



Πανεπιστήμιο Αιγαίου

Μεθοδολογίες και Γλώσσες Προγραμματισμού I

Εισαγωγή δεδομένων σε κλάση

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

- ✓ Επικοινωνία με την κλάση
- ✓ Πρόσβαση σε στοιχεία κλάσης
- ✓ Τελεστής Εμβέλειας ::
- ✓ Απόδοση τιμής σε μεταβλητή
- ✓ Επιστροφή δεδομένων
- ✓ Constructor (Δομητής)
- ✓ Destructor (αποδομητής)

Επικοινωνία με την κλάση

- ✓ Για να καλέσουμε μία συνάρτηση-μέλος από το κυρίως πρόβλημα γίνεται ΠΑΝΤΑ με αναφορά στο αντικείμενο που αντιστοιχεί:

<όνομα αντικειμένου>.<όνομα συνάρτησης-μέλους>

Παράδειγμα

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

Πρόσβαση σε στοιχεία κλάσης

- ✓ Ότι είναι ως public, έχει πρόσβαση από οπουδήποτε στο πρόγραμμα.

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";
}
```


Πρόσβαση σε στοιχεία κλάσης

- ✓ Αν δεν διευκρινιστεί ο τύπος των δεδομένων στην κλάση, ο εξ' ορισμού τύπος είναι private
- ✓ Στα private δεδομένα δεν μπορούμε να έχουμε πρόσβαση εκτός κλάσης.
- ✓ Αυτό συμβαίνει για την ασφάλεια των δεδομένων
- ✓ Για το λόγο αυτό πρέπει να δημιουργήσουμε κατάλληλες συναρτήσεις για εργασίες όπως
 - Απόδοση τιμής σε μεταβλητή
 - Επιστροφή δεδομένων κλπ

Τελεστής Εμβέλειας ::

“Όταν δηλώνουμε πρότυπο συνάρτησης μέσα σε μία κλάση και θέλουμε να ορίσουμε τη συνάρτηση εκτός κλάσης πρέπει να χρησιμοποιήσουμε τον Τελεστής Εμβέλειας ::

Τελεστής Εμβέλειας ::

```
class foo {
```

```
    ...
```

```
    int sum(void);
```

```
};
```

```
int foo::sum(void) {
```

```
    int sum=0;
```

```
    for (int i=0;i<foo::i;i++) {
```

```
        sum += a[i];
```

```
    }
```

```
    return (sum);
```

```
}
```

Απόδοση τιμής σε μεταβλητή

- ✓ Υπάρχουν συναρτήσεις που χρησιμοποιούνται για να αποδώσουν τιμή σε `private` μεταβλητή μιας κλάσης
- ✓ Συνήθως ονομάζονται
`Set_<όνομα μεταβλητής>`

Παράδειγμα

```
class Cat {  
public:  
    void SetAge (int age);  
    void Meow();  
private:  
    int itsAge;  
};  
...  
void Cat::SetAge(int age) {  
    itsAge = age;  
}  
...
```

Επιστροφή δεδομένων

- ✓ Υπάρχουν συναρτήσεις που χρησιμοποιούνται για να μεταφέρουν τιμή private μεταβλητής εκτός κλάσης
- ✓ Συνήθως ονομάζονται
Get_<όνομα μεταβλητής>

Παράδειγμα

```
class Cat {  
public:  
    void SetAge (int age);  
    int GetAge();  
    void Meow();  
private:  
    int itsAge;  
};  
...  
int Cat::GetAge() {  
    return itsAge;  
}  
...
```

Πρόγραμμα/1

```
#include <iostream.h>
class Cat {
    public:
        int GetAge();
        void SetAge (int age);
        void Meow();
    private:
        int itsAge;
};
```


Πρόγραμμα/2

```
int Cat::GetAge() {  
    return itsAge;  
}  
void Cat::SetAge(int age) {  
    itsAge = age;  
}  
void Cat::Meow() {  
    cout << "Meow.\n";  
}
```

Πρόγραμμα/3

```
int main() {  
    Cat Frisky;  
    Frisky.SetAge(5);  
    Frisky.Meow();  
    cout << "Frisky is a cat who is " ;  
    cout << Frisky.GetAge() << " years old.\n";  
    Frisky.Meow();  
    return 0;  
}
```

Output

Meow.

Frisky is a cat who is 5 years old.

Meow.

Αρχικοποίηση κλάσης

- ✓ Οι classes περιλαμβάνουν μια ιδιαίτερη συνάρτηση-μέλος που ονομάζεται constructor (δομητής).
- ✓ Ο constructor έχει το ίδιο όνομα με την κλάση
- ✓ Μπορεί να παίρνει παράμετρο αν χρειάζεται αλλά δεν έχει τύπο επιστροφής, ούτε καν void

Constructor (Δομητής)

```
class Date
{
public:
    Date(int d, int m, int y);
    // ...
};
```

```
Date::Date(int d, int m, int y)
: day(d),
  month(m),
  year(y)
{ }
```

Constructor (Δομητής)

- ✓ Μεγάλο πλεονέκτημα των ΔΟΜΗΤΩΝ είναι το ότι η ύπαρξή τους απομακρύνει την πιθανότητα μη αρχικοποιημένων μεταβλητών. Αν έχει δηλωθεί η `Date(int, int, int)` τότε η `Date d;` είναι λάθος
- ✓ Στις κλάσεις μπορεί να ορίζονται περισσότεροι από ένας δομητές αλλά μόνο ένας από αυτούς θα είναι ο προκαθορισμένος (default)

Constructor (Δομητής)

```
class Date
{
public:
    Date();
    Date(int d, int m, int y);
    Date(String);
    // ...
};

Date::Date(int d, int m, int y)
: day(d),
  month(m),
  year(y)
{ if (year == 0) year = current year;
```

Destructor (αποδομητής)

- ✓ Όταν δημιουργούμε ένα constructor, δημιουργούμε και ένα Destructor (αποδομητής)
- ✓ Δεν έχει παραμέτρους και δεν επιστρέφει τίποτα
- ✓ Πρέπει να αποδεσμεύσει τη δεσμευμένη μνήμη memory και να καθαρίσει την κλάση
- ✓ Ο destructor έχει τη μορφή:
~ClassName() {...}

Παράδειγμα/1.1

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

Παράδειγμα/1.2

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

Παράδειγμα/2.1

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

Παράδειγμα/2.2

```
int main () {  
    Rectangle rect, rectb;  
    rect.set_values (3,4);  
    rectb.set_values (5,6);  
    cout << "rect area: " << rect.area() << endl;  
    cout << "rectb area: " << rectb.area() << endl;  
    return 0;  
}
```

Παράδειγμα/3.1

```
#include <iostream>
class Rectangle {
    int width, height;
public:
    Rectangle (int,int);
    int area () {return (width*height);}
};
Rectangle::Rectangle (int a, int b) {
    width = a;
    height = b;
}
```

Παράδειγμα/3.2

```
int main () {  
    Rectangle rect (3,4);  
    Rectangle rectb (5,6);  
    cout << "rect area: " << rect.area() << endl;  
    cout << "rectb area: " << rectb.area() << endl;  
    return 0;  
}
```