

## Οντοκεντρικός Προγραμματισμός I (Java)

Ακαδ. Έτος: 2011-2012

ΕΡΓΑΣΤΗΡΙΟ JAVA

Μονοπάτης Δημήτριος

A.M.:4776

### ΑΣΚΗΣΗ 4

1) Ως abstract ορίζονται (υπερ)κλάσεις οι οποίες θα έχουν abstract μεθόδους οι οποίες δε μπορούν να οριστούν στη συγκεκριμένη υπερκλάση αλλά θα πρέπει να καθοριστούν στην εκάστοτε υποκλάση ξεχωριστά, αφού η ίδια μέθοδος δεν είναι κατάλληλη για όλες τις υποκλάσεις που προκύπτουν από τη συγκεκριμένη κλάση. Στη συγκεκριμένη κλάση δε χρειαζόταν το abstract.

Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married.

Betty Tront is 31 years old, gets a 980.5 Euros salary and is married with 3 children.

2) public void printInfo()

```
{  
    System.out.print(getFirstName()+" "+getLastName()+" is "  
    +getAge()+" years old, gets a "+getSalary()  
    +" Euros salary and is");  
    if (isMarried() == false)  
        System.out.print(" not");  
    System.out.println(" married.");  
}
```

Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married.

Betty Tront is 31 years old, gets a 980.5 Euros salary and is married.

Ως πλεονέκτημα υπάρχει το ότι η εκτύπωση των στοιχείων του κάθε person υπάρχει διαμορφωμένη από πριν και δε χρειάζεται να ξαναγράφεται κώδικας κάθε φορά που κάποια άλλη κλάση θέλει να εκτυπώσει τα στοιχεία του κάθε person. Ακόμα, ο κώδικας στη κλάση Mytester έχει γίνει πολύ πιο μικρός.

Αυτό που λείπει είναι ο αριθμός των παιδιών, που είναι ιδιότητα του MarriedPerson και επειδή το printInfo καλείται και στις δύο περιπτώσεις στο Person αφού δεν υπάρχει άλλη εκδοχή του στην κλάση MarriedPerson, δεν υπάρχει δυνατότητα να εκτυπωθεί ο αριθμός των παιδιών.

3) public void printInfo()

```
{  
    super.printInfo();  
    System.out.print("He/She has ");  
    if (getNoOfChildren() > 0) System.out.print(getNoOfChildren());  
    else System.out.print("no");  
    System.out.println(" children.");  
}
```

Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married.

Betty Tront is 31 years old, gets a 980.5 Euros salary and is married.

He/She has 3 children.

4) Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married.  
Betty Tront is 31 years old, gets a 980.5 Euros salary and is married.  
He/She has 3 children.

Αρχικά δεν εκτυπώνεται το φύλο αφού δεν υπάρχει κάποια εντολή για αυτό.  
Διορθώνεται ως εξής:

```
public void printInfo()
{
    System.out.print(getFirstName()+" "+getLastName()+" is "
        +getAge()+" years old ");
    if (sex == MALE) System.out.print("male");
    else System.out.print("female");
    System.out.print(", gets a "+getSalary()
        +" Euros salary and is");
    if (isMarried() == false)
        System.out.print(" not");
    System.out.println(" married.");
}
```

Paul Kings is 22 years old male, gets a 1200.0 Euros salary and is married.  
Betty Tront is 31 years old female, gets a 980.5 Euros salary and is married.  
He/She has 3 children.

5)  
α,β)

```
public void setLastName(String lastname) {this.lastname = lastname;}
public void setFirstName(String firstname) {this.firstname = firstname;}
public void setAge(int age) {this.age = age;}
public void setMarried(boolean married) {this.married = married;}
public void setSalary(float salary) {this.salary = salary;}
```

```
public void setNoOfChilderen(int children) {this.children = children;}
```

Paul Kings is 22 years old male, gets a 1200.0 Euros salary and is not married.  
Betty Tront is 31 years old female, gets a 980.5 Euros salary and is not married.  
He/She has 3 children.  
Is mp1 married?: false

Πράγματι ενώ και τα δύο άτομα ορίζονται ως μη παντρεμένα, με τις νέες  
μεθόδους μπορούμε και εκ των υστέρων αλλάζουμε αυτή την ιδιότητα.

γ)

```
public void setMarried(boolean married) {};
```

Paul Kings is 22 years old male, gets a 1200.0 Euros salary and is not married.  
Betty Tront is 31 years old female, gets a 980.5 Euros salary and is married.  
He/She has 3 children.  
Is mp1 married?: true

Απλά φτιάχνουμε την ίδια μέθοδο και στην MarriedPerson, χωρίς εκτελούμενο κώδικα, έτσι όποτε κάποιος πάει να αλλάξει την ιδιότητα Married του MarriedPerson αυτή δε θα αλλάζει.

6)

```
public int getSex() { return sex; }
public void setSex(int Sex) {this.sex = Sex;}

public void setSalary(MarriedPerson spouse)
{
    if (spouse.getSex() != this.getSex())
    {this.setSalary(this.getSalary() +
    spouse.getSalary());}
}
```

Betty Tront is 31 years old female, gets a 980.5 Euros salary and is married.  
He/She has 3 children.

Kirk Tront is 31 years old male, gets a 2080.0 Euros salary and is married.  
He/She has 2 children.

Sonia Tront is 31 years old female, gets a 600.0 Euros salary and is married.  
He/She has no children.

Betty Tront is 31 years old female, gets a 3060.5 Euros salary and is married.  
He/She has 3 children.

Betty Tront is 31 years old female, gets a 3060.5 Euros salary and is married.  
He/She has 3 children.

Betty Tront is 31 years old female, gets a 2080.0 Euros salary and is married.  
He/She has 3 children.

Αφού δημιουργήσουμε τα 3 MarriedPerson, πρώτα βάζουμε τον mp2 ως spouse της mp1, και άρα ο μισθός του προστίθεται στο μισθό της αφού είναι αντίθετο το φύλο τους. Ύστερα προσπαθούμε να βάλουμε την mp3 ως spouse στην mp1 αλλά δεν αλλάζει κάτι αφού είναι του ίδιου φύλου. Τέλος, δίνουμε ως παράμετρο στο setSalary το Salary του mp2, το οποίο προκαλεί κλήση της setSalary της Person, αφού αυτή είναι που παίρνει ως παράμετρο int Salary.

Η σχέση μεταξύ των δύο μεθόδων είναι ότι η καθεμία καλείται ανάλογα με την παράμετρο που της περνάμε. Αν περάσουμε ένα MarriedPerson καλείται η setSalary της MarriedPerson, ενώ αν περάσουμε int Salary καλείται η setSalary της Person.

## 2ο ΣΕΤ ΑΣΚΗΣΕΩΝ

1)

```
public abstract class Employee
{
    private long EmpID;
    private static long aa;
    private String name = "";
    private String afm = "";
```

```

public Employee()
{
    aa = aa + 1;
    this.EmpID = aa;
}

public void setName(String name) {this.name = name;}
public String getName() {return name;}
public void setAfm(String a) {this.afm = a;}
public String getAfm() {return afm;}
public long getEmpID() {return EmpID;}
abstract int payment();
}

public class SalariedEmployee extends Employee
{
    private int salary = 0;

    public SalariedEmployee()
    {
    }

    public void setSalary(int s) {salary = s;}
    public int payment() {return salary;}
}

public class HourlyEmployee extends Employee
{
    private int hoursWorked = 0;
    private int hourlyPayment = 0;

    public HourlyEmployee ()
    {
    }

    public void setHoursWorked(int hw) {hoursWorked = hw;}
    public int getHoursWorked() {return hoursWorked;}
    public void setHourlyPayment(int hp) {hourlyPayment = hp;}
    public int getHourlyPayment() {return hourlyPayment;}
    public int payment() {return hoursWorked * hourlyPayment;}
}

public class Main
{
    public static void main(String args[])
    {
        Employee e[] = new Employee[2];
        e[0] = new SalariedEmployee();
        e[1] = new HourlyEmployee();
        e[0].setName("Gewrgiou");
        e[0].setAfm("7777777");
    }
}

```

```

((SalariedEmployee)e[0]).setSalary(300000);
e[1].setName("Karamitros");
e[1].setAfm("88888888");
((HourlyEmployee)e[1]).setHoursWorked(3000);
((HourlyEmployee)e[1]).setHourlyPayment(40);

for (int i=0; i<2; i++) {
    System.out.println ("Employee ID:" + e[i].getEmpID());
    System.out.println ("Employee name:" + e[i].getName());
    System.out.println ("Employee AFM:" + e[i].getAfm());
    System.out.println ("Employee salary:" + e[i].payment());
    System.out.println(" ");
}
}
}

```

2)

```
import java.io.*;
```

```
public class Divide
```

```

{
    public static void main(String args[])
    {
        System.out.println("Dwste ton 1o akeraio:");
        int a = readInt();
        System.out.println("Dwste ton 2o akeraio:");
        int b = readInt();
        try {
            System.out.println(a + "/" + b + "=" + a/b);
        } catch(ArithmeticException ae) {
            System.out.println("Divider can't be zero.");
        }
    }
}

public static int readInt() {
    byte b[] = new byte[16];
    String str;
    try {
        System.in.read(b);
        str = (new String(b)).trim();
        return Integer.parseInt(str);
    } catch (IOException e) {
        System.out.println("Exception: " + e.toString());
        return 0;
    } catch (NumberFormatException e) {
        System.out.println("Exception: " + e.toString() + "\nReturned value: -1");
        return -1;
    }
}
}

```

3)

```
import java.io.*;

public class Main
{
    public static void main(String args[])
    {
        int a,p=0;
        try{
            File f1 = new File("src.txt");
            boolean find = f1.exists();
            FileInputStream fis = new FileInputStream("src.txt");
            if (find){
                System.out.println("To arxeio uparxei!");}
            }
            catch (FileNotFoundException e) {
                System.out.println("File not found");}

        try {
            File f1 = new File("src.txt");
            File f2 = new File("dest.txt");

            FileReader fin = new FileReader(f1);
            FileWriter fout = new FileWriter(f2);

            System.out.println("Prin");
            System.out.println("-----");
            System.out.println("Megethos Source = " + f1.length());
            System.out.println("Megethos Dest = " + f2.length());

            do
            {
                a = fin.read();
                if (a != -1)
                {
                    if ((char)a == ' ')
                    {
                        a = (int)'@';
                        p++;
                    }
                    fout.write(a);
                }
            } while (a != -1);
            fin.close();
            fout.close();
            System.out.println("-----");
            System.out.println("Meta");
            System.out.println("-----");
            System.out.println("Megethos Source = " + f1.length());
            System.out.println("Megethos Dest = " + f2.length());
            System.out.println("Pli8os kenwn pou antikatasta8ikan: " + p);
```

```
    } catch(FileNotFoundException fnfe) {  
        System.exit(0);  
    } catch(IOException ioe) {  
        System.out.println("IO error.");  
        System.exit(0);  
    }  
}  
}
```