

Vježbe 5 - kreiranje i korištenje biblioteka u *Javi*

Matej Mihelčić

Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu

matmih@math.hr

9. studenoga, 2022.



Biblioteka u programskom jeziku *Java* je **skup paketa i kompilacijskih jedinica** koje je **napisao neki programer** u programskom jeziku *Java* sa svrhom **ponovnog korištenja od strane drugih korisnika**.

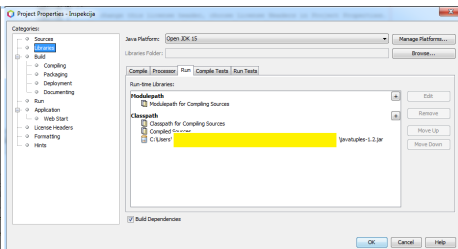
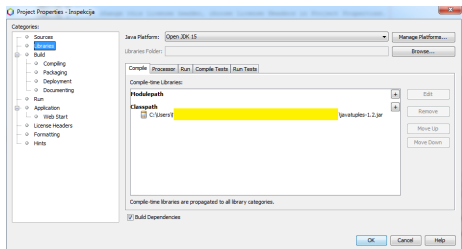
Biblioteka ne mora imati glavnu funkciju (*main*) već je cilj **pružati implementaciju neke funkcionalnosti preko niza klasa i metoda**. Biblioteka **može sadržavati i kod za testiranje** (tzv. *unit test*).

Kod korištenja biblioteke, **korisnik ne mora imati (i najčešće nema) izvorni kod biblioteke**, već dobije **izlazni .jar dokument** iz kojega se **učitaju odgovarajuće klase i sučelja**.

Zbog toga je bitno **generirati odgovarajuću dokumentaciju za biblioteke**. Dokumentaciju **generiramo koristeći alat javadoc** i njegove **razne opcije**.

Dodavanje biblioteke u *Apache NetBeans*-u

Za dodavanje biblioteke projektu u *Apache NetBeans*-u treba: desni klik na projekt → Properties → Libraries → dodati u compile classpath i run classpath.



Zadatak 1

Napišite biblioteku koja reprezentira funkcioniranje inspeksijske službe. Inspeksijska služba sadrži niz postaja, od kojih svaka sadrži niz inspektora. Inspeksijska služba, na prijavu korisnika (nekog službenog entiteta - npr. firme), stvara predmet i dodijeli ga inspektoru s najmanjim zaduženjem (ukupnim predviđenim brojem sati potrebnim za rješavanje predmeta). Inspektor rješava predmete redosljedom kojim ih prima. Inspektor može riješiti određeni predmet, međutim i korisnik može povući svoju prijavu. Napravite kod tako da je moguće koristiti i ispisati klase pozivom `System.out.println()` naredbe. Simulirajte upotrebu korištenja biblioteke od strane novog korisnika.

Za rješavanje ovog zadatka je korisno koristiti biblioteku `javatuples`¹.

¹<https://www.javatuples.org/>

Korištenje biblioteke JFreeChart

JFreeChart je Java biblioteka koja sadrži klase, sučelja i metode potrebne za spremanje i crtanje raznih grafova i dijagrama^{2,3}.

```
1 double[] vrijednosti = {}; //staviti neke vrijednosti
2 var podaci = new HistogramDataset(); //koristiti za
   spremanje podataka za histogram
3 podaci.addSeries("element", vrijednosti, 10); //dodamo
   vrijednosti, stvorimo 10 binova, element je vrijednost
   na legendi
4 JFreeChart histogram = ChartFactory.createHistogram("
   Normalna distribucija", "Vrijednosti", "Pojavljivanje",
   podaci); //stvorimo histogram, dodamo naslov i labele
   osima
5 ChartUtils.saveChartAsPNG(new File("histogram.png"),
   histogram, 450, 400); //spremimo graf u .png formatu
```

Crtanje histograma.

²<https://www.jfree.org/jfreechart/javadoc/index.html>

³<https://www.jfree.org/jfreechart/>

Korištenje biblioteke JFreeChart

```
1 var podaci1 = new XYSeriesCollection();//stvaramo kolekciju
   tocaka
2 var series = new XYSeries("Linijski plot"); //xy serija
3 int count=1;
4     for(double i:vrijednosti){
5         series.add(count++,i);//dodamo parove x,y
6     }
7     podaci1.addSeries(series);
8 JFreeChart chart = ChartFactory.createXYLineChart(
9     "Proba", "x", "y", podaci1, PlotOrientation.
   VERTICAL, true, true, false);
10 ChartUtils.saveChartAsPNG(new File("line.png"),chart, 450,
   400); //crtamo linijski graf i spremimo u .png dokument.
   Dimenzije su 450x400.
```

Crtanje linijskog grafa.

Korištenje biblioteke Apache Commons Math

Apache Commons Math⁴ sadrži niz klasa, sučelja i metoda koje pružaju dodatne funkcionalnosti u odnosu na standardni paket `java.lang.math`.

```
1 NormalDistribution distribucija = new NormalDistribution
    (0,0.2);
2 double vrijednosti[] = distribucija.sample(10000); //kreiram
    uzorak od 10000 tocaka iz normalne distribucije sa
    srednjom vrijednosti 0 i devijacijom 0.2 (uocimo
    prednost u odnosu na gornji pristup definiranja podataka
    ).
3 var podaci = new HistogramDataset();
4 podaci.addSeries("key", vrijednosti, 10);
5 JFreeChart histogram = ChartFactory.createHistogram("Prava
    normalna distribucija", "Vrijednosti", "Pojavljivanje",
    podaci);
6 ChartUtils.saveChartAsPNG(new File("histogram1.png"),
    histogram, 450, 400); //pogledajte razlike u dobivenim
    histogramima
```

Crtanje linijskog grafa.

⁴<https://commons.apache.org/proper/commons-math/>

U primjerima su korišteni sljedeći paketi:

```
1 import java.io.File;
2 import java.io.IOException;
3 import org.apache.commons.math3.distribution.
   NormalDistribution;
4 import org.jfree.chart.ChartFactory;
5 import org.jfree.chart.ChartUtils;
6 import org.jfree.chart.JFreeChart;
7 import org.jfree.chart.plot.PlotOrientation;
8 import org.jfree.data.statistics.HistogramDataset;
9 import org.jfree.data.xy.XYSeries;
10 import org.jfree.data.xy.XYSeriesCollection;
```

Korišteni paketi.

Stvaranje testova

Unit test ili test cjelina programa, stvaramo tako da: desni klik na kompilacijsku jedinicu → Tools → Create (Create/Update) Tests. Među testnim paketima nam se stvori ime paketa i testna klasa klase nad kojom smo izvršili akciju.

```
1 public class InspektorTest {
2     public InspektorTest() {}
3     @BeforeClass public static void setUpClass() {}
4     @AfterClass public static void tearDownClass() {}
5
6     @Test
7     public void testGetPredmeti() {
8         System.out.println("getPredmeti");
9         Inspektor instance = new Inspektor();
10        ArrayDeque<Predmet> expectedResult = null;
11        ArrayDeque<Predmet> result = instance.getPredmeti();
12        assertEquals(expectedResult, result); // ukoliko
13        getPredmeti vrati null, konstruktor ne radi dobro.
14    }
```

Definiranje testova cjelina klase.

Stvaranje testova

```
1  @Test
2  public void testGetIdentifikator() {
3      System.out.println("getIdentifikator");
4      Inspektor instance = new Inspektor();
5      int expResult = 0;
6      int result = instance.getIdentifikator();
7      assertEquals(expResult, result); //nesmije biti 0
8      Assert.assertTrue( result >= -1 ); //mora biti >= -1
9  }
10
11 @Test
12 public void testGetPostaja() {
13     System.out.println("getPostaja");
14     Inspektor instance = new Inspektor();
15     int result = instance.getPostaja();
16     Assert.assertTrue( result >= -1 ); //mora biti >= -1
17 }
```

Definiranje testova cjelina klase.

Stvaranje testova

```
1  @Test
2  public void testGetImeiPrezime() {
3      System.out.println("getImeiPrezime");
4      Inspektor instance = new Inspektor();
5      Pair<String, String> expectedResult = null;
6      Pair<String, String> result = instance.
getImeiPrezime();
7      assertEquals(expResult, result); //povratni par
nesmije biti null
8
9      String a = null;
10     assertEquals(a, result.getValue0());
11     assertEquals(a, result.getValue1()); //komponente
para nesmiju biti null
12 }
13
```

Definiranje testova cjelina klase.

Stvaranje testova

```
1  @Test
2  public void testRijesi() {
3      System.out.println("rijesi");
4      Inspekcija insp = new Inspekcija(10);
5      Inspektor instance = new Inspektor("Pero", "Peric");
6      insp.zaposliInspektora(instance);
7      Klient k = new Klient("Marko", "Markic", 12334);
8      String opis = "Neki opis";
9      Tuzeni t = new Tuzeni("Firma 1", 1);
10     k.podigniTuzbu(insp, t, opis);
11     int brEl = instance.predmeti.size();
12     assertEquals(brEl, 1); // nakon podizanja tuzbe, jedan
13     predmet mora biti dodijeljen inspektoru
14     instance.rijesi(insp);
15     brEl = instance.predmeti.size();
16     assertEquals(brEl, 0); // nakon rjesavanja, inspektor
17     ima 0 predmeta. Trebalo bi provjeriti i ostale klase...
18     } // Napravite provjeru ostalih klasa za DZ.
```

Definiranje testova cjelina klase. 

Stvaranje testova

```
1  @Test
2  public void testToString() {
3      System.out.println("toString");
4      Inspektor instance = new Inspektor();
5      String expectedResult = "";
6      String result = instance.toString();
7      assertEquals(expectedResult, result);
8      expectedResult = null;
9      assertEquals(expectedResult, result); //izlazni string
10     nesmije biti null
11 }
```

Definiranje testova cjelina klase.

Zadatak - DZ

Proširite biblioteku inspeksijske službe tako da uzima u obzir i dio grada. Svaka postaja treba imati zabilježen dio grada. Tužba se prosljeđuje postaji iz odgovarajućeg dijela grada koja trenutno ima najmanje zahtjeva.

Zadatak - DZ

Nacrtajte *box and whiskers* dijagram (podatke konstruirajte sami). Napravite usporedne *box and whiskers* dijagrame. Isprobajte i druge vrste dijagrama kao što su *pie* dijagram, *bubble* dijagram, *gant* dijagram.