

Προγραμματισμός II

8/4/2020

Σύνθετες κλάσεις και μέθοδοι

Κλάσεις με μεταβλητές υπόστασης τύπου κλάσης

Στην κλάση Human, η μεταβλητή name είναι τύπου κλάσης (String). Δεν απαιτούσε κάποιο ιδιαίτερο χειρισμό, γιατί η δημιουργία ενός αντικειμένου τύπου String δεν απαιτεί τη new:

```
class Human
{
    String name;
    int height;

    Human (String n, int h)
    {
        name=n;
        height=h;
    }

    public String toString()
    {
        return name+" "+height;
    }
}
```

Κλάση ευθείας με μεταβλητές υπόστασης τύπου σημείου

```
class Point
{
    int x,y;

    Point(int x, int y)
    {
        this.x=x;
        this.y=y;
    }

    public String toString()
    {
        return "(" + x + "," + y + ")";
    }
}
```

Κλάση ευθείας με μεταβλητές υπόστασης τύπου σημείου

```
class StraightLine
{
    //Instance variables of class type Point
    private Point a,b;

    //Constructor based on coordinates
    public StraightLine(int x1,int y1,int x2,int y2)
    {
        this.a=new Point(x1,y1);
        this.b=new Point(x2,y2);
    }

    //Constructor based on point objects
    public StraightLine (Point a, Point b)
    {
        this.a=new Point(a.x,a.y)
        this.b=new Point(b.x,b.y)
    }
}
```

Κλάση ευθείας με μεταβλητές υπόστασης τύπου σημείου

```
//Calculate the length of the line
public double length()
{
    //Returns the Euclidean distance of the points
    double d1=a.x-b.x;
    double d2=a.y-b.y;
    return Math.sqrt(d1*d1+d2*d2);
}

//Accessor method – returns an independent point
public Point getPointA()
{
    Point first=new Point(a.x,a.y);
    return first;

    //if you use return a; the point will be dependent
}
```

Κλάση ευθείας με μεταβλητές υπόστασης τύπου σημείου

```
//Accessor method-returns an independent point
```

```
public Point getPointB()
```

```
{
```

```
    Point second=new Point(b.x,b.y);
```

```
    return second;
```

```
}
```

```
// Constructs a new Straightline with inverted points
```

```
public StraightLine inverted()
```

```
{
```

```
    StraightLine r=new StraightLine(b,a);
```

```
    return r;
```

```
}
```

Κλάση ευθείας με μεταβλητές υπόστασης τύπου σημείου

```
public String toString()
{
    return "Line from " + a.toString() + " to " + b.toString();
}
```

```
public static void main(String[] args)
{
    //Construct a line from coordinates
    StraightLine e1=new StraightLine(1,5,8,2);

    //Construct a line from two point objects
    Point p1=new Point(2,3);
    Point p2=new Point(5,6);
    StraightLine e2=new StraightLine(p1,p2);
}
```


Κλάση ευθείας με μεταβλητές υπόστασης τύπου σημείου

```
//Get the length of a line  
System.out.println("The first line has a length of " +e1.length()  
+ " points");
```

```
//Get the first point  
Point c=e1.getPointA();
```

```
//Get an inverted straight line  
StraightLine e3=e2.inverted();
```

```
}
```

```
}
```

ΑΡΙ, ΙΔΕ, ΤΕΚΜΗΡΙΩΣΗ ΚΑΙ ΔΙΑΝΟΜΗ
ΛΟΓΙΣΜΙΚΟΥ

Χρήση πακέτων

- Τα πακέτα είναι οργανωμένες συλλογές (βιβλιοθήκες) κλάσεων
- Για να προσθέσουμε μια κλάση από ένα πακέτο χρησιμοποιούμε την `import`:

```
import java.util.Random;
```

- Για να προσθέσουμε όλες τις κλάσεις ενός πακέτου:

```
import java.util.*;
```

Οι έμπειροι προγραμματιστές συνήθως επιλέγουν πολλές φορές τον πρώτο τρόπο για να γνωρίζουν τι έχουν εισάγει

Παιχνίδι

```
import java.util.Random;  
import java.util.Scanner;
```

```
class GuessGame  
{
```

```
    public static void main(String[] args)  
    {
```

```
        int score=0;
```

```
        for (int i=0;i<10;i++)
```

```
        {
```

```
            //Ask for a number between 0-10
```

```
            System.out.print("Number? ");
```

```
            Scanner s=new Scanner(System.in);
```

```
            //Generate a random number between 0-10
```

```
            Random r=new Random();
```

```
            int guess=r.nextInt(10);
```

Παιχνίδι

```
        //if (input==guess)
        {
            score++;
            System.out.println("You won!");
        } else
        {
            System.out.println("You lost.");
        }
    }
    System.out.println("Your score is " + score + "/10.");
}
}
```

Κατασκευή τεκμηρίωσης API

- Η εφαρμογή **javadoc** κατασκευάζει αυτόματη τεκμηρίωση στις κλάσεις μας
- Το επιτυγχάνει διαβάζοντας τα σχόλια που έχουμε εισάγει στον κώδικα με ειδικό τρόπο
- Στα σχόλια χρησιμοποιούμε κάποια σύμβολα και ετικέτες (tags) που σημειώνονται με '@'

Κατασκευή τεκμηρίωσης API

Σύμβολα	Χρήση
<code>/**</code>	Έναρξη σχολίων που αναγνωρίζει το Javadoc για την αυτόματη κατασκευή τεκμηρίωσης. Χρησιμοποιείται αντί του τυπικού <code>/*</code> .
<code>@param</code>	Μετά την ετικέτα αυτή ακολουθεί το όνομα και η περιγραφή μιας παραμέτρου μιας μεθόδου
<code>@return</code>	Με την ετικέτα αυτή ακολουθεί η περιγραφή αναφορικά με το τι επιστρέφεται από μία μέθοδο
<code>@author</code>	Ακολουθεί το όνομα του συγγραφέα (προγραμματιστή)
<code>@version</code>	Ακολουθεί η έκδοση του λογισμικού

Κατασκευή τεκμηρίωσης API

- Για να παραχθεί απλά η τεκμηρίωση σε ένα νέο φάκελο με όνομα `documentation`:

```
javadoc -d documentation GuessGame.java
```

- Για να προστεθούν οι παράμετροι `-author` και `-version`:

```
javadoc -author -version -d documentation GuessGame.java
```

- Για να εμφανιστούν επιπλέον παράμετροι (`help`), απλά

```
javadoc
```


Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
import java.util.Random;
import java.util.Scanner;

/**
GuessGame: A number guessing game.
@author Dimitris Iakovidis
@version 0.1
*/

public class GuessGame
{
    public static String
        MESSAGE_WON = "You won!\n",
        MESSAGE_LOST = "You lost\n",
        MESSAGE_SCORE = "Your score is ",
        MESSAGE_QUESTION = "Number? ";
```

Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
private int maxIterations,  
        maxRandom,  
        iteration,  
        score,  
        input;
```

```
/**
```

```
Constructs a guess game given a set of parameters
```

```
@param maxIterations The number of iterations is to run the game
```

```
@param maxRandom The maximum random number to guess
```

```
*/
```

```
public GuessGame(int maxIterations,int maxRandom)
```

```
{
```

```
    score=0;
```

```
    this.maxIterations=maxIterations;
```

```
    this.maxRandom=maxRandom;
```

```
}
```

Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
/**  
Constructs a guess game without a set of parameters  
*/  
public GuessGame()  
{  
    this(10,10);  
}  
  
/**  
Increases the score by one  
*/  
public void increaseScore()  
{  
    score++;  
}
```

Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
/**
Returns the current score
@return The current score
*/
public int getScore()
{
    return score;
}

/**
Returns the number of iterations that the game will run
@return The number of iterations that the game will run
*/
public int getMaxIterations()
{
    return maxIterations;
}
```

Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
/**  
Returns a random number between 1 and a maximum value maxRandom  
@return A random number between 1 and a maximum value maxRandom  
*/  
public int generateRandom()  
{  
    Random r=new Random();  
    int guess=r.nextInt(maxRandom);  
    return guess;  
}
```

Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
/**  
Asks the user a number from the command line  
@return The number entered by the user  
*/  
public int read()  
{  
    System.out.print(MESSAGE_QUESTION);  
    Scanner s=new Scanner(System.in);  
    return input;  
}
```

Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
/**  
Outputs a string to the command line  
@param s The string to be displayed in the output  
*/  
public void write(String s)  
{  
    System.out.print(s);  
}
```

Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
/**
Initiates a single question game play
*/
public void playSingleQuestionGame()
{
    for (int i=0;i<maxIterations;i++)
    {
        int input=read();
        int guess=generateRandom();

        //if the numbers match you win!
        if (input==guess)
        {
            increaseScore();
            write(MESSAGE_WON);
        } else
        {write(MESSAGE_LOST); } }
    }
```


Παιχνίδι (αντικειμενοστραφής έκδοση/documentation)

```
        write(MESSAGE_SCORE+score+"/"+"maxIterations+");
    }

    /**
    Executes a test game
    */
    public static void Main(String[] args)
    {
        GuessGame x=new GuessGame();
        x.playingSingleQuestionGame();
    }
}
```

Κατασκευή πακέτων

- Ένα πακέτο κατασκευάζεται όπως και μια οργάνωση αρχείων σε φακέλους
- Πχ ένα πακέτο animals με κλάσεις ζώα της γης (earth), του ουρανού (sky) και της θάλασσας (sea)
- Κατασκευάζουμε στην τρέχουσα διαδρομή του συστήματος αρχείων μας τους εξής φακέλους:

animals

animals/earth

animals/sky

animals/sea

Κατασκευή πακέτων

- Στον κάθε φάκελο τοποθετούμε τις κλάσεις με τα αντίστοιχα ζώα, πχ στον φάκελο earth τις κλάσεις Cat.java, Dog.java, στον sky την κλάση Eagle.java, στον sea την κλάση dolphin.java κλπ
- Στο αρχείο κάθε κλάσης υπάρχει μία δήλωση του πακέτου στο οποίο ανήκει. Πχ στην κλάση Cat.java υπάρχει η πρόταση:

package animals.earth;

- Οι κλάσεις που επιθυμούμε να προσπελούνται εξωτερικά από τους χρήστες κάθε πακέτου δηλώνονται ως public

Κατασκευή πακέτων

- Μεταγλωττίζουμε κάθε κλάση του πακέτου μας για να παραχθούν τα αντίστοιχα αρχεία τύπου class.
- Κατασκευάζουμε ένα αρχείο animals.jar ως εξής:

```
jar -cvf animals.jar animals
```

- Εισάγουμε την κλάση αυτή στα προγράμματά μας με:

```
import animals.earth.Cat;
```

- Θα πρέπει να προσθέσουμε τη διαδρομή προς το πακέτο animals.jar στη μεταβλητή **classpath**.

Κατασκευή πακέτων

SET CLASSPATH=.;C\Program

Files\jdk1.7.0_40\LIB\TOOLS.JAR;C:\myfolder\animals.jar

- Αν (όπως στο παραπάνω παράδειγμα) υπάρχει ο φάκελος “*.*” στην αρχή της πρότασης SET, αρκεί να διασφαλίσουμε ότι το πακέτο μας (animals.jar) βρίσκεται στον ίδιο φάκελο με το αρχείο java του προγράμματος που χρησιμοποιεί το πακέτο.

Κατασκευή πακέτων

```
package animals.earth;
```

```
public class Cat
```

```
{
```

```
    public String name;
```

```
    //Enables a cat to speak
```

```
    public void speak()
```

```
    {
```

```
        System.out.println(name+" says Miaou");
```

```
    }
```

```
    //Cat becomes executable
```

```
    public static void main(String[] args)
```

```
    {
```

```
        //Our cat's name is Goldy
```

```
        Cat g=new Cat();
```

```
        g.name="Goldy";
```

```
        g.speak;
```

```
    }
```

```
}
```