

ΚΛΑΣΕΙΣ - ΑΝΤΙΚΕΙΜΕΝΑ

ΟΡΙΣΜΟΣ

Κλάση είναι μια αφηρημένη δομή δεδομένων, μία συλλογή από οντότητες συναφείς μεταξύ τους.

Οι οντότητες ονομάζονται **αντικείμενα** ή στιγμιότυπα της κλάσης.

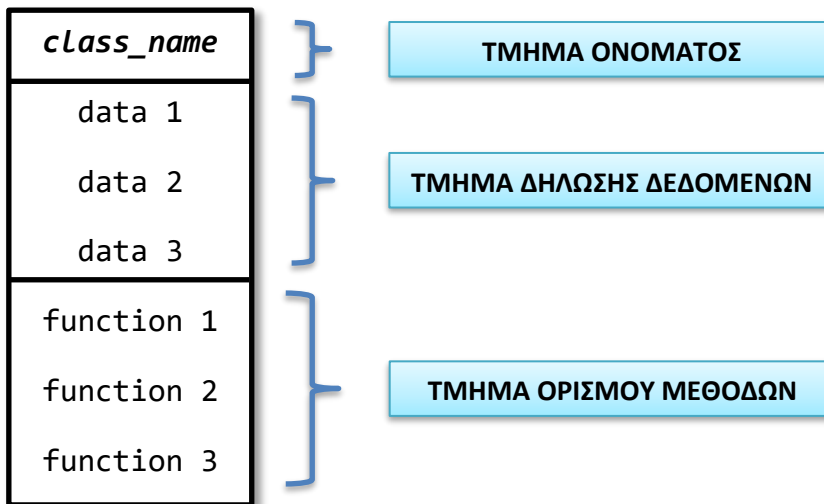
Μια κλάση είναι ένα σύνολο από:

- **δεδομένα** που δηλώνουν τα χαρακτηριστικά της κλάσης και κατ' επέκταση των αντικειμένων της (τι είναι) και
- **μεθόδους** (συναρτήσεις) που προδιαγράφουν την συμπεριφορά της κλάσης (τι κάνει).

Π.χ. μια κλάση μπορεί να είναι οι «εργαζόμενοι» με χαρακτηριστικά: Α.Μ, Όνομα, Επώνυμο, ΑΦΜ, Μισθός, ή ακόμη και η «εγγραφή» με χαρακτηριστικά: Κωδικός, Ημερομηνία, Φοιτητής, Μάθημα.

ΣΧΗΜΑΤΙΚΗ ΠΑΡΑΣΤΑΣΗ ΚΛΑΣΗΣ

Μία κλάση παριστάνεται σχηματικά στην γλώσσα UML (*Unified Modeling Language - Ενοποιημένη Γλώσσα Σχεδίασης Προτύπων*) με ένα παραλληλόγραμμο χωρισμένο σε τρία τμήματα



ΔΗΛΩΣΗ ΚΛΑΣΗΣ ΣΤΗΝ C++

Η δήλωση μιας κλάσης στην C++ γίνεται έξω από την main(). Αρχίζει με τη δεσμευμένη λέξη **class** και το όνομα της κλάσης, ακολουθεί **αριστερό άγκιστρο**, κατόπιν ακολουθούν οι δεσμευμένες λέξεις **public** ή **private** (θα αναφερθούμε σε αυτές πιο κάτω) και το **κυρίως σώμα** της κλάσης που περιέχει τα **δεδομένα - μέλη** και τις **μεθόδους ή συναρτήσεις - μέλη**. Στο τέλος η κλάση κλείνει με το **δεξί άγκιστρο** και το **τερματικό εντολής** (ελληνικό ερωτηματικό).

ΠΑΡΑΔΕΙΓΜΑ ΔΗΛΩΣΗΣ ΤΗΣ ΚΛΑΣΗΣ CAR

```
class Car { //Όνομα κλάσης

    public: //Προσδιοριστικό ορατότητας
        int numOfDoors; //Δεδομένο-μέλος
        int numOfWheels;

    private: //Προσδιοριστικό ορατότητας
        Start(); //Συνάρτηση-μέλος ή Μέθοδος κλάσης
        Run();
        Break();

};
```

ΠΡΟΣΒΑΣΗ ΣΤΑ ΔΕΔΟΜΕΝΑ ΤΗΣ ΚΛΑΣΗΣ

Στη δήλωση μιας κλάσης χρησιμοποιούμε τις λέξεις **public** (δημόσιο) ή **private** (ιδιωτικό) για να δηλώσουμε τον τρόπο πρόσβασης των μελών της κλάσης. Τα μέλη της κλάσης μπορεί να είναι **δημόσια ή ιδιωτικά προσπελάσιμα** (ορατά).

Μέλη ιδιωτικά ορατά (-) είναι αυτά που δεν μπορεί να τα δει κανείς. Αν πρόκειται για δεδομένα - μέλη είναι ορατά μόνο από τις συναρτήσεις της ίδιας της κλάσης στην οποία ανήκουν.

Μέλη δημόσια ορατά (+) είναι αυτά που μπορούν να τα δουν όλοι. Συνήθως δηλώνονται οι συναρτήσεις - μέλη της κλάσης.

Για να προστατέψουμε τα δεδομένα των αντικειμένων από αλλοιώσεις τα αποκρύπτουμε από πελάτες της κλάσης δηλαδή τα δηλώνουμε private. Ενώ τις μεθόδους στις οποίες πρέπει να έχουν πρόσβαση οι πελάτες της κλάσης τις δηλώνουμε public.

Εξ ορισμού τα δεδομένα και οι μέθοδοι είναι ιδιωτικά ορατά. Αν δεν χρησιμοποιήσουμε τις δηλώσεις public ή private, τότε ούτε τα δεδομένα θα είναι προσπελάσιμα αλλά ούτε και οι μέθοδοι θα μπορούν να χρησιμοποιηθούν από τα αντικείμενα της κλάσης.

Διαφορά μεταξύ της κλάσης στον Αντικειμενοστραφή και της δομής στον Διαδικαστικό προγραμματισμό είναι ότι στη δομή δεν μπορούμε να κρύψουμε τα δεδομένα όπως στην κλάση. Αυτό έχει σαν αποτέλεσμα τα δεδομένα να είναι προσβάσιμα σε όλους με πιθανή αλλοίωση των τιμών τους από εξωτερικούς παράγοντες.

ΣΧΗΜΑΤΙΚΗ ΠΑΡΑΣΤΑΣΗ ΠΡΟΣΒΑΣΙΜΟΤΗΤΑΣ ΜΙΑΣ ΚΛΑΣΗΣ

class
-data 1 -data 2 -data 3
+method 1() +method 2() +method 3()

ΟΡΙΣΜΟΣ ΕΝΟΣ ΑΝΤΙΚΕΙΜΕΝΟΥ ΤΗΣ ΚΛΑΣΗΣ

Για να ορίσουμε ένα **αντικείμενο** μιας κλάσης στο πρόγραμμα κάνουμε την παρακάτω δήλωση:

```
όνομα_κλάσης όνομα_αντικειμένου;
```

Για να ορίσουμε ένα αντικείμενο **obj1** της κλάσης **myClass** γράφουμε την πιο κάτω δήλωση:

```
myClass obj1;
```

Σημ. Ο ορισμός αντικειμένου γίνεται έξω από την ίδια την κλάση (μέσα στην *main* ή μέσα σε άλλη κλάση).

ΠΡΟΣΒΑΣΗ ΕΝΟΣ ΑΝΤΙΚΕΙΜΕΝΟΥ ΣΤΑ ΔΕΔΟΜΕΝΑ ΚΑΙ ΤΙΣ ΜΕΘΟΔΟΥΣ ΤΗΣ ΚΛΑΣΗΣ

Για να έχει πρόσβαση το αντικείμενο που φτιάξαμε στα δεδομένα και τις συναρτήσεις της κλάσης, χρησιμοποιούμε τον τελεστή τελεία (.) που είναι ο **τελεστής κλήσης**.

Για την πιο πάνω κλάση με την δήλωση: **obj1.data_1 = 10;** το δεδομένο μέλος *data_1* του αντικειμένου *obj1* της κλάσης *class*, παίρνει την τιμή 10.

Για να καλέσει το αντικείμενο μία από τις μεθόδους της κλάσης χρησιμοποιούμε πάλι τον τελεστή κλήσης. Με την δήλωση: **obj1.method_3();** το αντικείμενο *obj1* καλεί την μέθοδο *method_3()* της κλάσης *class*.

=====

ΠΑΡΑΔΕΙΓΜΑΤΑ ΔΗΛΩΣΗΣ ΚΛΑΣΕΩΝ

ΟΡΙΣΜΟΣ ΜΕΘΟΔΩΝ ΜΕΣΑ ΣΤΗΝ ΚΛΑΣΗ

Παράδειγμα 1: ΑΠΟΔΟΣΗ ΤΙΜΩΝ ΜΕ ΜΕΘΟΔΟ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ – ΕΚΤΥΠΩΣΗ ΤΙΜΩΝ

Στο παρακάτω πρόγραμμα δηλώνεται μία κλάση εργαζόμενος η οποία έχει ιδιωτικά ορατά τα δεδομένα: *κωδικό, όνομα, ημερομίσθιο, μισθό* και δημόσια ορατές τις μεθόδους *readData()* που είναι μέθοδος αλληλεπίδρασης με τον χρήστη για την απόδοση τιμών στα αντικείμενα και *printData()* που είναι μέθοδος εκτύπωσης των τιμών των αντικειμένων. Κατόπιν μέσα στην *main()* κατασκευάζονται δύο αντικείμενα της κλάσης τα οποία χειρίζονται (καλούν) τις μεθόδους της κλάσης για να εισαχθούν και να τυπωθούν οι τιμές των δεδομένων των αντικειμένων.

```
#include <iostream>
using namespace std;
class Employee{ //Δήλωση της κλάσης
    private: //χώρος ιδιωτικής πρόσβασης
        int code;
        char name[30];
        double payrate;
        double salary;

    public: //χώρος δημόσιας πρόσβασης
        void readData(){ //Ορισμός μεθόδου αλληλεπίδρασης για την συλλογή τιμών
            cout << "Give Code:"; cin >> code;
            cout << "Give Name:"; cin >> name;
            cout << "Give Payrate:"; cin >> payrate;
            salary = payrate * 8;
        }
        void printData(){ //Ορισμός μεθόδου εκτύπωσης τιμών
            cout << "Employee Informations" << endl;
            cout << "Employee Code: " << code << endl;
            cout << "Employee Name: " << name << endl;
            cout << "Employee Salary: " << salary << endl;
        }
}; //Τέλος κλάσης

int main(){
    Employee e1, e2; //Δήλωση των αντικειμένων e1 και e2 της κλάσης Employee

    cout << "=====  
DATA COLLECTION BEGINS =====" << endl;
    e1.readData(); //Χειρισμός μεθόδου από το αντικείμενο e1 για εισαγωγή δεδομένων
    e2.readData();
    cout << endl;
    cout << "=====  
PRINTING INFORMATION =====" << endl;
    e1.printData(); //Χειρισμός μεθόδου από αντικείμενο e1 για εκτύπωση δεδομένων
    e2.printData();

    return 0;
}
```

Παράδειγμα 2: ΟΡΙΣΜΟΣ ΚΡΥΦΗΣ ΜΕΘΟΔΟΥ

Στην περίπτωση που θέλουμε να υπολογίζουμε τον μισθό μετά από 8 ώρες εργασίας φτιάχνουμε την εσωτερική (κρυφή) μέθοδο *salaryCalculate(double prt)* που δεν χρειάζεται να την δει κανείς έξω από την κλάση γι' αυτό την κρύβουμε την ορίζουμε δηλαδή στο χώρο ιδιωτικής πρόσβασης (*private*). Την καλούμε στο σημείο που θα πάρει τιμή το δεδομένο *salary*. Το υπόλοιπο πρόγραμμα παραμένει ίδιο.

```
#include <iostream>
using namespace std;
class Employee{ //Δήλωση της κλάσης
private:
    int code;
    char name[30];
    double payrate;
    double salary;
    double salaryCalculate(double prt){ //Ορισμός κρυφής μεθόδου υπολογισμού μισθού
        return 8 * prt;
    }

public:
    void readData(){ //Ορισμός μεθόδου αλληλεπίδρασης για την συλλογή τιμών
        cout << "Give Code:"; cin >> code;
        cout << "Give Name:"; cin >> name;
        cout << "Give Payrate:"; cin >> payrate;
        salary = salaryCalculate(payrate);
    }
    void printData(){ //Ορισμός μεθόδου εκτύπωσης τιμών
        cout << "Employee Informations" << endl;
        cout << "Employee Code: "<< code<< endl;
        cout << "Employee Name: " << name << endl;
        cout << "Employee Salary: " << salary << endl;
    }
}; //Τέλος κλάσης

int main(){
    Employee e1; //Δήλωση αντικειμένου

    cout << "=====  
DATA COLLECTION BEGINS =====< << endl;
    e1.readData();
    cout << endl;
    cout << "=====  
PRINTING INFORMATION =====< << endl;
    e1.printData();

    return 0;
}
```

Παράδειγμα 3: ΜΕΘΟΔΟΣ ΑΠΟΔΟΣΗΣ ΤΙΜΩΝ ΜΕΣΩ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΗΣ (ΜΕΘΟΔΟΣ SETTER)

Στο πιο κάτω πρόγραμμα ορίζουμε μία μέθοδο **setData()**, η οποία όταν κληθεί από κάποιο αντικείμενο στην main(), μεταβιβάζονται οι τιμές 002, Papadopoulos και 10 στις αντίστοιχες **aCode**, **aName** και **aPayrate**. Έτσι όταν τρέξει το πρόγραμμα το αντικείμενο θα πάρει τιμές. Τα ορίσματα στην μέθοδο setData(), δηλώνονται γιατί το αντικείμενο μέσα από την main() δεν μπορεί να έχει πρόσβαση στα δεδομένα της κλάσης επειδή αυτά έχουν δηλωθεί ως private.

Σημ. Χρησιμοποιούμε το αρχείο κεφαλίδας (βιβλιοθήκη ή κλάση) **string.h** το οποίο περιλαμβάνει την συνάρτηση **strcpy** η οποία μας βοηθά να μεταφέρουμε τις τιμές του πίνακα χαρακτήρων aName στον πίνακα χαρακτήρων name.

```
#include<iostream>
#include<string.h> //συμπεριλαμβάνεται η βιβλιοθήκη string.h
using namespace std;
class Employee{ //Δήλωση της κλάσης
private:
    int code;
    char name[30];
    double payrate;
    double salary;

public:
    //μέθοδος απόδοσης τιμών μέσω παραμέτρων
    void setData(int aCode, char aName[], double aPayrate){
        code= aCode;
        strcpy(name, aName);
        payrate = aPayrate;
        salary = payrate*8;
    }
    void printData(){
        cout << "Employee Informations" << endl;
        cout << "Employee Code: " << code << endl;
        cout << "Employee Name: " << name << endl;
        cout << "Employee Salary: " << salary << endl;
    }
}; //Τέλος κλάσης

int main(){
    Employee e1;
    e1.setData(2, "Papadopoulos", 10); //απόδοση τιμών στο αντικείμενο
    e1.printData();

    return 0;
}
```

Παράδειγμα 4: ΜΕΘΟΔΟΣ ΕΠΙΣΤΡΟΦΗΣ ΤΙΜΗΣ (ΜΕΘΟΔΟΣ GETTER)

Στο πιο κάτω πρόγραμμα ορίζουμε δύο μεθόδους επιστροφής τιμής **returnSalary()** και **returnCode()** οι οποίες όταν κληθούν από το αντικείμενο μέσα στη main() επιστρέφουν τον κωδικό και το μισθό του (μόνο τιμές).

```
#include<iostream>
#include<string.h>
using namespace std;
class Employee{ //Δήλωση της κλάσης
private:
    int code;
    char name[30];
    double payrate;
    double salary;

public:
    void setData(int aCode, char aName[], double aPayrate){
        code = aCode;
        strcpy(name, aName);
        payrate = aPayrate;
        salary = payrate*8;
    }
    void printName(){
        cout << "Employee's Name is: " << name << endl;
    }
    double returnSalary(){ //μέθοδος επιστροφής τιμής
        return salary;
    }
    int returnCode(){ //μέθοδος επιστροφής τιμής
        return code;
    }
}; //Τέλος κλάσης

int main(){
    Employee e1;
    e1.setData(2, "Papadopoulos", 10); //απόδοση τιμών στο αντικείμενο
    e1.printName();
    cout << "Employee's Code is: "<< e1.returnCode() << endl;
    cout << "Employee's Salary is: "<< e1.returnSalary()<< endl;

    return 0;
}
```

ΟΡΙΣΜΟΣ ΜΕΘΟΔΩΝ ΕΞΩ ΑΠΟ ΤΗΝ ΚΛΑΣΗ

Παράδειγμα 5: Στο πιο κάτω πρόγραμμα οι μέθοδοι δηλώνονται μέσα στην κλάση αλλά ορίζονται έξω από αυτή. Για την αναφορά στις μεθόδους της κλάσης χρησιμοποιούμε το όνομα της κλάσης συνοδευόμενο από τον **τελεστή διακρίβωσης εμβέλειας (::)**.

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

class Employee { //Δήλωση κλάσης
private:
    int code;
    char name[30];
    double payrate;
    double salary;

public:
    //Δηλώσεις μεθόδων
    void readData();
    void setData(int aCode, char aName[], double aPayrate);
    double salaryCalculate(double prt);
    void printData();
    int returnCode();
    double returnSalary();
}; //Τέλος κλάσης

//Ορισμός μεθόδων έξω από την κλάση
void Employee::readData(){
    cout << "Give Code:"; cin >> code;
    cout << "Give Name:"; cin >> name;
    cout << "Give Payrate:"; cin >> payrate;
    salary = payrate * 8;
}

void Employee::printData(){
    cout << "Employee Informations" << endl;
    cout << "Employee Code: "<< code<< endl;
    cout << "Employee Name: " << name << endl;
    cout << "Employee Salary: " << salary << endl;
}

void Employee::setData(int aCode, char aName[], double aPayrate){
    code = aCode;
    strcpy(name, aName);
    payrate = aPayrate;
    salary = salaryCalculate(payrate);
}

double Employee::salaryCalculate(double prt){
    return 8 * prt;
}

double Employee::returnSalary(){
    return salary;
}

int Employee::returnCode(){
    return code;
}

int main(){
```

```
Employee e1;
e1.readData();
e1.printData();
cout << "Code: " << e1.returnCode() << endl;
cout << "Salary: " << e1.returnSalary() << endl;

return 0;
}
```