ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Αντικειμενοστρεφής Προγραμματισμός
Ασκήσεις
25/6/2024

Εαρινό Εξάμηνο 2024

Κούτσικας Χρήστος

Anonymous classes – Παράδειγμα abstract

```
abstract class AbClass {
    abstract void Method();
}

class Class1 extends AbClass {
    void Method() {
        System.out.println("Message from method");
    };
}

public class SpecialCases {
    public static void main(String[] args) {
      Class1 ob1 = new Class1();
      ob1.Method();
    }
}
```

Anonymous classes – Παράδειγμα abstract

```
abstract class AbClass {
    abstract void Method();
}

public class Anonymous {
   public static void main(String[] args) {
      new AbClass() {
         void Method() {System.out.println("Message from method");}
      }.Method();

      AbClass a = new AbClass() {
         void Method() {System.out.println("Message from method");}
      };
      a.Method();

   }
}
```

Anonymous classes – Παράδειγμα inteface

```
interface IClass {
    abstract void Method();
}

class Class1 implements IClass {
    public void Method() {
        System.out.println("Message from method");
    };
}

public class SpecialCases {
    public static void main(String[] args) {
      IClass obl = new Class1();
      obl.Method();
    }
}
```

Anonymous classes – Παράδειγμα inteface

```
interface IClass {
    abstract void Method();
}

public class Anonymous {
   public static void main(String[] args) {
      new IClass() {
         public void Method() {System.out.println("Message from method");}
      }.Method();

      IClass a = new IClass() {
         public void Method() {System.out.println("Message from method");}
      };
      a.Method();

   }
}
```

Εφαρμογές δικτύου

- Επίπεδο εφαρμογής (application layer)
  - HTTP, ftp, telnet, …
- Επίπεδο μεταφοράς (transport layer)
  - TCP/IP, UDP, …
- Επίπεδο δικτύου (network layer)
  - IP, …
- Επίπεδο συνδέσμου (link layer)
  - device driver

Εφαρμογές δικτύου – Ανάγνωση από ένα URL

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


public class URLRead {
    public static void main(String[] args) throws Exception {
      try {
          URL aueb = new URL("https://www.aueb.gr/");
          DataInputStream s = new DataInputStream(aueb.openStream());
          String inputLine;

          while ((inputLine = s.readLine()) != null)
             System.out.println(inputLine);
          s.close();
       }
       catch (IOException e) {
           System.err.println("I/O error: " + e);
       }
   }

}
```

Anonymous classes – Παράδειγμα inteface

```
run:
<!DOCTYPE html>
<html lang="el" dir="ltr">
<head>
<!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-NG594K3');</script>
<!-- End Google Tag Manager -->

<meta name="description" content="ÎOÎ¹Î°Î¿Î¼Î¿Î¼Î¹Î° Î Î±Î¼Î¹ÏOÎ¹ÏOÏOÎ®Î¼Î¹Î¿ ÎOÎ¸Î·Î½Î±, At
<meta name="keywords" content="ÎOÎ¹Î°Î¿Î¼Î¿Î¼Î¹Î° Î Î±Î¼Î¹ÏOÎ¹ÏOÏOÎ®Î¼Î¹Î¿ ÎOÎ¸Î·Î½Î±, Î£ÏOÎ¿
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="alternate" type="application/rss+xml" title="ÎOÎ¹Î°Î¿Î¼Î¿Î¼Î¹Î° Î Î±Î¼Î¹ÏOÎ¹ÏOÏOÎ®Î¼
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=10" />
<link rel="shortcut icon" href="https://www.aueb.gr/sites/default/files/hermis.png" type="image/p
<meta name="facebook-domain-verification" content="rwe3cazxdt2wpc066xvv7tu8lk5jom" />
<link type="text/css" rel="stylesheet" href="https://www.aueb.gr/sites/default/files/css/css_xE-r
<link type="text/css" rel="stylesheet" href="https://www.aueb.gr/sites/default/files/css/css_U9OE
<link type="text/css" rel="stylesheet" href="https://www.aueb.gr/sites/default/files/css/css_6xNx
<link type="text/css" rel="stylesheet" href="https://www.aueb.gr/sites/default/files/css/css_5tTi
```

Εφαρμογές δικτύου – Γράψιμο σε μια URL σύνδεση

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


public class SpecialCases {
    public static void main(String[] args) {
      try {
         //Δημιουργία του αντικειμένου URL
         URL aueb = new URL("https://www.aueb.gr/");
         //Δημιουργία της σύνδεσης
         URLConnection connection = aueb.openConnection();
         //Δημιουργία ρεύματος εξόδου
         PrintStream out = new PrintStream(connection.getOutputStream());
         //Εγγραφή στο ρεύμα εξόδου των δεδομένων που θέλουμε να στείλουμε
         out.println("searchFor="+"my test");
         //Κλείσιμο του ρεύματος
         out.close();
      }
      catch (IOException e) {
      }
    }
}
```

Εφαρμογές δικτύου – Κατέβασμα αρχείου

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


public class SpecialCases {
    public static void main(String[] args) throws Exception {
      URL page = new URL("https://www.aueb.gr/sites/default/files/Spring-Exams-23-24%2020240531.pdf");

      HttpURLConnection conn = (HttpURLConnection) page.openConnection();
      conn.connect();

      BufferedInputStream buff = new
            BufferedInputStream(conn.getInputStream());
      FileOutputStream out = new FileOutputStream("C:\\MAIN\\AuebProgram.pdf");

      byte data[]=new byte[1024];
      int count;
      while ((count=buff.read(data,0,1024)) !=-1) {
         out.write(data,0,count);
      }

      buff.close();
      out.close();
      conn.disconnect();
    }
}
```

Εφαρμογές δικτύου – Πρόγραμμα πελάτης

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

public class MyClient {
  public static void main(String[] args) {
    Socket server;
    try {
      server = new Socket("172.17.30.227", 25);
      System.out.println(server);
      InputStream in = server.getInputStream();
      OutputStream out = server.getOutputStream();

      out.write(12); //send a byte
      byte b = (byte)in.read(); //get a byte
      System.out.println(b);

      PrintStream pout = new PrintStream(out);
      DataInputStream din = new DataInputStream(in);

      pout.println("Hello!"); //send a string
      String response = din.readLine(); //get a string
      System.out.println(response);
    }
    catch (Exception e) {
        System.err.println(e.getMessage());
    }
  }
}
```

Εφαρμογές δικτύου – Πρόγραμμα εξυπηρετητής

```
public class MyServer {

  public static void main(String[] args) {
    ServerSocket listener=null;
    try {
      listener = new ServerSocket(25);
      while (true) {
          Socket aClient = listener.accept();
          System.out.println("Server");
          InputStream in = aClient.getInputStream();
          OutputStream out = aClient.getOutputStream();
          byte b = (byte)in.read(); //get a byte
          out.write(43); //send a byte

          DataInputStream din = new DataInputStream(in);
          PrintStream pout = new PrintStream(out);

          String response = din.readLine(); //get a string
          pout.println("Goodbye!"); //send a string

      }
    }
    catch (Exception e) {
        System.err.println(e.getMessage());
    }
    finally {
        try {
          listener.close();
        }
        catch (IOException e) {
          System.err.println(e.getMessage());
        }
    }
  }

}
```

Νήματα (threads)

```
class MyThread extends Thread{
    int name;

    public MyThread (int name) {
       this.name=name;
    }

    public void run(){
       System.out.println("My name is "+name);
    };
}

public class Anonymous {
   public static void main(String[] args) {
      for (int i=0; i<100; i++) {
         new MyThread(i).start();
      }
   }
}
```

```
run:
My name is 0
My name is 1
My name is 6
My name is 3
My name is 4
My name is 5
My name is 2
My name is 7
My name is 20
My name is 9
My name is 10
My name is 11
My name is 12
My name is 14
My name is 13
My name is 15
My name is 17
My name is 34
My name is 35
My name is 18
My name is 41
My name is 21
My name is 8
My name is 23
My name is 22
```

Εκφράσεις Lambda

Οι εκφράσεις Lambda ορίζουν στιγμιότυπα functional interfaces (interface με μια abstract μέθοδο.

- Μπορούμε να χειριστούμε μια μέθοδο σαν παράμετρο και τον κώδικα σαν δεδομένα.
- Μπορεί να δημιουργηθεί μια συνάρτηση χωρίς να ανήκει σε κάποια κλάση.
- Μια lambda έκφραση μπορεί να χρησιμοποιηθεί σαν να ήταν ένα αντικείμενο.

Εκφράσεις Lambda – Παράδειγμα

```java
// Java program to demonstrate lambda expressions
// to implement a user defined functional interface.

// A sample functional interface (An interface with
// single abstract method
interface FuncInterface
{
    // An abstract function
    void abstractFun(int x);

    // A non-abstract (or default) function
    default void normalFun()
    {
        System.out.println("Hello");
    }
}

class Test
{
    public static void main(String args[])
    {
        // lambda expression to implement above
        // functional interface. This interface
        // by default implements abstractFun()
        FuncInterface fobj = (int x)->System.out.println(2*x);

        // This calls above lambda expression and prints 10.
        fobj.abstractFun(5);
    }
}
```

Εκφράσεις Lambda – Παράδειγμα

```java
// A Java program to demonstrate simple lambda expressions
import java.util.ArrayList;
class Test {
    public static void main(String args[])
    {
        // Creating an ArrayList with elements
        // {1, 2, 3, 4}
        ArrayList<Integer> arrL = new ArrayList<Integer>();
        arrL.add(1);
        arrL.add(2);
        arrL.add(3);
        arrL.add(4);

        // Using lambda expression to print all elements
        // of arrL
        arrL.forEach(n -> System.out.println(n));

        // Using lambda expression to print even elements
        // of arrL
        arrL.forEach(n -> {
            if (n % 2 == 0)
                System.out.println(n);
        });
    }
}
```

Εκφράσεις Lambda – Παράδειγμα

```java
// Java program to demonstrate working of lambda expressions
public class Test {
    // operation is implemented using lambda expressions
    interface FuncInter1 {
        int operation(int a, int b);
    }

    // sayMessage() is implemented using lambda expressions
    // above
    interface FuncInter2 {
        void sayMessage(String message);
    }

    // Performs FuncInter1's operation on 'a' and 'b'
    private int operate(int a, int b, FuncInter1 fobj)
    {
        return fobj.operation(a, b);
    }

    public static void main(String args[])
    {
        // lambda expression for addition for two parameters
        // data type for x and y is optional.
        // This expression implements 'FuncInter1' interface
        FuncInter1 add = (int x, int y) -> x + y;

        // lambda expression multiplication for two
        // parameters This expression also implements
        // 'FuncInter1' interface
        FuncInter1 multiply = (int x, int y) -> x * y;

        // Creating an object of Test to call operate using
        // different implementations using lambda
        // Expressions
        Test tobj = new Test();

        // Add two numbers using lambda expression
        System.out.println("Addition is "
                           + tobj.operate(6, 3, add));

        // Multiply two numbers using lambda expression
        System.out.println("Multiplication is "
                           + tobj.operate(6, 3, multiply));

        // lambda expression for single parameter
        // This expression implements 'FuncInter2' interface
        FuncInter2 fobj = message
            -> System.out.println("Hello " + message);
        fobj.sayMessage("Geek");
    }
}
```