



Αντικειμενοστρεφής Προγραμματισμός

Εργαστήριο 1

Βασικά χαρακτηριστικά της C++

Κλάσεις (Classes)

Μία κλάση (class) αποτελεί ένα «αντικειμενοστρεφές εργαλείο» για τη δημιουργία νέων τύπων δεδομένων. Είναι ένα νέο είδος δομής, το οποίο όχι μόνο καθορίζει τα δεδομένα που αυτή η δομή περιέχει, αλλά και τις συναρτήσεις που δρουν πάνω σε αυτά.

Η γενική μορφή ορισμού μια κλάσης είναι η ακόλουθη:

```
Class όνομα {  
private:  
    Ιδιωτικός κώδικας (μεταβλητές και συναρτήσεις)  
public:  
    Γενικός κώδικας (μεταβλητές και συναρτήσεις)  
};
```

Η διακήρυξη του γενικού τμήματος του κώδικα πραγματοποιείται κάτω από τη λέξη κλειδί *public*. Το τμήμα αυτό περιέχει διακηρύξεις συναρτήσεων ή και μεταβλητών οι οποίες είναι προσπελάσιμες από οποιοδήποτε σημείο του προγράμματος.

Κάθε άλλο τμήμα κώδικα στο εσωτερικό της κλάσης θεωρείται ιδιωτικό (*private*) για την κλάση. Ο κώδικας που αναγράφεται στο «ιδιωτικό τμήμα» περιέχει μεταβλητές ή και συναρτήσεις που είναι δυνατό να χρησιμοποιηθούν **μόνο** από συναρτήσεις οι οποίες είναι μέλη της ίδιας της κλάσης. Δεν υπάρχει δυνατότητα προσπέλασης σ' αυτό το τμήμα από άλλα σημεία του προγράμματος έτσι ώστε να προστατεύονται τα δεδομένα και ο κώδικας από λανθασμένη χρήση του.

Συγκρίνοντας μια απλή δομή της C (*struct*) και την «αντίστοιχη» C++ κλάση στα επόμενα δυο προγράμματα μπορείτε να δείτε τα βασικά της κατασκευής κλάσεων και αντικειμένων.

Παράδειγμα σε C:

```
# include <iostream>  
using namespace std;  
  
// Conventional struct  
struct FMRadio_struct  
{
```

```

    double frequency;
    int volume;
};

// Function prototype
void changeFrequency(FMRadio_struct rad, double freq);

int main()
{
    FMRadio_struct radio;
    //we give an incorrect frequency for FM
    radio.frequency = 109.5;
    //loud volume!
    radio.volume = 150;

    changeFrequency(radio, 110.2);
    return 0;
}

// A conventional function that changes the frequency
// Does it know what it's doing?
void changeFrequency(FMRadio_struct rad, double freq)
{
    rad.frequency = freq;
}

```

Παράδειγμα σε C++:

```

#include <iostream>
using namespace std;

// Conventional struct
class FMRadio
{
private:
    double frequency;
    int volume;
public:
    void setFrequency(double freq)
    {
        if((freq > 88) && (freq < 105.5))
            frequency = freq;
        else
            frequency = 88;
    }

    void setVolume(double vol)
    {
        if((vol > 0) || (vol < 100)) //0-100 allowed
            volume = vol;
        else
            volume = 0; //default value
    }

    void print()
    {
        cout << "Volume: "<<volume<<" , frequency:"<<frequency << endl;
    }
};

int main()
{
    FMRadio radio;

```

```

//we cannot directly change freq. or volume
//radio.frequency = 109.5;
//radio.volume = 150;

//we should use "accessor" functions
radio.setVolume(40);
radio.setFrequency(109.9); //will set to default 88
radio.print();

system("pause");
return 0;
}

```

Κάτι που μπορούμε να παρατηρήσουμε από το προηγούμενο παράδειγμα είναι το γεγονός ότι οι μεταβλητές στην κλάση είναι προστατευμένες και ότι δεν μπορούμε να έχουμε άμεση πρόσβαση σε αυτές, παρά μόνο μέσω ειδικών συναρτήσεων. Αυτές οι συναρτήσεις ονομάζονται **Accessor Functions** και γενικώς είναι της μορφής `getXXX` και `setXXX` όπου `XXX` το όνομα της μεταβλητής.

Αυτό το χαρακτηριστικό ονομάζεται **encapsulation** (ενθυλάκωση) και έχει ως σκοπό την προστασία των μεταβλητών της κλάσης, μεταξύ άλλων, από αδέξιους χειρισμούς. Φανταστείτε για παράδειγμα αν κάποιος μπορούσε να αλλάξει απευθείας την μεταβλητή `frequency` στην τιμή 110.3, μια τιμή που δεν ανήκει στην μπάντα των FM. Αν όμως δώσει την τιμή αυτή στην συνάρτηση `setFrequency`, τότε αυτή η τιμή δεν θα γίνει δεκτή, αλλά θα ανατεθεί αντί για αυτήν μια default τιμή (88) που είναι μέσα στα αποδεκτά όρια.

Παράδειγμα

Για να ορίσουμε μία κλάση η οποία προσδιορίζει τα χαρακτηριστικά της οντότητας «Φοιτητής» θα μπορούσαμε να γράψουμε:

```

class Student
{
public:
    string name;
    int code;
    float grd;
};

```

Αφού οριστεί μία κλάση, μπορούμε να δηλώσουμε μεταβλητές αυτού του τύπου. Αυτές οι μεταβλητές ονομάζονται αντικείμενα (*objects*) ή στιγμιότυπα (*instances*) της κλάσης και δημιουργούνται με βάση τις προδιαγραφές που ορίζει η κλάση. Κάθε αντικείμενο αποτελεί μία πραγματική υπόσταση μίας συγκεκριμένης κλάσης. Για παράδειγμα, με την παρακάτω εντολή δηλώνουμε δύο αντικείμενα της κλάσης `Student`:

```
Student s1, s2;
```

Στις μεταβλητές-μέλη των `s1` και `s2` μπορούν να αποθηκευτούν τα δεδομένα δύο φοιτητών.

Ένας τρόπος πρόσβασης στα μέλη ενός αντικειμένου είναι με χρήση του τελεστή τελεία (`.`). Για παράδειγμα, το παρακάτω πρόγραμμα διαβάζει τα στοιχεία ενός φοιτητή, τα αποθηκεύει στα αντίστοιχα μέλη ενός αντικειμένου της κλάσης `Student` και τα εμφανίζει στην οθόνη.

```

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

```

```

class Student
{
public:
    string name;
    int code;
    float grd;
};

int main()
{
    Student s;

    cout << "Name: ";
    getline(cin, s.name);

    cout << "Code: ";
    cin >> s.code;

    cout << "Grade: ";
    cin >> s.grd;

    cout << "N: " << s.name << " C: " << s.code << " G: " << s.grd << endl;
    system("pause");
    return 0;
}

```

Άσκηση 1:

Να ορίσετε μια κλάση *Movie* η οποία θα πρέπει να περιγράφει τα στοιχεία μιας ταινίας. Τα στοιχεία που μας ενδιαφέρουν είναι τα εξής:

- Όνομα ταινίας
- Έτος παραγωγής
- Εταιρία παραγωγής
- Σκηνοθέτης

Αφού υλοποιήσετε την κλάση *Movie*, να γράψετε μια συνάρτηση *main* όπου θα πρέπει:

1. Να δημιουργήσετε ένα αντικείμενο τύπου *Movie*
2. Να δώσετε τιμές στο αντικείμενο αυτό
3. Να εκτυπώσετε τα στοιχεία της ταινίας στην οθόνη

Accessor Functions

Αν αλλάξουμε τις 3 μεταβλητές της κλάσης *Student* από *public* σε *private* δεν θα είναι δυνατή η άμεση πρόσβασή τους (π.χ. `cin >> s.code;`). Για να μπορέσουμε πια να τις αλλάξουμε θα πρέπει να φτιάξουμε συναρτήσεις πρόσβασης σε αυτές (*Accessor functions*).

Επίσης, για να μπορέσουμε να εκτυπώσουμε στην οθόνη τα στοιχεία του φοιτητή, αναγκαστήκαμε να φτιάξουμε μια συνάρτηση `display_student`.

```

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

class Student
{
    string name;
    int code;
    float grd;
}

```

```

public:
    void display_student();
    void setName(string n);
    void setCode(int cd);
    void setGrade(float g);
};

void Student::display_student()
{
    cout << "Student details: -----" << endl;
    cout << "Name: " << name << endl;
    cout << "Code: " << code << endl;
    cout << "Grade " << grd << endl;
}

void Student::setName(string n)
{
    name = n;
}

void Student::setCode(int cd)
{
    code = cd;
}

void Student::setGrade(float g)
{
    grd = g;
}

int main()
{
    Student s;
    string nam;
    int cod;
    float gg;

    cout << "Name: ";
    getline(cin, nam);

    cout << "Code: ";
    cin >> cod;

    cout << "Grade: ";
    cin >> gg;

    s.setName(nam);
    s.setCode(cod);
    s.setGrade(gg);

    s.display_student();
    system("pause");
    return 0;
}

```

Άσκηση 2:

Να συνεχίσετε τον κώδικα της πρώτης άσκησης ως εξής:

1. Να μετατρέψετε όλες τις μεταβλητές σε ιδιωτικές (private)
2. Να προσθέσετε accessor functions για να μπορέσετε να δώσετε τιμές στις μεταβλητές
3. Να υλοποιήσετε μια συνάρτηση display() που θα εκτυπώνει στην οθόνη τα στοιχεία της ταινίας.
4. Να τροποποιήσετε την main ώστε να χρησιμοποιεί αυτές τις συναρτήσεις

Οι συναρτήσεις που έχουμε δει ως τώρα αφορούν την απόδοση τιμών σε μεταβλητές της κλάσης. Μια κλάση όμως μπορεί να περιέχει και άλλου είδους συναρτήσεις. Για παράδειγμα, αν η κλάση Student περιείχε ένα πίνακα με όλους τους βαθμούς που έχει πάρει ο φοιτητής ως τώρα, θα μπορούσαμε να φτιάξουμε μια συνάρτηση που θα έβγαζε το μέσο όρο αυτόματα.

```
#include <iostream>
#include <string>
#define MAX_COURSES 50 //maximum 50 courses
using namespace std;

class Student
{
public:
    string name;
    int code;
    float grades[MAX_COURSES];
    float calculate_average();
};

float Student::calculate_average()
{
    int count = 0;
    float sum = 0.0;
    for(int i=0; i<MAX_COURSES; i++)
    {
        if(grades[i] > 0)
        {
            sum += grades[i];
            count++;
        }
    }
    if(count > 0)
        return sum/count;
    else
        return 0;
}

int main()
{
    Student s;

    s.name = "John Doe";
    s.code = 1234;
    s.grades[0] = 8.2;
    s.grades[1] = 6.2;
    s.grades[2] = 5.1;
    s.grades[3] = 7.7;

    float avg = s.calculate_average();
    cout << "Average is: " << avg << endl;
```

```

    system("pause");
    return 0;
}

```

Άσκηση 3:

Στην κλάση `Movie`, να υλοποιήσετε μια συνάρτηση `set_values`, η οποία θα σας δίνει την δυνατότητα να δώσετε τιμές σε ένα αντικείμενο τύπου `Movie` με μια κίνηση, αντί να πρέπει να καλέσετε μια προς μια τις αντίστοιχες accessor functions.

Τρόπος κλήσης:

```

v.set_values("The revenant", 2015, "New Regency Productions", "A.
Inaritu");

```

Επίσης, να προσθέσετε μια νέα συνάρτηση μέσα στην κλάση η οποία να παίρνει ως παράμετρο το τρέχον έτος και θα επιστρέφει την ηλικία της ταινίας.

Hint: η συνάρτηση ορίζεται ως εξής:

```

int getMovieAge(int current_year);

```

Παράρτημα: Η κλάση `string`

Για τον χειρισμό των strings στην C++, εκτός από τα ήδη γνωστά strings της C, μπορείτε να χρησιμοποιείτε την νέα κλάση `string` της C++. Η κλάση αυτή παρέχει πολλές ευκολίες στον χρήστη της, όπως η προσθήκη ενός string σε ένα άλλο (με την συνάρτηση `append`), την αλλαγή ενός κομματιού του string (με την `replace`), και κυρίως η δυνατότητα αύξησης του μεγέθους του string χωρίς να προβληματιζόμαστε για θέματα χώρου μνήμης.

Για την χρήση της κλάσης είναι απαραίτητες οι δηλώσεις:

```

#include <string>
using namespace std;

```

Ακολουθούν κάποια παραδείγματα χρήσης της κλάσης:

```

string s = "abc def abc";
string s2 = "abcde uvwxyz";
char c;
char ch[] = "aba rrr ww";
char *cp = ch;
unsigned int i;

```

Stream input	<code>cin >> s;</code>
Stream output	<code>cout << s;</code>
Line input	<code>getline(cin, s);</code>
Ανάθεση	<code>s = s2;</code> <code>s = "abc";</code> <code>s = ch;</code> <code>s = cp;</code>
Subscript	<code>s[1] = 'c';</code> <code>c = s[1];</code>
Μήκος	<code>int i = s.length();</code> <code>int i = s.size();</code>
Έλεγχος αν είναι άδειο	<code>if(s.empty()) ...</code> <code>if(s == "") ...</code>
Σύγκριση	<code>if (s < s2) ...</code>
Ένωση	<code>s2 = s2 + "x";</code> <code>s2 += "x";</code>

Substring	<pre>s = s2.substr(1,4); s = s2.substr(1,50);</pre>
-----------	---

Περισσότερες πληροφορίες μπορείτε να αναζητήσετε στο link:
<http://www.cplusplus.com/reference/string/string/>

Σε αυτό το site γενικότερα θα βρείτε πολλές πληροφορίες για την C++ και για τα χαρακτηριστικά της. Να σημειωθεί ότι καλό είναι να αρχίσετε από τώρα να εξοικειώνεστε με την αναζήτηση ανάλογων πληροφοριών (π.χ. το τι συναρτήσεις και μεταβλητές περιέχει μια κλάση, ή το πως καλείται μια νέα συνάρτηση) από το Internet.