

Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό με τη C++

#1

Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)
Πανεπιστήμιο Ιωαννίνων
Γκόγκος Χρήστος

Δομημένος προγραμματισμός και αντικειμενοστραφής προγραμματισμός

- Δομημένος προγραμματισμός (structured programming)
 - Ο προγραμματισμός στηρίζεται σε καλά ορισμένες δομές ελέγχου
 - Δομή ακολουθίας, δομές ελέγχου (if, switch), δομές επανάληψης (π.χ. while), εκφράσεις και εκχωρήσεις.
 - Δεδομένα (μεταβλητές, πίνακες, εγγραφές) είναι ξεχωριστά από τις λειτουργίες σε αυτά.
- Διαδικασιακός προγραμματισμός (procedural programming)
 - Ο διαδικασιακός προγραμματισμός ακολουθεί τον δομημένο προγραμματισμό και διασπά το πρόγραμμα σε δομές δεδομένων και υποπρογράμματα (διαδικασίες και συναρτήσεις) που επιδρούν πάνω στις δομές δεδομένων.
- Αντικειμενοστραφής προγραμματισμός (object oriented programming)
 - Αναπτύχθηκε ως μια επέκταση του διαδικασιακού προγραμματισμού.
 - Ο αντικειμενοστραφής προγραμματισμός βασίζεται στην έννοια του **αντικειμένου**
 - Τα αντικείμενα ομαδοποιούν δεδομένα και τις λειτουργίες πάνω σε αυτά τα δεδομένα.
 - Επιτρέπουν την **απόκρυψη πληροφορίας** (information hiding) που οδηγεί σε προγράμματα που είναι ευκολότερο να αναπτυχθούν, να συντηρηθούν και να επεκταθούν.

Βασικές έννοιες Αντικειμενοστραφούς Προγραμματισμού

- Μια βασική έννοια των αντικειμενοστραφών γλωσσών είναι η **ενθυλάκωση (encapsulation)** δεδομένων και συναρτήσεων μαζί σε μονάδες που ονομάζονται **αντικείμενα (objects)**.
- Ένα αντικείμενο στη C++ αποτελείται από:
 - **Όνομα**: αποτελεί τον τρόπο με τον οποίο γίνεται η αναφορά στα αντικείμενα μέσα στο πρόγραμμα.
 - **Μέλη δεδομένα (member data)**: είναι τα δεδομένα που περιέχονται σε ένα αντικείμενο. Τα member data αναφέρονται και ως κατάσταση (state) του αντικειμένου, ενώ σε άλλες αντικειμενοστραφείς γλώσσες προγραμματισμού όπως η Java και η Python ονομάζονται χαρακτηριστικά (attributes) ή ιδιότητες (properties) του αντικειμένου.
 - **Μέλη συναρτήσεις (member functions)**: είναι οι συναρτήσεις που επιδρούν στα δεδομένα του αντικειμένου. Τα member functions σε γλώσσες προγραμματισμού όπως η Java και η Python ονομάζονται μέθοδοι (methods).
 - **Διεπαφή (interface)**: Καθορίζει τους τρόπους με τους οποίους ο προγραμματιστής μπορεί απευθείας να προσπελάει member data και member functions ενός αντικειμένου.

Κλάσεις

- Η **κλάση** (class) είναι μια ακόμα θεμελιώδης έννοια στον αντικειμενοστραφή προγραμματισμό και μπορεί να περιγραφεί ως το «αρχιτεκτονικό σχέδιο» (blueprint) που προσδιορίζει ένα νέο τύπο αντικειμένων.
- Η κλάση καθορίζει:
 - τα δεδομένα, τις συναρτήσεις και τη διεπαφή των αντικειμένων της κλάσης.
 - το πως τα αντικείμενα μιας κλάσης συμπεριφέρονται παρέχοντας κώδικα που υλοποιεί τις συναρτήσεις που σχετίζονται με την κλάση.
- Ο προγραμματιστής μπορεί να δημιουργεί ένα ή περισσότερα αντικείμενα μιας κλάσης.

Πως ορίζεται και χρησιμοποιείται μια κλάση σε ένα πρόγραμμα;

- **Δήλωση (declaration) της κλάσης:** επιλογή του τι θα αποθηκεύουν (member data) και πως θα συμπεριφέρονται (member functions) τα αντικείμενα της κλάσης.
- **Ορισμός (definition) των μελών συναρτήσεων:** παρέχεται υλοποίηση για κάθε μέλος συνάρτηση της κλάσης.
- **Χρήση της κλάσης για τη δημιουργία αντικειμένων:** δήλωση και ορισμός νέων αντικειμένων ή όπως αλλιώς ονομάζονται στιγμιοτύπων της κλάσης με τον τρόπο που δηλώνονται και ορίζονται οι απλές μεταβλητές.

Παράδειγμα: Δήλωση κλάσης

```
class Circle {  
public:                                // διεπαφή (interface)  
    void SetRadius(double r); // θέτει το μέλος δεδομένων radius στην τιμή r  
    double AreaOf();          // επιστρέφει το εμβαδόν του κύκλου  
private:  
    double radius; // ακτίνα του κύκλου  
};
```

Ορισμός των συναρτήσεων μελών

- Υπάρχουν 2 τρόποι με τους οποίους μπορεί να **οριστεί** ο κώδικας που περιέχουν οι συναρτήσεις μέλη μιας κλάσης:
 - Μαζί με τη δήλωση της κλάσης (έχουμε δηλαδή δήλωση και ορισμό μαζί)
 - Μετά τη δήλωση της κλάσης.
- Η αναφορά σε μια συνάρτηση μέλος γίνεται ως εξής:
όνομαΚλάσης::όνομαΣυνάρτησηςΜέλους
- Το αναγνωριστικό **όνομαΚλάσης::όνομαΣυνάρτησηςΜέλους** αναφέρεται στη συνάρτηση μέλος **όνομαΣυνάρτησηςΜέλους** της κλάσης **όνομαΚλάσης**
- Ο τελεστής :: ονομάζεται τελεστής προσδιορισμού εμβέλειας (scope resolution operator).
- Μετά τη δήλωση της κλάσης, οι συναρτήσεις μέλη ορίζονται παρόμοια με τις συνηθισμένες συναρτήσεις.

Παράδειγμα: ορισμός υλοποιήσεων συναρτήσεων μελών **εκτός** της κλάσης

```
class Circle
{
public:
    // διεπαφή (interface)
    void SetRadius(double r); // θέτει το μέλος δεδομένων radius στην τιμή r
    double AreaOf();          // επιστρέφει το εμβαδόν του κύκλου
private:
    double radius; // ακτίνα του κύκλου
};

// υλοποιήσεις member functions
void Circle::SetRadius(double r) {
    radius = r;
}
double Circle::AreaOf() {
    return (3.14 * radius * radius);
}
```


Παράδειγμα: ορισμός υλοποιήσεων συναρτήσεων μελών **εντός** της κλάσης

```
class Circle
{
public:
    void SetRadius(double r) {
        radius = r;
    }
    double AreaOf() {
        return (3.14 * radius * radius);
    }
private:
    double radius; // ακτίνα του κύκλου
};
```

Αντικείμενα

- Από τη στιγμή που μια κλάση έχει δηλωθεί και οριστεί, μπορούν να δηλώνονται και να χρησιμοποιούνται αντικείμενα της κλάσης όπως οποιοσδήποτε άλλος τύπος δεδομένων.
- Ο προγραμματιστής μπορεί να δηλώνει και να ορίζει ταυτόχρονα ένα αντικείμενο με τον ακόλουθο τρόπο:
 - **ΌνομαΚλάσης όνομαΑντικειμένου;**
- Η παραπάνω εντολή δημιουργεί ένα αντικείμενο βάσει των «οδηγιών» που περιέχονται στην κλάση **ΌνομαΚλάσης** και το αντικείμενο μπορεί πλέον να αναφερθεί με το αναγνωριστικό **όνομαΑντικειμένου**
- Ο τελεστής . (dot operator) μπορεί να χρησιμοποιηθεί για να προσπελαστούν τα δημόσια μέλη ενός αντικειμένου.
- Η μορφή με την οποία γίνεται η αναφορά στα **δημόσια μέλη** ενός αντικειμένου είναι:
 - **όνομαΑντικειμένου.ΜέλοςΣυνάρτηση()**
 - **όνομαΑντικειμένου.ΜέλοςΔεδομένων**

Αντικείμενα

```
int main() {  
    Circle C1;  
    Circle C2; } Δημιουργία 2 αντικειμένων Circle (στιγμιότυπων της κλάσης Circle)  
  
    C1.SetRadius(1);  
    C2.SetRadius(10); } Κλήση της συνάρτησης μέλους SetRadius()  
    cout<<"The area of C1 is "<<C1.AreaOf()<<endl;  
    cout<<"The area of C2 is "<<C2.AreaOf()<<endl;  
  
    C1.SetRadius(2);  
    C2.SetRadius(3);  
    cout<<"The area of C1 is "<<C1.AreaOf()<<endl;  
    cout<<"The area of C2 is "<<C2.AreaOf()<<endl;  
    return 0;  
}
```

The area of C1 is 3.14
The area of C2 is 314
The area of C1 is 12.56
The area of C2 is 28.26

Ο κώδικας συνολικά

- Ο κώδικας του `sample1.cpp`
 - Δηλώνει την κλάση `Circle` και ορίζει τα μέλη της και τη διεπαφή της.
 - Ορίζει την υλοποίηση των συναρτήσεων μελών της κλάσης `Circle`.
 - Δηλώνει 2 αντικείμενα της κλάσης `Circle` με ονόματα `C1` και `C2`.
 - Χρησιμοποιεί τις διεπαφές των `C1` και `C2` για να αποθηκεύσει τις ακτίνες των 2 κύκλων και στη συνέχεια για να υπολογίσει το εμβαδό τους.

<https://github.com/chgogos/oop/blob/master/various/COP3330/lect1/sample1.cpp>

<https://github.com/chgogos/oop/tree/master/various/COP3330/lect1/sample1> έκδοση με διαμέριση κώδικα

Άσκηση

- Δημιουργήστε μια κλάση με όνομα Clock που να αναπαριστά μια χρονική στιγμή σε μια ημέρα ως ΩΩ:ΛΛ:ΔΔ (π.χ. 17:03:55) και να έχει ως μέλη δεδομένων τις ακέραιες μεταβλητές hour, min, sec.
- Συμπληρώστε τις ακόλουθες συναρτήσεις μέλη:
 - set_hour_min_sec(...) που θέτει τιμές στα τρία μέλη δεδομένων
 - reset() μηδενίζει την ώρα, τα λεπτά και τα δευτερόλεπτα
 - tick() που προσθέτει ένα δευτερόλεπτο στη χρονική στιγμή
 - display() που εμφανίζει τη χρονική στιγμή σε μορφή ΩΩ:ΛΛ:ΔΔ
- Στη main() δημιουργήστε ένα αντικείμενο Clock με αρχική τιμή 12:00:00, καλέστε την συνάρτηση tick() 5000 φορές, εμφανίστε την ώρα με τη display(), καλέστε την reset() και εμφανίστε πάλι την ώρα με την display().
- Παρατήρηση: Δηλώστε όλα τα μέλη δεδομένα και τα μέλη συναρτήσεις ως public.

Λύση άσκησης

```
#include <iomanip> // για setw και setfill
#include <iostream>
using namespace std;
class Clock {
public:
    int hour, min, sec;
    void set_hour_min_sec(int h, int m, int s) {
        hour = h; min = m; sec = s;
    }
    void reset() {hour = 0; min = 0; sec = 0;}
    void tick() {
        sec++;
        if (sec >= 60) {
            sec = 0;
            min++;
            if (min >= 60) {
                min = 0;
                hour++;
                if (hour >= 24) hour = 0;
            }
        }
    }
    void display() { cout << setfill('0') << setw(2) <<
        hour << ":" << setw(2) << min << ":" << setw(2) <<
        sec << endl;
    }
};
```

```
int main() {
    Clock c;
    c.set_hour_min_sec(12, 0, 0);
    for (int i = 0; i < 5000; i++) c.tick();
    cout << "Μετά από 5000 δευτερόλεπτα: ";
    c.display();
    c.reset();
    cout << "Μετά το reset: ";
    c.display();
    return 0;
}
```

```
$ g++ exercise1.cpp -o a.exe
$ ./a.exe
Μετά από 5000 δευτερόλεπτα: 13:23:20
Μετά το reset: 00:00:00
```

<https://github.com/chgogos/oop/blob/master/variou/COP3330/lect1/exercise1.cpp>

https://github.com/chgogos/oop/blob/master/variou/COP3330/lect1/exercise1_C++20.cpp

https://github.com/chgogos/oop/blob/master/variou/COP3330/lect1/exercise1_C++23.cpp

Σύνοψη (ερωτήσεις)

- Τι είναι κλάση;
- Τι είναι αντικείμενο;
- Ποια είναι τα βασικά στοιχεία ενός αντικειμένου;
- Πως καθορίζεται η μορφή των αντικειμένων (δηλαδή τα μέλη δεδομένων και τα μέλη συναρτήσεις που περιέχει);
- Με ποια σειρά γίνεται χρήση των αντικειμένων σε ένα πρόγραμμα;
- Τι είναι ο τελεστής :: ;
- Τι είναι ο τελεστής . ;

Σύνοψη (ερωτήσεις/απαντήσεις)

- Τι είναι κλάση;
 - Κλάση είναι η περιγραφή ενός νέου τύπου με δεδομένα και συναρτήσεις που επιδρούν σε αυτά.
- Τι είναι αντικείμενο;
 - Αντικείμενο είναι μια μονάδα που ενθυλακώνει δεδομένα και συναρτήσεις.
- Ποια είναι τα βασικά στοιχεία ενός αντικειμένου;
 - Τα βασικά στοιχεία ενός αντικειμένου είναι τα ακόλουθα 4: όνομα , μέλη δεδομένα, μέλη συναρτήσεις και διεπαφή.
- Πως καθορίζεται η μορφή των αντικειμένων;
 - Η μορφή των αντικειμένων καθορίζεται από την κλάση στην οποία ανήκουν.
- Με ποια σειρά γίνεται χρήση των αντικειμένων σε ένα πρόγραμμα;
 - Η χρήση των αντικειμένων σε ένα C++ πρόγραμμα ακολουθεί τη σειρά: δήλωση, ορισμός και χρήση.
- Τι είναι ο τελεστής :: ;
 - Ο τελεστής :: είναι ο τελεστής εμβέλειας και χρησιμοποιείται έτσι ώστε να οριστούν οι συναρτήσεις μιας κλάσης εκτός της δήλωσης της κλάσης.
- Τι είναι ο τελεστής . ;
 - Ο τελεστής . είναι ο τελεστής τελείας και χρησιμοποιείται για να κληθεί μια συνάρτηση μέλος ή να προσπελαστούν τα μέλη δεδομένων ενός αντικειμένου.

Αναφορές

- <http://www.cs.fsu.edu/~xyuan/cop3330/>