που δέχτηκε ως όρισμα.

Κλάση

```
//Class Athletes

#include<iostream>
#include<string.h>
using namespace std;

class Athletes{ //Δήλωση κλάσης

private:
    string name;
    float score;

public:
    //Constructor χωρίς όρισμα
    Athletes(){
        name = "+++";
        score = 0;
    }

    /* Μέθοδος εισαγωγής τιμών
    με την μέθοδο της αληλεπίδρασης */
    void readData(){
        cout << "Give Athlete's name: ";
        cin >> name;
        cout << "Give Athlete's score (in meters): ";
        cin >> score;
    }

    //Μέθοδος εκτύπωσης τιμών
    void printData(){
        cout << "\nThe Best Athlete is: " << name << endl;
        cout << "With Score " << score << " Meters" << endl;
    }

    //Μέθοδος σύγκρισης τιμών των αντικειμένων
    bool is_Better_Than(Athletes other){ //Αντικείμενο ως όρισμα της μεθόδου
        if(score > other.score) //Σύγκριση τιμών
            return true;
        else
            return false;
    }
};
```

Κυρίως πρόγραμμα

```
//Main programm
//Includes Athletes.hpp

#include<iostream>
#include"Athletes.hpp" //Συμπεριλαμβάνεται η κλάση Athletes
#include <string.h>
using namespace std;

int main(){

    /* Δημιουργία του αντικειμένου
    καλύτερος αθλητής */
    Athletes bestAthlete;

    bool next = true;
    while(next){
        /* Δημιουργία του αντικειμένου
        νέος αθλητής */
        Athletes newAthlete;

        /* Συμπλήρωση στοιχείων
        του νέου αθλητή */
        newAthlete.readData();

        /* Έλεγχος αν το σκορ του νέου αθλητή
        είναι καλύτερο από του αρχικού καλύτερου αθλητή */
        if(newAthlete.is_Better_Than(bestAthlete)){

            //Τότε ο νέος είναι ο καλύτερος
            bestAthlete = newAthlete;
        }

        cout << "Continue with other Athlete? [1->Yes, 2->No]: ";
        int choice;
        cin >> choice;
        if (choice != 1)
            next = false;
    }

    /* Εκτύπωση στοιχείων
    του καλύτερου αθλητή */
    bestAthlete.printData();

    return 0;
}
```

Παράδειγμα 2: ΕΠΙΣΤΡΟΦΗ ΑΝΤΙΚΕΙΜΕΝΟΥ ΑΠΟ ΜΕΘΟΔΟ

Έχουμε τη δυνατότητα να επιστρέψουμε αντικείμενο με την χρήση μεθόδου με τύπο επιστροφής την ίδια την κλάση. Στο παράδειγμα που ακολουθεί εφοδιάζουμε την κλάση με την μέθοδο `addBalance()` η οποία δέχεται ένα αντικείμενο ως όρισμα και μπορεί να επιστρέψει ένα τοπικό αντικείμενο (*result*) το οποίο κατασκευάζεται μέσα στην μέθοδο. Το τοπικό αντικείμενο (*result*) έχει το αποτέλεσμα της άθροισης του υπολοίπου του αντικειμένου που καλεί την μέθοδο (*Acc1*) συν το υπόλοιπο του αντικειμένου το οποίο στέλνεται ως όρισμα στην μέθοδο (*Acc2*). Το επιστρεφόμενο αντικείμενο (*result*) αποδίδεται ως τιμή στο αντικείμενο *Acc3*. Άρα το αντικείμενο *Acc3* έχει υπόλοιπο το άθροισμα των υπολοίπων των άλλων δύο αντικειμένων. (*Acc1*+ *Acc2*).

Κλάση

```
//Κλάση Accounts.hpp

#include <iostream>
using namespace std;
class Accounts{ //Δήλωση κλάσης
private:
    float balance; //Δήλωση δεδομένου μέλους
public:
    //Δήλωση constructor χωρίς ορίσματα
    Accounts(){
        balance = 0;
    }

    //Δήλωση constructor με όρισμα
    Accounts(float aBalance){
        balance = aBalance;
    }

    //Δήλωση destructor
    ~Accounts(){
    }

    //Ορισμός μεθόδου ανάληψης
    void withdrawal(float amount){
        if (amount <= balance){ //Έλεγχος υπέρβασης υπολοίπου
            balance -= amount; //Ενημέρωση υπολοίπου
            cout << "Withdrawal " << amount << "Euros Done!!" << endl;
        }
        else{ //Μήνυμα αδυναμίας ανάληψης ποσού
            cout<<"ATTENTION!!!!!!"<<endl;
            cout<<"THE AMOUNT IS GREATER THAN THE BALANCE OF THE ACCOUNT"<<endl;
        }
    }

    //Ορισμός μεθόδου κατάθεσης
    void deposit(float amount){
        cout << "Deposit: " << amount << " Euros" << endl;
        balance += amount; //Ενημέρωση υπολοίπου
    }

    //Ορισμός μεθόδου επιστροφής υπολοίπου
    float returnBalance(){
        return balance; //Επιστροφή υπολοίπου
    }

    /* Μέθοδος τύπου Accounts που δέχεται αντικείμενο ως όρισμα
    και επιστρέφει αντικείμενο */
    Accounts addBalance(Accounts anAcc)
```

```

        {
            Accounts result; // τοπικό αντικείμενο
            result.balance = balance + anAcc.balance;
            return result; // επιστροφή αντικειμένου
        }
}; //Τέλος κλάσης

```

Κυρίως πρόγραμμα

```

//Κυρίως πρόγραμμα
//Banking_transactions.cpp

#include<iostream>
#include"Accounts.hpp" //Συμπερίληψη της κλάσης Accounts.hpp

int main(){

    //Κατασκευή αντικειμένων
    Accounts Acc1(200), Acc2(500); //Λειτουργεί ο constructor με όρισμα
    Accounts Acc3; //Λειτουργεί ο constructor χωρίς όρισμα

    cout << "\n===NEW BALANCE===" << endl; //Νέα υπόλοιπα
    cout << "Acc1 Balance is: " << Acc1.returnBalance() << " Euros" << endl;
    cout << "Acc2 Balance is: " << Acc2.returnBalance() << " Euros" << endl;
    cout << "Acc3 Balance is: " << Acc3.returnBalance() << " Euros" << endl;

    /* Κλήση μεθόδου addBalance από το αντικείμενο Acc1
    με ταυτόχρονη αποστολή σε αυτήν του αντικειμένου Acc2
    Το αποτέλεσμα εκχωρείται απευθείας στο αντικείμενο Acc3*/
    cout << "\nSum of the balances of Acc1 + Acc2" << endl;
    Acc3 = Acc1.addBalance(Acc2);

    cout << "\n===NEW BALANCE===" << endl; //Νέα υπόλοιπα
    cout << "Acc1 Balance is: " << Acc1.returnBalance() << " Euros" << endl;
    cout << "Acc2 Balance is: " << Acc2.returnBalance() << " Euros" << endl;
    cout << "Acc3 Balance is: " << Acc3.returnBalance() << " Euros" << endl;

    cout << endl;
    return 0;
}

```

Παράδειγμα 3: ΑΠΟΔΟΣΗ ΤΙΜΩΝ ΣΤΗΝ ΚΛΑΣΗ

Σε αυτό το πρόγραμμα εισάγουμε 2 αριθμούς και υπολογίζουμε το άθροισμα των τετραγώνων τους και την τετραγωνική ρίζα του αθροίσματος, καθώς και το άθροισμα των κύβων τους και την κυβική ρίζα του αθροίσματος.

```
#include <iostream>
#include <math.h>
class paradeigma
{
private:
    int numa,numb,sqareSum,cubeSum;
public:
    void getData(int _numa, int _numb) //metafora timon sthn klash
    {
        numa =_numa;
        numb =_numb;
    }
    int calcDataSqr() //athroisma ton tetragwnwn
    {
        sqareSum = pow(numa,2)+pow(numb,2);
        return sqareSum;
    }
    int calcDataCube() //athroisma ton kyvwn
    {
        cubeSum = pow(numa,3)+pow(numb,3);
        return cubeSum;
    }
    void printSqrtData() //ektyposh athroismatos twn tetragwnwn
    {
        cout<<" 1os arithmos: "<< numa <<endl;
        cout<<" 2os arithmos: "<< numb <<endl;
        cout<<"Athroisma Tetragonon: "<< sqareSum <<endl;

    }
    void printCubeData() //ektyposh athroismatos twn kyvwn
    {
        cout<<" 1os arithmos: "<< numa <<endl;
        cout<<" 2os arithmos: "<< numb <<endl;
        cout<<"Athroisma Kyvon: "<< cubeSum <<endl;
    }
}; //telos klashs
void main()
{
    paradeigma par1, par2; //antikeimena ths klashs
    int za, zb, apotSqrt, apotcube; //topikes metavlhtes int
    float rizaSqrt, rizaCbrt; //topikes metavlhtes float

    while(1) //synexomenh epanalhpsi (1>0)
    {
        cout<<"dose 2 arithmous"<<endl;
        cout<<"h dose 2 fores 0 gia exodo"<<endl;
        cin>>za;
        cin>>zb;
        if((za==0)&&(zb==0)) //exodus an kai oi 2 arithmoi einai 0
        {
            cout<<"exodos apo to programma"<<endl;
            break;
        }

        cout<<"*****"<<endl;

        par1.getData(za,zb); //apostolh timon sthn klash
        apotSqrt=par1.calcDataSqr();
        par1.printSqrtData(); //ektyposh dedomenon
```

```

cout<<"*****"<<endl;
cout<<"to athroisma ton tetragonon einai: "<<apotSqrt<<endl;
cout<<"*****"<<endl;

//ypologismos tetragonikh rizas
rizaSqrt=sqrt(apotSqrt);
cout<<"H tetragonikh riza toy athroismatos einai: "<<endl;
cout<< rizaSqrt<<endl;
cout<<"*****"<<endl;

par2.getData(za,zb);
apotcube=par2.calcDataCube();
par2.printCubeData();
cout<<"*****"<<endl;
cout<<"to athroisma ton kyvon einai: "<<apotcube<<endl;

//ypologismos kyvikh rizas me casting ston ek8eth
rizaCbrt=pow(apotcube, (float)1/3);
cout<<"H kyvikh riza toy athroismatos einai: "<<endl;
cout<<rizaCbrt<<endl;
}
} //telos main

```