Dokumentacija za Izpit

Rok Kos

verzija: 26. januar 2017

Kazalo

1 Input/Output

1.1 Input

```java
import java.util.Scanner;
Scanner in = new Scanner(System.in);
byte b = in.nextByte();
int i = in.nextInt();
long l = in.nextLong();
double d = in.nextDouble();
String s = in.next();  // Vrne naslednji string
String line = in.nextLine();  // Prebere celotno vrstico in skoci v novo

// Branje do konca inputa
while(in.hasNextInt()){
        int a = in.nextInt();
}
// Namesto hasNextInt bi lahko bilo tudi:
// -hasNext()
// -hasNextDouble()
// -hasNextLong
// -hasNextLine
```

1.2 Output

```java
System.out.println(dnevi + ". dan: " + prej + " -> " + d + " (prehodil " + p + ")");
System.out.print("Ne gre v naslednjo vrsico" + 5);

System.out.printf("%d. dan: %d -> %d (prehodil %d)%n", dnevi, prej, d, p);
System.out.format("%1\$+020.10f", Math.PI);  // Enako kot print pri println in print(ne gre v novo vrstico)
// "1 dolar kateri argument"
// + pomeni predznacen, 0 pomeni da naj bodo spredaj vodilne 0
// 20.10 pomeni 20 mest spredaj in na 10 decimalk
// FORMATER
// %d - int, long, byte, %f - double, float, %s -string, %n - newline
// + -> predznak, - -> levo poravnan, 010.5 -> vodilne nicle 10 mest z 5 decimlkami
```

2 Delo s števili

2.1 Parsing

```java
String s = "123456789";
int a = Integer.parseInt(s);
long l = long.parseLong(s);
double d = Double.parseDouble(s);

String besedilo = "Neka dolga poved v kateri mores dobit vsako besedo";
String[] besede = besedilo.split(" ");  // Lahko tudi po , . !, ali celo po regex
for (int i = 0; i < besede.length; ++i) {
        System.out.println(besede[i]);
}
```

2.2 Math knjiznica

```java
double pi = Math.PI;  // Math.E
double a = abs(a);  // Tudi za int, long in float
double naj = Math.max(int, int)  // lahko tudi double, float in long
double naj = Math.min(int, int)  // lahko tudi double, float in long

// Trigonometricne funkcije
double kot = asin(val) / Math.PI * 180;  // acos, atan vrne vrednost v PI radianih
double val = sin(kot * Math.PI / 180);  // cos, tan
// Mozna pretvorba tudi z toDegrees ali toRadians

// Hiperbolicne funkcije
double h = sinh(val);  // cosh, tanh

// Zaokrozevanje
double navzgor = Math.ceil(decimalka);
double navzdol = Math.flor(decimalka);
long navzdol = Math.round(decimalka);  // vrne celo stevilo(lahko tudi int)

```

```java
// Korenjenje
double kvadratni = Math.sqrt(koren);
double kubicni = Math.cbrt(koren);

// Eksponetna funkcija
double potenca = Math.pow(osnova, eksponent);
double eNaEks = Math.exp(naDecimalko);
double obratno = Math.log(naravni);  // lahko tudi z desetisko osnovo (log10(a))

// Random
double r = Math.random()  // vrne od 0.0 do 1.0
```

2.3 Int/Long class

```java
int M = Integer.MAX_VALUE;  // MIN_VALUE
// Enako za byte, short, long, double

// Pretvorba velja tudi za long
String s = i.toString();
String b = i.toBinary();
String h = i.toHex();
String o = i.toOctal();
```

3 Delo z besedili

3.1 String class

```java
char a = besedilo.charAt(index);
int l = besedilo.length();
String s = str1.concat(str2);  // Doda str2 nakoncu str1

// Primerjanje
boolean enaka = string1.equals(string2);  // NUJNO UPORABLJAJ TO ZA PRIMERJANJE
int pred = string1.compareTo(string2);  // Vrne -1 ce je str1 pred str2 in 1 obratno, 0 ce sta enaka
// compareToIgnoreCase in equalsIgnoreCase je tudi na voljo

// Manipulacija stringov
String sub = str1.substring(zacetek, konec);
String[] s = str1.split(' ');  // Split po nekem znaku ali regex pravilu

String rep = str1.replace(kateriChar, zKaterimChar);  // zamenja vse pojavitve
String rep = str1.replaceAll(regex, sCim);  // zamnenja vse pojavitve, ki ustrezajo regex(lahko tudi normalen string
String rep = str1.replaceFirst(regex, sCim);  // zamnenja prvo pojavitev, ki ustreza regexu

String lower = str1.toLowerCase();
String upper = str1.toUpperCase();

boolean match = str1.matches(regex);

// Iskanje po stringu
int index = str1.indexOf(chr, fromIndex);  // Namesto chr lahko tudi String
int index = str1.lastIndexOf(chr, fromIndex);  // Namesto chr lahko tudi String

boolean seZacne = str1.startsWith(str2, fromIndex);
boolean seKonca = str1.startsWith(str2);
```

4 Razredi

4.1 Primer

```java
public class Primer {
        private int skrito;
        private static final int skritoSamoEnoKoncno = 1;
        protected int polaPola;  // Vidno razredom, ki dedujejo ta class, ostalim ne
        public int vsiVidijo;

        // Constructor
        public Primer () {
                this.skrito = 0;
                this.polaPola = 0;
        }
```

```
12          public Primer (int _skrito, int _polaPola) {
13                  this.skrito = _skrito;
14                  this.polaPola = _polaPola;
15          }
16
17          private int Metoda () {
18                  return 0;
19          }
20
21  }
22
23  public class PodPrimer extends Primer {  // Lahko bi exendali si en class takole : Primer, SeEnPrimer
24          private int samoOdTega;
25
26          public PodPrimer(int _skrito, int _polaPola, int _samoOdTega) {
27                  super(_skrito, _polaPola);  // Klic contruktorja od Primer
28                  this.samoOdTega = _samoOdTega;
29          }
30          @Override
31          public int Metoda() {
32                  super.Metoda();  // Klic metode Primer
33                  return 1;  // Mogoce je narobe
34          }
35
36  }
37
38  public abstract class AbstraktenPrimer {
39          // Enak kot primer samo da so v njem definirane metode in spremenljivke, ki jih kasneje
40          // drugi razredi podeduje, kot nek modelcek po katerem se dela ostale clase
41          // PAZI: ce podedujes tak class moras napisati definicije za vse njegove abstraktne metode
42
43          // Ce imamo abstrakno metodo hocemo, da imajo vsi, ki se dedujejo iz tega to metodo
44          // ampak jo vsak po svoje implementira
45          public abstract int Metoda();
46  }
47
48  public interface interfacePrimer {
49          // V interfacu samo specificiramo katere metode imamo(vse so abstrakne) in
50          // tudi class sam je abstrakten, ce ga podedujemo modramo definirati vse njegove
51          // metode
52
53          public void Metoda();
54  }
55
56  public class interfacePodPrimer implements interfacePrimer {
57          public void Metoda() {
58                  return 0;
59          }
60
61  }
62
63  public static void main(String[] args) {
64          Primer[] t = Primer[3];
65          t[0] = new Primer();
66          t[1] = new Primer(1,2);
67          t[2] = new PodPrimer(1,2,3);
68  }
69
70
71  // Sortiranje objektov
72  public class Primerjava implements Comparable<Primerjava> {
73          private int a;
74          @Override
75          public int compareTo (Primerjava other) {
76                  if (this.a < other.a) {
77                          return -1;
78                  } else if (this.a > other.a) {
79                          return 1;
80                  }
81                  return 0;
82          }
83  }
84
85  // Drug Primer
86  import java.util.Comparator;
87
88  private class Obj {
89          public int c;
90  }
91
92  private static class PrimerjajObj implements Comparator<Obj> {
```

```
93          @Override
94          public int compare (Obj a, Obj b) {
95                  if (a.c < b.c) {
96                          return -1;
97                  } else if (a.c > b.c) {
98                          return 1;
99                  }
100                 return 0;
101         }
102 }
```

5 TJ.exe

5.1 Uporaba

tj.exe <Program.java> <testi> <rezultati> -> normalno
tj.exe <razredi> <testi> <rezultati> -> razredi
tj.exe . . . -> slike
tj.exe -t 5s -> cas
tj.exe -p 5-10 -> primeri
Rocno:
javac program.java
java program < input.txt > output.txt
(java Program rezultat.png 700x500 za slike)
fc output.txt pravilno.txt (Linux/Mac diff)