



Jesenji semestar, 2021/22

***PREDMET: SE321 OBEZBEĐENJE KVALITETA, TESTIRANJE I
ODRŽAVANJE SOFTVERA***

Domaći zadatak: 4

Ime i prezime: Dušan Stanković

Broj indeksa: 3611

Datum izrade: 31.10.2021.

Tekst domaćeg zadatka:

Za proizvoljno odabranu aplikaciju koja je rađena na nekom od predmeta koji ste prethodno slušali i položili primeniti sledeće:

1. Definirati test slučajeve primenom testiranja na bazi specifikacije sistema i izvršiti ih.
2. Napraviti izveštaj o uočenim greškama ukoliko ih ima
3. Izvršiti JUnit testiranje dve odabrane klase aplikacije.

Rešenje zadatka:

Kao proizvoljno odabranu aplikaciju koristim aplikaciju koju sam radio u svrhu domaćeg zadatka koji sam koristio u par narednih vežbi. Radi se o sitnom programu koji “nagađa” koji je broj korisnik zamislio u datom rasponu brojeva. Kod i testove takođe prilažem u mail-u.

2. S obzirom na to da sam detaljno testirao program koji je takođe relativno jednostavan, pokrio sam sve moguće slučajeve testovima, i nikad nije dolazilo do greške, nevezano za raspon brojeva, algoritam uvek pogodi broj, i nema neočekivanih runtime grešaka.

3. Testirao sam četiri klase sa što više mogućnosti, i takođe sam komentarisao sve test metode kako bi kod bio pregledniji i čitljiviji.

```
//instanciranje guessera sa intervalom 0-10 zato što ćemo ga koristiti kroz celo testiranje
static NumGuess.Guesser g;
@BeforeClass
public static void classSetup() { g = new NumGuess.Guesser( l: 0, u: 10); }

@Test
public void test_instantiate() {
    assertEquals( expected: 0, g.lower);
    assertEquals( expected: 10, g.upper);
}
```

```
@Test()
public void test_output() {
    PrintStream sysOutBackup = System.out;
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    PrintStream out = new PrintStream(byteArrayOutputStream);
    System.setOut(out);
    // Dumpuje se ceo stream sa očekivanim odgovorima kako bi smo dobili rezultat za "zamišljeni" broj 5
    InputStream sysInBackup = System.in;
    ByteArrayInputStream in = new ByteArrayInputStream(new byte[]{'n', 'e', '\n', 'd', 'a', '\n', 'd', 'a', '\n'});
    System.setIn(in);

    g.quiz();

    // Upisujemo ceo string očekivanih povratnih poruka nakon dumpovanih bajtova u input stream kao očekivani rezultat
    System.setOut(sysOutBackup);
    System.setIn(sysInBackup);
    String output = byteArrayOutputStream.toString();
    String expected = "Zamislite broj između 0 i 10\n" +
        "da li je vaš broj veći od 5 ? \n" +
        "da li je vaš broj veći od 2 ? \n" +
        "da li je vaš broj veći od 4 ? \n" +
        "Vaš broj je: 5\n";
    assertEquals(expected, output);
}
```

```
// Test za exception koji vraća poruku ako se unese bilo koja druga vrednost osim "da" ili "ne"
@Test(expected = NumGuess.Guesser.InvalidInputException.class)
public void test_askIfGreaterThan_throwsOnInvalidInput() throws NumGuess.Guesser.InvalidInputException {
    InputStream sysInBackup = System.in;
    // Testiranje exceptiona za vrednost "dfa" posle koga se uspešno poziva exception da unesemo validne vrednosti
    ByteArrayInputStream in = new ByteArrayInputStream(new byte[]{'d', 'f', 'a', '\n'});
    System.setIn(in);
    g.askIfGreaterThan(10, new Scanner(System.in));
    System.setIn(sysInBackup);
}
```

```
// Test da li metoda lepo vraća veći broj ako se od njega to traži
@Test()
public void test_askIfGreaterThan() throws NumGuess.Guesser.InvalidInputException {
    InputStream sysInBackup = System.in;
    ByteArrayInputStream in = new ByteArrayInputStream(new byte[]{'d', 'a', '\n'});
    System.setIn(in);
    g.askIfGreaterThan(10, new Scanner(System.in));
    System.setIn(sysInBackup);
    assertTrue(condition: true);
}
```

Na kraju imamo uspešne rezultate svih testova:

✓	GuessTest	43 ms
✓	test_instantiate	2 ms
✓	test_askIfGreaterThan_throwsOnInv	29 ms
✓	test_output	11 ms
✓	test_askIfGreaterThan	1 ms