

Domaći zadatak 05 – Branislav Manojlović kp11 KI301-DZ05

Tekst zadatka

Odabrati aplikaciju u Java programskom jeziku i izvršiti sledeće:

1. Primentiti JUnit testiranje u razvojnom okruženju i izvršiti jedinično testiranje,
2. Prikazati dobijene rezultate JUnit testiranja i opisati tako dobijene rezultate.

Rešenje:

Kao primer JUnit testiranje ćemo izvršiti nad konzolnom aplikacijom Kalkulator. Glavna klasa ove aplikacije je **Kalkulator.java** i u njoj se nalaze kreirane sve metode koje će biti testirane.

Izvorni kod klase Kalkulator.java aplikacije koju testiram

```
package com.manojlovic.Funkcija;

import java.util.Scanner;

/**
 *
 * @author Bane
 */
public class Kalkulator {

    static Scanner sc = new Scanner(System.in);

    //metoda za sabiranje dva broja
    public static double sabiranje(int a, int b){
        int rez=0;
        rez = a + b;
        return rez;
    }

    //metoda za oduzimanje dva broja
    public static double oduzimanje(int a, int b){
        int rez=0;
        rez = a - b;
        return rez;
    }

    //metoda za mnozenje dva broja
    public static double mnozenje(int a, int b){
        int rez=0;
        rez = a * b;
        return rez;
    }
}
```

```

//metoda za deljenje dva broja
public static double deljenje(int a, int b){
    int rez=0;
    if(a==0 || b==0){
        throw new NullPointerException("Deljenje sa nulom nije moguće!!!");
    }else{
        rez = a / b;
    }
    return rez;
}

//metoda za pokretanje i ispis menija
public static void meni(){
    int a, b, opcija;
    System.out.println("Unesite prvi broj: ");
    a = sc.nextInt();
    System.out.println("Unesite drugi broj: ");
    b = sc.nextInt();
    System.out.println("-----MENI-----");
    System.out.println("1. Sabiranje");
    System.out.println("2. Oduzimanje");
    System.out.println("3. Mnozenje");
    System.out.println("4. Deljenje");
    System.out.println("Izaberite zeljenu operaciju: ");
    opcija = sc.nextInt();
    switch(opcija){
        case 1:
            System.out.println("Zbir iznosi: " + sabiranje(a, b));
            break;
        case 2:
            System.out.println("Razlika iznosi: " + oduzimanje(a, b));
            break;
        case 3:
            System.out.println("Proizvod iznosi: " + mnozenje(a, b));
            break;
        case 4:
            System.out.println("Kolicnik iznosi: " + deljenje(a, b));
            break;
        default:
            System.out.println("Izabrali ste napostojecu opciju");
    }
}

//main metoda za pokretanje programa
public static void main(String[] args) {
    meni();
}
}

```

Primenom NetBeans radnog okruženja u mogućnosti sam da automatski generišem testne metode pomoću kojih ću testirati svaku metodu iz klase **Kalkulator.java**. Osnovni koraci za generisanje ovih metoda su:

Desni klik na klasu Kalkulator.java → Tools → Create/Update Tests → Ok

Nakon ovih koraka NetBeans mi sam automatski generiše Testnu java klasu koja se zove **KalkulatorTest.java** i u njoj se nalaze izgenerisane i potrebne testne metode za testiranje svake od metoda koje postoje klasi **Kalkulator.java**.

U novoj testnoj klasi vršim modifikacije pojedinih testnih metoda tako da u njima definišem ulazne vrednosti i očekivane rezultate koje treba da dobijem, odnosno koje očekujem da dobijem izvršavanjem metoda pri testiranju. Takođe, mogu da dodam i određene delove koda koji će omogućiti ispis rezultata testiranja na konzoli NetBeans radnog okruženja.

Izvorni kod testne klase KalkulatorTest.java

```
package com.manojlovic.Funkcija;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 *
 * @author Bane
 */
public class KalkulatorTest {

    public KalkulatorTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }
}
```

```

}

/**
 * Test of sabiranje method, of class Kalkulator. Proverava da li metoda
 * pravilno sabira dva broja
 */
@Test
public void testSabiranje() {
    System.out.println("Test 1: sabiranje - ");
    int a = 7;
    int b = 9;
    double expectedResult = 16.0;
    double result = Kalkulator.sabiranje(a, b);
    assertEquals(expectedResult, result, 0.0);
    if (expectedResult == result) {
        System.out.println(" Passed!");
    } else {
        System.out.println(" Failed!");
    }
}

/**
 * Test of oduzimanje method, of class Kalkulator. Proverava da li metoda
 * pravilno oduzima dva broja
 */
@Test
public void testOduzimanje() {
    System.out.println("Test 2: oduzimanje - ");
    int a = 8;
    int b = 6;
    double expectedResult = 2.0;
    double result = Kalkulator.oduzimanje(a, b);
    assertEquals(expectedResult, result, 0.0);
    if (expectedResult == result) {
        System.out.println(" Passed!");
    } else {
        System.out.println(" Failed!");
    }
}

/**
 * Test of mnozenje method, of class Kalkulator. Proverava da li metoda
 * pravilno mnozi dva broja
 */
@Test
public void testMnozenje() {
    System.out.println("Test 3: mnozenje - ");
    int a = 8;
    int b = 5;
    double expectedResult = 40.0;

```

```

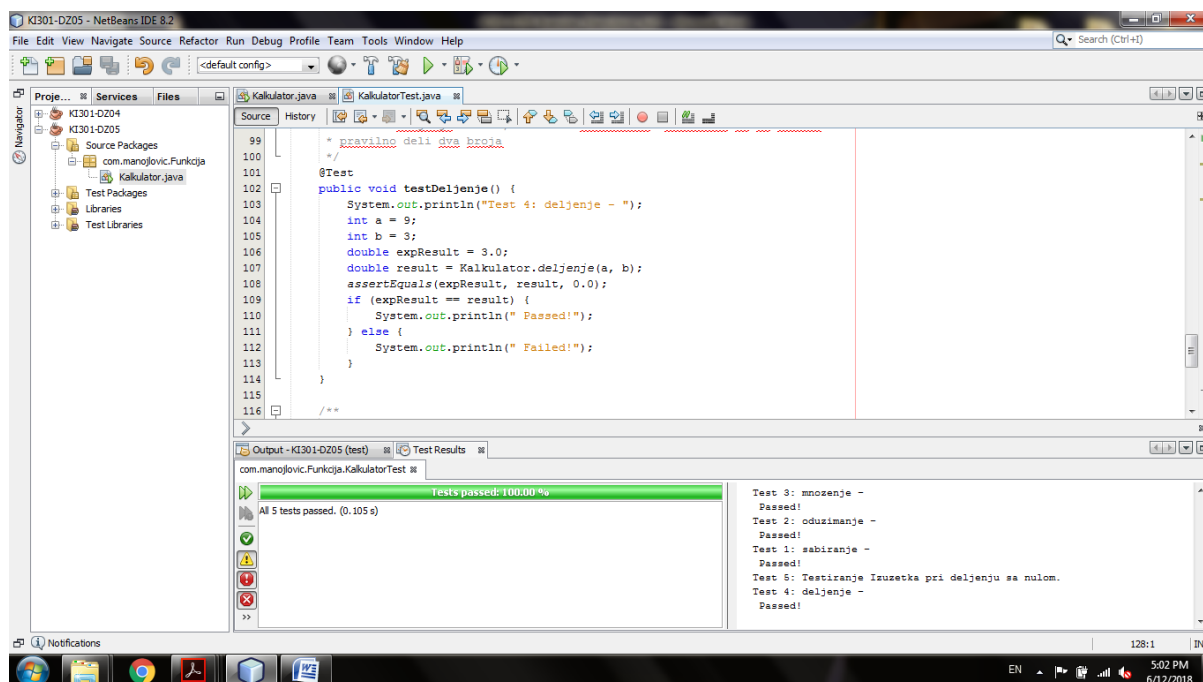
        double result = Kalkulator.mnozenje(a, b);
        assertEquals(expResult, result, 0.0);
        if (expResult == result) {
            System.out.println(" Passed!");
        } else {
            System.out.println(" Failed!");
        }
    }
}

/**
 * Test of deljenje method, of class Kalkulator. Proverava da li metoda
 * pravilno deli dva broja
 */
@Test
public void testDeljenje() {
    System.out.println("Test 4: deljenje - ");
    int a = 9;
    int b = 3;
    double expResult = 3.0;
    double result = Kalkulator.deljenje(a, b);
    assertEquals(expResult, result, 0.0);
    if (expResult == result) {
        System.out.println(" Passed!");
    } else {
        System.out.println(" Failed!");
    }
}

/**
 * Test of meni method, of class Kalkulator. Proverava da li postoji
 * NullPointerException
 */
@Test(expected = NullPointerException.class)
public void testDeljenje1() {
    System.out.println("Test 5: Testiranje Izuzetka pri deljenju sa nulom.");
    int a = 9;
    int b = 0;
    double result = Kalkulator.deljenje(a, b);
}
}

```

Ukoliko svi testovi uspešno prođu na konzoli NetBeans radnog okruženja se nakon izvršavanja testne klase kao izlaz dobije da su svi testovi **passed 100%**. A na desnoj strain se vide ispisi za svaku izvršenu testnu metodu. Ovaj ispis sam setovao kakko bih lakše pratio izvršavanje testova. Primer ispisa rezultata testova je dat na sledećoj slici.



Slika 1. Ispis rezultata izvršenih testova

Na slici 1 vidimo da su svi testovi izvršeni uspešno, što znači da sve metode u klasi Kalkulator.java rade ispravno, tj. daju željene rezultate koji su predefinisani u testnim metodama.

Kao dodatak domaćem zadatku 05 šaljem i fajl domaćeg zadatka kreiranog u NetBeans-u. To je fajl **KI301-DZ05** i u njemu se nalaze klasa koja je testirana (**Kalkulator.java**) i testna klasa (**KalkulatorTest.java**).

MSc Branislav Manojlović
Novi Sad 12.06.2018.