

ΑΡΧΕΙΑ

- Εκτελέστε τα παρακάτω προγράμματα και δείτε τις εξόδους που παρουσιάζουν.
- Κάντε αλλαγές όποιες εσείς νομίζετε και εκτελέστε εκ νέου.
- Συζητήστε τα αποτελέσματα των αλλαγών με την ομάδα σας.

Παράδειγμα 1: Αποθήκευση σε αρχείο δεδομένων μιας κλάσης σε δυαδική μορφή.

Κατασκευή προγράμματος για την καταγραφή πόντων σε αγώνα. Χρήση μιας κλάσης. Αποστολή δεδομένων σε αρχείο σε δυαδική μορφή.

```
#include<iostream>
#include<fstream>
using namespace std;
class Score{
private:
    int onePoint, twoPoints, threePoints;
public:
    Score(){
        onePoint = 0;
        twoPoints = 0;
        threePoints = 0;
    }
    Score(int one, int two, int three){
        onePoint = one;
        twoPoints = two;
        threePoints = three;
    }
    ~Score(){
    }
    void viewPoints(){
        cout << "You scored....." << endl;
        cout << onePoint << " (1p)" << " = " << onePoint << " Pts" << endl;
        cout << twoPoints << " (2p)" << " = " << twoPoints * 2 << " Pts" << endl;
        cout << threePoints << " (3p)" << " = " << threePoints * 3 << " Pts" << endl;
        cout << "\nTotal Score: " << onePoint+(twoPoints*2)+(threePoints*3) << " Pts" << endl;
    }
};

int main()
{
    ofstream outPoints; //αντικείμενό για την εγγραφή των δεδομένων
    ifstream inPoints; //αντικείμενό για την ανάγνωση των δεδομένων
    Score Sin, Sout(10,12,9); //αντικείμενα της κλάσης Score

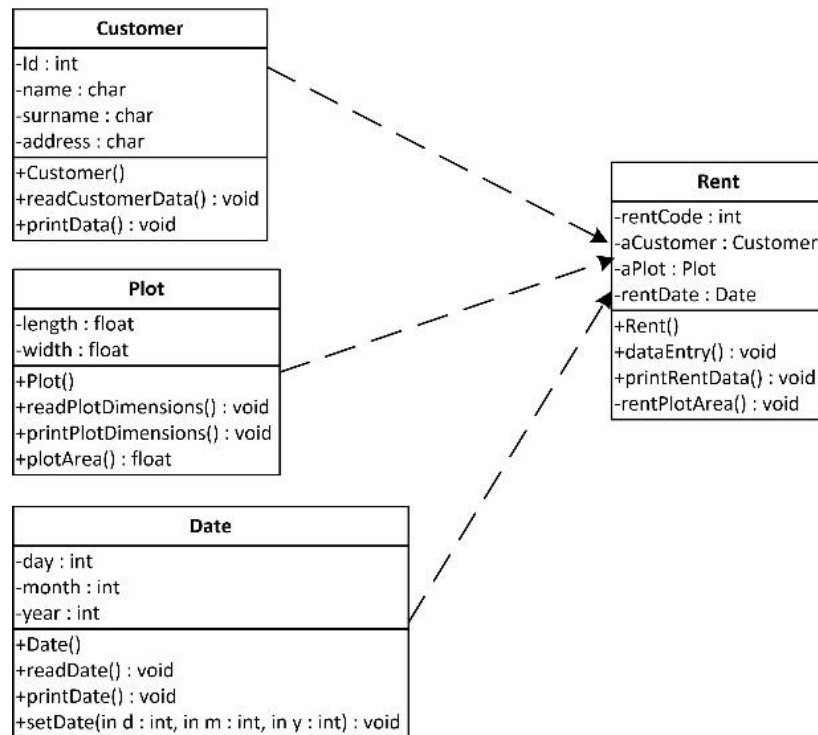
    outPoints.open("GamePoints.txt", ios::out|ios::binary); //Ανοιγμα ροής εξόδου
    outPoints.write((char*) &Sout, sizeof(Sout)); //εγγραφή δεδομένων σε δυαδική μορφή
    outPoints.close(); //κλείσιμο ροής εξόδου

    inPoints.open("GamePoints.txt", ios::in|ios::binary); //Ανοιγμα ροής εισόδου
    inPoints.read((char*) &Sin, sizeof(Sin)); //ανάγνωση δεδομένων από αρχείο
    Sin.viewPoints(); // εκτύπωση δεδομένων στην οθόνη
    inPoints.close(); // κλείσιμο ροής εισόδου

    return 0;
}
```

Παράδειγμα 2: Αποθήκευση σε αρχείο δεδομένων συσχετιζόμενων κλάσεων

Κατασκευή προγράμματος για την ενοικίαση οικοπέδων. Πολλαπλή συσχέτιση κλάσεων. Αποστολή δεδομένων σε αρχείο σε δυαδική μορφή.



Customer.hpp

```
// class Customer.h
#include<string.h>
class Customer{
private:
    int ID;
    char name[10];
    char surname[30];
    char address[20];
public:
    Customer(){
    }
    Customer(int anID, char aName[], char aSname[], char anAdrs[]){
        ID = anID;
        strcpy(name, aName);
        strcpy(surname, aSname);
        strcpy(address, anAdrs);
    }
    ~Customer(){
    }
    void readCustomerData(){
        cout << "Insert ID: "; cin >> ID;
        cout << "Insert Name: "; cin >> name;
        cout << "Insert Surname: "; cin >> surname;
        cout << "Insert Address: "; cin >> address;
    }
    void printData(){
```

```

        cout << "Customer Informations" << endl;
        cout << "The Id is: " << ID << endl;
        cout << "The Name is: " << name << endl;
        cout << "The Surname is: " << surname << endl;
        cout << "The Address is: " << address << endl;
    }
};

```

Plot.hpp

```

//class Plot.h
class Plot{
    private:
        float length, width;
    public:
        Plot(){
            length = 0;
            width = 0;
        }
        Plot(float l, float w){
            length = l;
            width = w;
        }
        ~Plot(){
        }
        void readPlotDimensions(){
            cout << "Insert length in (m): "; cin >> length;
            cout << "Insert width in (m): "; cin >> width;
        }
        void printPlotDimensions(){
            cout << "Plot Dimensions" << endl;
            cout << "Plot Length: " << length << "m" << endl;
            cout << "Plot Width: " << width << "m" << endl;
        }
        float plotArea(){
            return length * width;
        }
};

```

Date.hpp

```

//class Date.h
class Date{
    private:
        int day, month, year;
    public:
        Date(){
            day = 0;
            month = 0;
            year = 0;
        }
        Date(int _d, int _m, int _y){
            day = _d;
            month = _m;
            year = _y;
        }
        void readDate(){
            cout << "Insert Day: "; cin >> day;
            cout << "Insert Month: "; cin >> month;
            cout << "Insert Year: "; cin >> year;
        }
};

```

```

        void setDate(int d, int m, int y){
            day = d;
            month = m;
            year = y;
        }
        void printDate(){
            cout << day << "/" << month << "/" << year << endl;
        }
};

```

Rent.hpp

```

//class Rent.h
#include "Plot.hpp"
#include "Customer.hpp"
#include "Date.hpp"
class Rent{
    private:
        int rentCode;
        Customer aCustomer; //αντικείμενο συσχετιζόμενης κλάσης
        Plot aPlot; //αντικείμενο συσχετιζόμενης κλάσης
        Date rentDate; //αντικείμενο συσχετιζόμενης κλάσης
        void rentPlotArea(){ //ιδιωτικά ορατή βοηθητική μέθοδος
            cout << "Plot Area: " << aPlot.plotArea() << "m2" << endl;
        }
    public:
        void dataEntry(){
            cout << "\nInsert Rent Code: "; cin >> rentCode;
            cout << "\nCustomer Informations" << endl;
            aCustomer.readCustomerData();
            cout << "\nPlot Informations" << endl;
            aPlot.readPlotDimensions();
            cout << "\nRent Date" << endl;
            rentDate.readDate();
        }
        void printRentData(){
            cout << "Rent Code: " << rentCode << endl;
            aCustomer.printData();
            aPlot.printPlotDimensions();
            cout << "The Rent Date is...." << endl;
            rentDate.printDate();
            rentPlotArea();
        }
};

```

RentAPlot.cpp

```

#include<iostream>
#include<fstream>
#include"Rent.hpp"
using namespace std;
int main()
{
    ofstream outFile;
    ifstream inFile;
    Rent R1, R2;
    int morePlots;

    //open output stream to a file
    outFile.open("PlotRent.txt", ios::app|ios::binary);
    cout << "\nStart Rent Data Entry" << endl;

```

```
do{
    R1.dataEntry();
    outFile.write((char*) &R1, sizeof(R1));
    cout << "There are other plots for rent ?" << endl;
    cout << "1->Yes, 2->No: "; cin >> morePlots;
}while(morePlots == 1);
outFile.close();

//open input stream from a file
cout << "\nPrint contents of PlotRent.txt\n" << endl;
inFile.open("PlotRent.txt", ios::in|ios::binary);
inFile.read((char*) &R2, sizeof(R2));
do{
    R2.printRentData();
    cout << endl;
    inFile.read((char*) &R2, sizeof(R2));
}while(!inFile.eof());
inFile.close();
return 0;
}
```