



Πανεπιστήμιο Αιγαίου

Μεθοδολογίες και Γλώσσες Προγραμματισμού I

Αντικειμενοστρεφής προγραμματισμός

Εργίνα Καβαλλιεράτου (kavallieratou@aegean.gr)

Μόνιμη Επίκουρος Καθηγήτρια

Τμήμα Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Σημερινό μάθημα

- ✓ Μειονεκτήματα Δομημένου Προγραμματισμού
- ✓ Αντικειμενοστρεφής προγραμματισμός
- ✓ Ορισμοί
- ✓ Κλάσεις
- ✓ Αντικείμενα
- ✓ Χαρακτηριστικά ΑΠ
- ✓ C++ Class

Δομημένος Προγραμματισμός

Ο θεμέλιος λίθος του δομημένου προγραμματισμού είναι ο τεμαχισμός μεγάλων προγραμμάτων σε επιμέρους κομμάτια (λογισμικές μονάδες), όπου το καθένα έχει συγκεκριμένη αποστολή, αλλά συνδέεται και επικοινωνεί με τα άλλα για να δημιουργήσουν ένα ολοκληρωμένο πρόγραμμα.

Μειονεκτήματα Δομημένου Προγραμματισμού

- ✓ Η πολυπλοκότητα
- ✓ Τα δεδομένα έχουν υποεκτιμηθεί
- ✓ Η τοπικότητα των μεταβλητών
Υπάρχουν οι καθολικές αλλά τότε αυξάνει η πολυπλοκότητα
- ✓ Αυξημένη πιθανότητα λάθους
- ✓ Αυξημένη πολυπλοκότητα και στη διαχείριση των δεδομένων
- ✓ Η δημιουργία νέων τύπων δεδομένων είναι δύσκολη.

Αντικειμενοστρεφής προγραμματισμός

Στον Αντικειμενοστρεφή προγραμματισμό, δίνεται έμφαση στη δημιουργία μονάδων οι οποίες θα περιλαμβάνουν τα δεδομένα αλλά και τις εντολές που τα διαχειρίζονται.

Ορισμοί

- ✓ Ο κόσμος αποτελείται από αντικείμενα απλά ή σύνθετα, που αλληλεπιδρούν μεταξύ τους
- ✓ Για κάθε αντικείμενο του φυσικού κόσμου μπορούμε να ορίσουμε, μια αφηρημένη έννοια που περιγράφει:
 - τον τρόπο με τον οποίο το αντικείμενο επικοινωνεί με το περιβάλλον του
 - το πώς αυτό είναι οργανωμένο εσωτερικά
- ✓ Ο ΑΠ ονομάζει στιγμιότυπα (instances) τα πραγματικά αντικείμενα και κλάσεις (classes) τις αφηρημένες έννοιες που τα περιγράφουν.

Ορισμοί

- ✓ Ο όρος αντικείμενο χρησιμοποιείται και για την αφηρημένη έννοια (κλάση) αλλά και για τα στιγμιότυπά της
- ✓ Ο ΑΠ θεωρεί όλες τις οντότητες, πραγματικές και αφηρημένες, αντικείμενα (objects).
- ✓ Εδώ θα ονομάζουμε την αφηρημένη έννοια κλάση και τα στιγμιότυπα της αντικείμενα.
- ✓ Τα αντικείμενα είναι σαν μεταβλητές και οι κλάσεις σαν τύποι.

Κλάσεις

Καθηγητής
Αρ. Ταυτότητας
Όνομα
Επώνυμο
Διεύθυνση
Τηλέφωνο
Προσθήκη Καθηγητή()
Διαγραφή Καθηγητή()
Μεταβολή στοιχείων Καθηγητή()

Μάθημα
Κωδικός μαθήματος
Θεματική ενότητα
Τίτλος
Διδάσκων
Προσθήκη Μαθήματος()
Διαγραφή Μαθήματος()
Μεταβολή στοιχείων Μαθήματος()
Ανάθεση Μαθήματος()

Αντικείμενα ή Στιγμιότυπα

Καθηγητής 01

Αρ. Ταυτότητας: **A123456**

Όνομα: **Βασίλειος**

Επώνυμο: **Βασιλείου**

Διεύθυνση: **Αγ. Βασιλείου 1**

Τηλέφωνο: **9876543**

Προσθήκη Καθηγητή()

Διαγραφή Καθηγητή()

Μεταβολή στοιχείων Καθηγητή()

Καθηγητής 02

Αρ. Ταυτότητας: **B987654**

Όνομα: **Γεώργιος**

Επώνυμο: **Γεωργίου**

Διεύθυνση: **Αγ. Γεωργίου 1**

Τηλέφωνο: **7654321**

Προσθήκη Καθηγητή()

Διαγραφή Καθηγητή()

Μεταβολή στοιχείων Καθηγητή()

Μάθημα 01

Κωδικός μαθήματος: **ΠΛ-034**

Θεματική ενότητα: **Πληροφορική**

Τίτλος: **Προγραμματισμός Η/Υ**

Διδάσκων: **B987654**

Προσθήκη Μαθήματος()

Διαγραφή Μαθήματος()

Μεταβολή στοιχείων Μαθήματος()

Ανάθεση Μαθήματος()

Κλάση/Αντικείμενο

- ✓ Στον ΑΠ δεν ασχολούμαστε με το πώς θα χωρίσουμε το πρόβλημα σε συναρτήσεις αλλά σε κλάσεις και αντικείμενα.
- ✓ Μία κλάση μπορεί να έχει ένα ή περισσότερα αντικείμενα.
- ✓ Κάθε αντικείμενο μπορεί να περιέχει μία ή περισσότερες συναρτήσεις, **συναρτήσεις-μέλη**
- ✓ Η κλάση λειτουργεί ως πρότυπο: δηλώνουμε σε αυτή δεδομένα και συναρτήσεις που θα έχουν τα αντικείμενα της
- ✓ “Όταν δημιουργήσουμε αντικείμενα θα έχουν ότι έχει η κλάση που ανήκουν

Παράδειγμα

```
#include...
class Cat {
    unsigned int itsAge;
    unsigned int itsWeight;
    Meow();
};
//...
main()
{
    unsigned int GrossWeight;
    Cat Frisky;
    ...}
```

Χαρακτηριστικά ΑΠ

- ✓ Διαθέτει μηχανισμούς:
 - απόκρυψης πληροφορίας
 - περάσματος μηνυμάτων
 - δημιουργίας και καταστροφής στιγμιοτύπων
 - αναπαράστασης ιεραρχίας
 - κληρονομικότητας κ.α.

Χαρακτηριστικά ΑΠ

- ✓ Οι μηχανισμοί αυτοί βελτιώνουν την ποιότητα του λογισμικού ως προς τα χαρακτηριστικά του:
 - επαναχρησιμοποίηση (reusability)
 - επεκτασιμότητα (extendibility)
 - ορθότητα (correctness)
 - ευρωστία (robustness)
 - συντηρησιμότητα (maintainability)

C++ Class

```
class Classic_Example {  
public:  
// Δεδομένα και διαδικασίες προσβάσιμα από παντού  
protected:  
//Δεδομένα και διαδικασίες προσβάσιμα από την κλάση  
// τις παραγόμενες κλάσεις και friends κλάσεις  
private:  
// Δεδομένα και διαδικασίες προσβάσιμα από την κλάση  
// και friends κλάσεις  
};
```


C++ Class - Συμβουλή

- ✓ Δηλώστε τις συναρτήσεις στο PUBLIC
- ✓ Δηλώστε τα πεδία δεδομένων στο PRIVATE

C++ Class - Παράδειγμα

```
class Date{  
    private:  
        int _day;  
        int _month;  
        int _year;  
    public:  
        // ...  
};
```

C++ Classes

```
class Mailbox
{
private:
    //...
public:
    void add(Message);
    Message get_current();
    void delete_current();
    // ...
};
```

Classes vs structs

- ✓ Η κλάση Date είναι παρόμοια με τον ορισμό της στη C ως structure

```
struct date
```

```
{ int day;
```

```
  int month;
```

```
  int year;
```

```
};
```

- ✓ Τα structs δεν παρέχουν τη δυνατότητα ελέγχου πρόσβασης στα δεδομένα.

Πρόσβαση σε συναρτήσεις κλάσεων

- ✓ Εκτός κλάσης, public συναρτήσεις μπορούν να κληθούν μόνο με αναφορά σε αντικείμενο της κλάσης.
- ✓ Μέσα στην κλάση, οι διάφορες συναρτήσεις μπορούν να κληθούν από άλλες συναρτήσεις ως απλές συναρτήσεις

C++ Class

- ✓ Τα δεδομένα μιας κλάσης καλούνται ΔΕΔΟΜΕΝΑ ΜΕΛΗ ενώ οι διαδικασίες λέγονται ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΛΗ.
- ✓ Ο τελεστής '.' χρησιμοποιείται για πρόσβαση και στα δεδομένα και στις συναρτήσεις.
- ✓ Η συνάρτηση αλλάζει τις τιμές που ανήκουν στο αντικείμενο που καλεί τη διαδικασία
`b.advance(30);`
- ✓ Μια διαδικασία κλάσης μπορεί να έχει και ανεξάρτητες με την κλάση παραμέτρους

Άσκηση

Ο Γιώργος αποφασίζει να ανοίξει ένα μαγαζί πρόχειρου φαγητού (fast food). Γνωρίζει πως χρειάζεται ένα φούρνο μικροκυμάτων, μια τοστιέρα, ένα ψυγείο, ένα πλυντήριο πιάτων, και ορισμένα άλλα αντικείμενα, τα οποία είναι απαραίτητα για να παρέχει σωστές υπηρεσίες το μαγαζί του. Ο Γιώργος αρχίζει με την ετοιμασία του χώρου και όταν αυτός είναι έτοιμος, ειδοποιεί να του φέρουν και εγκαταστήσουν τα επί μέρους αντικείμενα συσκευές που απαρτίζουν το fast food. Μια ανακοίνωση στην είσοδο κάτω από την πινακίδα *fast food* ο Γιώργος γράφει «Ζητούνται υπάλληλοι».

Μετά από μερικές μέρες το μαγαζί είναι έτοιμο, και ο Γιώργος με τους δύο υπαλλήλους που ήδη προσέλαβε, τη Μαίρη και το Νίκο, είναι έτοιμοι να υποδεχτούν τον πρώτο τους πελάτη.

Ο Χρήστος, μπαίνει στο μαγαζί. «Ένα τοστ με τυρί ζαμπόν παρακαλώ, μια τυρόπιτα και μια παγωμένη ΗΒΗ». Ο Γιώργος λέει στη Μαίρη «Φτιάξε σε παρακαλώ το τοστ», και αυτός αναλαμβάνει την τυρόπιτα. Διαπιστώνει όμως πως είναι κρύα και αποφασίζει να χρησιμοποιήσει το φούρνο μικροκυμάτων.

Μετά από λίγο ο Χρήστος πήρε μια σακούλα και αφού πλήρωσε βγήκε από το μαγαζί.

Πιθανή λύση

Κλάση	Στιγμιότυπο
Ιδιοκτήτης	Γιώργος
fast food	«fast food ο Γιώργος»
Υπάλληλος	Μαίρη, Νίκος
Πελάτης	Χρήστος, Κώστας
Τοστιέρα	οι δύο τοστιέρες του μαγαζιού
Φούρνος μικροκυμάτων	ο φούρνος του μαγαζιού
Πρόσωπο	Γιώργος, Μαίρη, Νίκος, Χρήστος, Κώστας

Παράδειγμα

```
#include <iostream>
const double pi = 3.14159;
const char newline = '\n';
int main ()
{
    double r=5.0;
    double circle;
    circle = 2 * pi * r;
    cout << circle;
    cout << newline;
}
```

Παράδειγμα

```
#include <iostream>
#include <string>
int main ()
{
    string mystring;
    mystring = "This is a string";
    cout << mystring;
    return 0;
}
```

Παράδειγμα

```
#include <iostream>
int main ()
{
    int i;
    cout << "Please enter an integer value: ";
    cin >> i;
    cout << "The value you entered is " << i;
    cout << " and its double is " << i*2 << ".\n";
    return 0;
}
```

Παράδειγμα

```
#include <iostream>
int main ()
{
    int a,b,c;
    a=2;
    b=7;
    c = (a>b) ? a : b;
    cout << c << '\n';
}
```

Παράδειγμα

```
#include <iostream>
using namespace std;
int main ()
{
    int a, b=3;
    a = b;
    a+=2;
    cout << a;
}
```

Παράδειγμα

```
#include <iostream.h>
class Cat
{ public:
    int itsAge;
    int itsWeight;
};
void main() {
    Cat Frisky;
    Frisky.itsAge = 5;
    cout << "Frisky is a cat who is " ;
    cout << Frisky.itsAge << " years old.\n";
}
```

Παράδειγμα

```
#include <iostream>
class CRectangle {
int x, y;
public:
    void set_values (int,int);
    int area () {return (x*y);}
};
void CRectangle::set_values (int a, int b) {
    x = a;
    y = b;
}
```

Παράδειγμα

```
int main () {  
    CRectangle rect;  
    rect.set_values (3,4);  
    cout << "area: " << rect.area();  
    return 0;  
}
```