

Εργαστήριο μαθήματος

Οντοκεντρικός Προγραμματισμός Ι

(γλώσσα Java)

```
<% java
while(wake == true){
    System.out.println("I'm not crazy");
}

if(request.getParameter("age") < 18){
    StringBuffer sb = new StringBuffer();
    sb.append("Sorry ");
    sb.append(request.getParameter("name"));
    System.out.println(sb.toString());
    System.out.println("Bring an extra $500"); break;
}
"Amsterdam":
System.out.println("Bring an open mind"); break;
"Tokyo":
System.out.println("Bring lots of money"); break;
"Caribbean Islands":
System.out.println("Bring a swimsuit"); break;
}

while(!weekend){
    jumpToNextDay();
}

import java.util.*;
for(long j=1; j<1000000000L; j++){
    System.out.println("Repeat... I'm not crazy");
}
DELETE FROM boss;
```

Σιαφλέκης Αναστάσιος

A.M.4634

Έτος 7ο

siaflekis@ceid.upatras.gr

Εργαστηριακές Ασκήσεις

Άσκηση 1^η

3) Δίνοντας το όρισμα 25 εμφανίζεται στην οθόνη το μήνυμα:

```
You are  
25  
years old.
```

Αυτό συμβαίνει γιατί μέσα στο πρόγραμμα υπάρχει η εντολή εκτύπωσης `System.out.println("...")`, που εμφανίζει το μήνυμα που έχει ως όρισμα κι έπειτα αλλάζει γραμμή.

Επαναλαμβάνοντας την ίδια διαδικασία άλλες δυο φορές παίρνουμε τα παρακάτω:

```
You are  
30  
years old.
```

και

```
You are  
15  
years old.
```

Όταν δεν δοθεί ηλικία, δηλαδή δεν δοθεί κανένα όρισμα η οθόνη εκτύπωσης δεν εμφανίζεται, γιατί στο πρόγραμμα δεν υπάρχουν εντολές να εκτελεστούν όταν δεν δίνονται ορίσματα. Άρα το πρόγραμμα δεν εκτελεί τίποτα.

4) Αλλάζουμε τις εντολές εκτύπωσης όπως μας δίνονται. Η διαφορά είναι ότι στο τέλος των εντολών δεν υπάρχει το `ln` (εκτός απ' την 3^η), επομένως περιμένουμε το πρόγραμμα να εμφανίσει το αποτέλεσμα σε μία γραμμή. Όντως, δίνοντας όρισμα «12» παίρνουμε το παρακάτω μήνυμα:

```
You are 12 years old.
```

5) Προσθέσαμε το τμήμα που μας δίνεται στον κώδικά μας. Πράγματι με την εκτέλεση του προγράμματος χωρίς να δοθούν ορίσματα εμφανίζεται στην οθόνη το ακόλουθο μήνυμα:

You forgot to give your age.

6) Αντικαθιστούμε τον κώδικα όπως μας ζητείται και δίνουμε δυο ορίσματα(ηλικία και όνομα), οπότε το πρόγραμμα μας εκτυπώνει τα παρακάτω:

Kostas is 20 years old.

Αν δώσουμε σαν όρισμα μόνο την ηλικία μας εκτυπώνει:

You are 20 years old.

Τέλος αν δεν δώσουμε κανένα όρισμα μας εκτυπώνει:

You forgot to give your age.

7) Για να δίνουμε πρώτα το όνομα και μετά την ηλικία διαμορφώνουμε το σώμα της main ως εξής:

```
if (args.length == 1)
{
    System.out.print(args[0]);
    System.out.println(" you forgot to give your age");
}
else if (args.length == 0)
{
    System.out.println("You forgot to give your name and age.");
}
else
{
    System.out.print(args[0]);
    System.out.print(" is ");
    System.out.print(args[1]);
    System.out.println(" years old.");
}
```

Έτσι τα ορίσματα δίνονται ανάποδα, δηλαδή πρώτα το όνομα και μετά η ηλικία.

8)Αντικαθιστούμε σύμφωνα με τις υποδείξεις τις εντολές εκτύπωσης μηνυμάτων. Για όλες τις περιπτώσεις το πρόγραμμά μας δουλεύει όπως πριν. Όμως αν δεν κάνουμε clear στην οθόνη εμφάνισης μηνυμάτων, όλα θα εκτυπωθούν στην ίδια γραμμή, αφού από τις εντολές μας λείπει το `ln` που αλλάζει γραμμή μετά την εκτύπωση κάθε μηνύματος.

Άσκηση 2^η

3)Η οθόνη όταν δεν δίνουμε κανένα όρισμα εκτυπώνει τα παρακάτω:

The size of window w1 is: 5

The size of window w2 is: 4

The size of window w3 is: 6

The size of window w4 is: 5

4)Χρησιμοποιώντας την εντολή `this.size=x`; αντί της `size= x`; στο δεύτερο δημιουργό, παίρνουμε τα παρακάτω:

The size of window w1 is: 5

The size of window w2 is: 4

The size of window w3 is: 6

The size of window w4 is: 5

Τα αποτελέσματα είναι ακριβώς ίδια με πριν. Όσον αφορά την αλλαγή, η λέξη `this` αναφέρεται στο τρέχον αντικείμενο κάθε κλάσης και μπαίνει στην θέση του ονόματος του αντικειμένου. Μπορεί να χρησιμοποιηθεί οπουδήποτε καλείται ένα αντικείμενο δηλαδή στο προσδιορισμό μεταβλητής και μεθόδου, σαν όρισμα και σαν αντικείμενο επιστροφής από μια μέθοδο. Παρά την αλλαγή στον κώδικα, το αποτέλεσμα θα είναι το ίδιο διότι η μέθοδος **`getSize()`** είναι μέθοδος αντικειμένου και ήδη αναφέρεται στο τρέχον αντικείμενο. Αφού δεν ορίζεται στο σώμα της μεθόδου άλλη μεταβλητή με το όνομα `x` το **`this`** μπορεί να παραληφθεί, καθώς είναι μεταβλητή αντικειμένου αναφέρεται δηλαδή στο εκάστοτε αντικείμενο. Άρα, η εισαγωγή του `this` σε αυτή την περίπτωση δεν έπαιξε ρόλο.

5)Τροποποιούμε το πρόγραμμα όπως μας ζητείται. Οπότε παίρνουμε τα παρακάτω αποτελέσματα:

The size of window w1 is: 10

The size of window w2 is: 10

Σ' αυτή την περίπτωση έχουμε μόνο δύο στιγμιότυπα τα `w1` & `w2`. Επίσης στη μέθοδο `getSize()` δηλώνεται ως `size` το 10. Κάτι που σημαίνει πως κάθε φορά που καλείται έχει την τιμή 10 ανεξάρτητα από τα ορίσματα των στιγμιότυπων της `main`.

6)Τροποποιούμε το πρόγραμμα όπως μας ζητείται, παίρνοντας τα παρακάτω:

The size of window w1 is: 5

The size of window w2 is: 4

Γίνεται σαφές τώρα ότι με τη χρήση της `this` χρησιμοποιήθηκαν οι τιμές του `private int size=5` και η δήλωση ως όρισμα του 4 για το `size` στη `main`. Επαναλαμβάνουμε ότι η λέξη `this` αναφέρεται στο τρέχον αντικείμενο κάθε κλάσης και μπαίνει στην θέση του ονόματος του αντικειμένου. Μπορεί να χρησιμοποιηθεί οπουδήποτε καλείται ένα αντικείμενο δηλαδή στο προσδιορισμό μεταβλητής και μεθόδου, σαν όρισμα και σαν αντικείμενο επιστροφής από μια μέθοδο.

Στο παράδειγμα αυτό, στη μέθοδο χρησιμοποιείται η λέξη `this`, η οποία μπορεί να κληθεί μόνο μέσα σε μέθοδο και παράγει αναφορά στο αντικείμενο για το οποίο κλήθηκε η μέθοδος. Άρα, η εισαγωγή του `this` σε αυτή την περίπτωση έπαιξε ρόλο.

Άσκηση 3^η

2)α)Μεταφράζουμε και τρέχουμε την κλάση, παίρνοντας τα παρακάτω αποτελέσματα:

Window size=1

Window size=2

1

2

Αυτό συμβαίνει διότι επιτελείται η εξής διαδικασία: Η `RunWindow` είναι κλάση που περιέχει τη `main` και από εκεί τρέχει το πρόγραμμα. Έτσι λοιπόν, δημιουργούνται 2 νέα στιγμιότυπα τα `w1` και `w2` τα οποία παίρνουν τιμές από τη `Window`, η οποία δέχεται ως όρισμα μια ακέραια μεταβλητή τύπου `size`. Το `w1` δεν στέλνει κάποιο όρισμα κι επομένως το `size` θα έχει την τιμή που έχει στην κλάση `Window`, δηλαδή 1. Άρα `w1=1` αφού παίρνει την τιμή του. Το `w2` στέλνει ως όρισμα το 2 και επικρατεί της τιμής του `size` που δηλώνεται στην κλάση από τον δημιουργό. Άρα τελικά στο `w2` στέλνεται η τιμή 2. Παράλληλα, εκτυπώνονται και τα δύο στην οθόνη. Αφού επιστρέψουμε στη `main` εκτυπώνονται και οι τιμές των στιγμιότυπων στην οθόνη δηλαδή τα 1 και 2 και όλα αυτά σε καινούργια γραμμή γιατί στη `System.out` υπάρχει το `ln`.

2)β)Κάνουμε τις τροποποιήσεις και παίρνουμε τα παρακάτω αποτελέσματα:

Window size=1

Window size=2

1

2

Window size=1

Window message = First MWindow

Window size=3

Window message = Second MWindow

2

3

Τα 4 πρώτα αποτελέσματα είναι τα ίδια με προηγουμένως και προκύπτουν με τον ίδιο ακριβώς τρόπο αφού δεν μεταβλήθηκαν οι εντολές τους. Τα επόμενα προκύπτουν ως εξής: Δημιουργούνται 2 νέα στιγμιότυπα mw1 και mw2. Όσον αφορά, το mw1 καλεί την πρώτη MWindow στέλνοντας ως όρισμα μόνο ένα string. Όμως η MWindow κληρονομεί τη Window και άρα τα χαρακτηριστικά της. Επομένως η τιμή του size αρχικά είναι 1 και αυτό εκτυπώνεται πρώτα στην οθόνη. Μετά εκτυπώνεται το μήνυμα Window message = First MWindow της δεύτερης κλάσης. Όσον αφορά το mw2 καλεί τη δεύτερη MWindow και με τη χρήση της super κληρονομείται η τιμή του size από την ανώτερη κλάση(υπερκλάση) η οποία είναι 1 (αρχικοποιημένη). Όμως επικρατεί η τιμή του size που δόθηκε ως όρισμα και στο message επικρατεί η τιμή του message που στάλθηκε. Επομένως, εκτυπώνεται στην οθόνη τα μηνύματα Window size=3 και Window message = Second MWindow όπου Second MWindow το όρισμα string που δόθηκε. Τέλος, εκτυπώνονται τα εκάστοτε size των mw1 και mw2 (2 για το mw1 επειδή εκτυπώνει το size της πρώτης MWindow αφού αναφέρεται σε αυτήν και 3 για το mw2 επειδή είναι η τιμή για το size η οποία επικρατεί των υπολοίπων τιμών).

2)γ) Ύστερα κι απ τη νέα τροποποίηση έχουμε τα παρακάτω:

Window size=1

Window size=2

1

2

Window size=1

Window message = First MWindow

Window size=3

Window message = Second MWindow

2

3

4

6

8

Τα 10 πρώτα αποτελέσματα συμπίπτουν με τα προηγούμενα γιατί δεν μεταβλήθηκαν οι εντολές του και επομένως η εξήγηση παραμένει η ίδια. Όσον αφορά τα τρία επόμενα, στην πρώτη περίπτωση καλείται η μέθοδος `setSize1` της κλάσης `MWindow` με βάση το `mw1` η οποία δέχεται ως όρισμα έναν ακέραιο και τον τοποθετεί στο `size` επομένως έπειτα στην οθόνη που εκτυπώνει το `size` εκτυπώνει το 4. Στη δεύτερη περίπτωση, καλείται η `setSize2` της κλάσης `MWindow` με όρισμα την τιμή 2 η οποία καλεί κληρονομικά την `setSize` της κλάσης `Window` έχοντας ως όρισμα το `z=2` με το οποίο αυξάνεται η προηγούμενη τιμή του `size` κατά 2 άρα $4+2=6$ το οποίο κατόπιν εμφανίζεται στην οθόνη. Στην τρίτη περίπτωση καλείται η μέθοδος `setSize` της κλάσης `Window` κατευθείαν και με την εντολή `size+=x` δίνοντας ως όρισμα το 2 αυξάνεται η προηγούμενη τιμή του `size` κατά 2 άρα $6+2=8$ το οποίο εκτυπώνεται στην οθόνη. Όλα αυτά εκτυπώνονται σε καινούργια γραμμή γιατί στις `System.out` υπάρχει το `ln`.

3)Υστερα κι απ αυτή την τροποποίηση παίρνουμε τα εξής αποτελέσματα:

```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
4
4
4
```

Τα πρώτα 9 αποτελέσματα καθώς και το 11ο συμπίπτουν με τα προηγούμενα και επομένως οι αιτιολογήσεις παραμένουν οι ίδιες. Έχουν διαφοροποιηθεί το 10ο από 3 σε 7, και τα δύο τελευταία από 6 και 8 σε 4 και 4. Το πρώτο δικαιολογείται από το ότι οι μεταβλητές με το όρισμα `protected` χρησιμοποιούνται στις κλάσεις τους, σε κληρονομικές κλάσεις και σε πακέτα επομένως αυτή τη φορά η τιμή του `size` που είναι 7 δεν αλλάζει από το όρισμα 3 που δίνεται στη `main` άρα στην οθόνη εμφανίζει το 7. Στις 2 επόμενες περιπτώσεις το αποτέλεσμα είναι 4 γιατί καλείται η `setSize` της κλάσης `Window`. Δηλαδή η τιμή του `size` δεν αλλάζει σε σχέση με το πρώτο όρισμα της `mw1` που στέλνει στην `setSize1`.

4)Κάνοντας τις αλλαγές που μας υποδεικνύονται έχουμε τα παρακάτω αποτελέσματα:

Window size=1

Window size=2

1

2

Window size=1

Window message = First MWindow

Window size=3

Window message = Second MWindow

2

7

4

4

4

Size=1

MSize=4

5

Size=5

Για τα 13 πρώτα αποτελέσματα που συμπίπτουν με αυτά του προηγούμενου ερωτήματος ισχύουν όσα αναφέρθηκαν και προηγουμένως. Με την προσθήκη των εντολών που δόθηκαν προέκυψαν τα εξής: Καλείται η μέθοδος `printSize` της κλάσης `Window` χωρίς όρισμα, με βάση το στιγμιότυπο `w1` και εκτυπώνεται `Size=1`. Δηλαδή η τιμή του `size` είναι αυτή με την οποία έχει αρχικοποιηθεί στην κλάση στην οποία ανήκει (`Window`) και η οποία κλήθηκε σε αυτή την περίπτωση. Καλείται η μέθοδος `printSize` της κλάσης `MWindow` χωρίς όρισμα με βάση το στιγμιότυπο `mw1` και εκτυπώνεται `MSize=4` δηλαδή η τιμή του `size` είναι αυτή η οποία έχει δοθεί ως όρισμα στην μέθοδο `setSize` με βάση το `mw1` το οποίο αναφέρεται στη συγκεκριμένη κλάση. Καλείται η μέθοδος `printSize1` της κλάσης `MWindow` χωρίς όρισμα με βάση το στιγμιότυπο `mw1` και εκτυπώνεται 5 δηλαδή το `size` κληρονομεί την προηγούμενη τιμή 1 που έχει αρχικοποιηθεί στη κλάση `Window` αυξημένη κατά 4 που είναι η τιμή του `size` που δίνεται από το `mw1` δηλαδή $4+1=5$, το οποίο και εκτυπώνεται στην οθόνη. Καλείται η μέθοδος `printSize2` της κλάσης `MWindow` χωρίς όρισμα με βάση το στιγμιότυπο `mw1` και εκτυπώνεται `Size=5` δηλαδή το `size` κληρονομεί την μέθοδο `printSize` της κλάσης `Window` έχοντας ως τιμή την προηγούμενη δηλαδή 5 και έτσι εκτυπώνεται στην οθόνη.

5)Κάνουμε τις ζητούμενες αλλαγές, οπότε παίρνουμε τα παρακάτω αποτελέσματα:

```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
4
4
4
Size=1
MSize=4
5
MSize=4
```

Βλέπουμε ότι διαφοροποιούνται μόνο στο τελευταίο αποτέλεσμα όπου αντί για Size=5 εκτυπώνεται MSize=4 (η αλλαγή της printSize1 δεν διαφοροποίησε το αποτέλεσμα της). Αυτό συμβαίνει διότι με την αλλαγή της printSize2 της MWindow, καλείται η μέθοδος της κλάσης MWindow και όχι της Window με την οποία έχουν το ίδιο όνομα με όρισμα 4 του στιγμιότυπου mw1. Λόγω της χρήσης της λέξης this η οποία επιλέγει τη μέθοδο της ίδιας της κλάσης στην οποία βρισκόμαστε και παράγει αναφορά στο αντικείμενο για το οποίο κλήθηκε η μέθοδος.

1^ο Σετ Ασκήσεων

Μέρος Α

1→C,E

2→C

3→D

4→B

5→A

6→C

Μέρος Β

1) Η μέθοδος `calculate` χρησιμοποιεί `width`, `height`, `length` για να υπολογίσει τον όγκο ενός ορθογώνιου παραλληλογράμμου κι επιστρέφει το αποτέλεσμα διαιρεμένο με το δύο.

2) Ο δημιουργός είναι ο εξής:

```
Box (double height, double width, double length)
```

```
{  
    this.height= height;  
    this.width= width;  
    this.length= length;  
}
```

3) Οι μέθοδοι που ζητούνται είναι οι εξής:

```
double epifaneia()  
{  
    return ((length * width)+ (length * height) + (width * height) ) * 2;  
}  
double ogkos()  
{  
    return calculate() * 2;  
}
```

4) Παρατηρούμε ότι δεν μπορούμε να δημιουργήσουμε μία δική μας μέθοδο με το όνομα toString(), καθώς αυτή είναι ήδη δηλωμένη στη βιβλιοθήκη Java.lang.Object.

5) Η μέθοδος toString είναι:

```
public String toString()
{
    return "Epifaneia: " + epifaneia() + ", Ogkos " + ogkos();
}
```

6) Η κλάση MyBox είναι η παρακάτω:

```
MyBox(double myheight, double mywidth, double mylength)
{
    super(myheight,mywidth,mylength);
    elegxos();
}
//Methods
void elegxos()
{
    if ((height==width && height==length && width==length) && height>0 && width>0 && length>0){}
    else System.out.println("Lathos arxikopoihsh kyvou");
}
```

7) Οι μέθοδοι ogkos κι epifaneia της MyBox:

```
double epifaneia()
{
    return super.epifaneia();
}
double ogkos()
{
    return super.ogkos();
}
```

8) Η toString() της MyBox είναι:

```
public String toString()
{
    return "Cube: " + super.toString();
}
```

9), 10), 11) Ο κώδικας της κλάσης Spirtokouto που μας ζητείται είναι ο παρακάτω:

```
class spirtokouto extends Box
{
    double weight;
    double sheight;
    double swidth;
    double slength;
    double w;

    spirtokouto (double weight, double sheight, double swidth, double slength)
    {
        super(swidth,sheight,slength);
        w= weight;
    }

    //Methods
    public String toString()
    {
        return "Matchbox: " + super.toString()+ ", Varos: "+w;
    }
}
```

12) Η κλάση RunMe φαίνεται παρακάτω:

```
class RunMe
{
    public static void main(String args[])
    {
        Box box1= new Box (2.0,3.5,5.0);
        MyBox box2= new MyBox (4.0,4.0,4.0);
        spirtokouto box3= new spirtokouto(16.0,2.0,4.0,3.0);

        System.out.println(box1.toString());
        System.out.println(box2.toString());
        System.out.println(box3.toString());
    }
}
```