Inżynieria oprogramowania - sprawozdanie

Testy jednostkowe z użyciem narzędzi JUnit oraz JMockit

Termin zajęć:

21.12.2017

Autorzy:

- Paweł Biel, 225949
- Bartosz Rodziewicz, 226105

Prowadzący:

dr inż. Paweł Głuchowski

Dane testowe do testów jednostkowych

```
package MainApp;

public final class TestData {
    private TestData() {}

    static String studentsData[][] = new String[][] {
            {"226105","xxx","Bartosz","Rodziewicz","baato@chan.com"}, {"123456","test","Adam","K
            {"234567","zzz","Damian","Nowak","damiannowak@wp.pl"}, {"111","password","Maciek","B
            {"123","haslo1","Adam","Nowak","adam@nowak.pl"}
    };

    static Student students[] = {
            new Student("226105","xxx","Bartosz","Rodziewicz","baato@chan.com"),
            new Student("123456","test","Adam","Kowalski","adam.kowalski@gmail.com"),
            new Student("234567","zzz","Damian","Nowak","damiannowak@wp.pl")
    };

    static String adminsData[][] = new String[][] {
            {"barto","zzz","Bartosz","Rodziewicz"}, {"root","toor","ROOT",""}, {"adam123","haslo
    };

    static Admin admins[] = {
            new Admin("barto","zzz","Bartosz","Rodziewicz"),
            new Admin("root","toor","ROOT","")
    };

    static String coursesData[][] = new String[][] {
            {"Test Tset","A","1"}, {"Inzynieria Oprogramowania","C","111"}, {"XXX","B","2"}
    };

    static Course courses[] = {
            new Course("Test Tset","A",1),
            new Course("Inzynieria Oprogramowania","C",111)
    };

    static String coursesPrintOutput[] = {
            "A         |Test Tset                                              |1   |0",
            "C         |Inzynieria Oprogramowania                              |111 |0"
    };

    static String groupsData[][] = new String[][] {
            {"test 123","AA","marcin jakistam","c13 1.12", "10"}, {"Inzynieria Oprogramowania L"
            {"Inzynieria Oprogramowania L","CB","Marek","C2 24","5"}, {"test 321","AB","jerzy pr
    };

    static Group groups[] = {
            new Group("test 123","AA","marcin jakistam","c13 1.12", 10, courses[0]),
            new Group("Inzynieria Oprogramowania L","CA","Jedrzej","C 244",1, courses[1]),
            new Group("Inzynieria Oprogramowania L","CB","Marek","C2 24",5, courses[1])
    };
}
```

Należy wykonać test jednostkowy metod klasy, która stanowi klasę końcową w łańcuchu powiązań na diagramie klas

```java
package MainApp;

import org.junit.Assert;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import java.util.ArrayList;
import java.util.Arrays;

import static org.junit.Assert.*;

@Category(TestCategory2.class)
public class StudentTest {
    private static Student s1 = TestData.students[0];
    private static Student s2 = TestData.students[1];
    private static Group g1 = TestData.groups[0];
    private static Group g2 = TestData.groups[1];
    private static Group g3 = TestData.groups[2];

    @BeforeClass
    public static void setUp() throws Exception {
        Main.students.addAll(Arrays.asList(TestData.students));
        Main.admins.addAll(Arrays.asList(TestData.admins));
        Main.courses.addAll(Arrays.asList(TestData.courses));
        Main.groups.addAll(Arrays.asList(TestData.groups));
        s1.addGroup(g1);
        g1.addStudent(s1);
        s1.addGroup(g2);
        g2.addStudent(s1);
        s2.addGroup(g3);
        g3.addStudent(s2);

    }

    @Test
    public void checkIfAddedToAnotherGroupOfThisCourse() {
        assertFalse(s2.checkIfAddedToAnotherGroupOfThisCourse(g1.getCourse()));
        assertTrue(s1.checkIfAddedToAnotherGroupOfThisCourse(g3.getCourse()));
    }

    @Test
    public void getGroups() {
        ArrayList<Group> g = new ArrayList<>();
        g.add(g1);
        g.add(g2);

        assertEquals(g, s1.getGroups());
    }
}
```

Należy wykonać test jednostkowy metod klasy, która stanowi klasę w łańcuchu powiązań na diagramie klas

```java
package MainApp;

import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import java.util.Arrays;

import static org.junit.Assert.*;

@Category({TestCategory1.class, TestCategory2.class})
public class CourseTest {

    @BeforeClass
    public static void setUp() throws Exception {
        Main.students.addAll(Arrays.asList(TestData.students));
        Main.admins.addAll(Arrays.asList(TestData.admins));
        Main.courses.addAll(Arrays.asList(TestData.courses));
        Main.groups.addAll(Arrays.asList(TestData.groups));
    }

    @Test
    public void exists() {
        for (int i = 0; i < 2; i++) {
            assertTrue(Course.exists(TestData.coursesData[i][1]));
        }
        assertFalse(Course.exists(TestData.coursesData[2][1]));
    }

    @Test
    public void print() {
        int i = 0;
        for (Course c: Main.courses) {
            assertEquals(TestData.coursesPrintOutput[i], c.print());
            i++;
        }
    }
}
```

Należy wykonać testy jednostkowe wybranych metod klasy opartej na wzorcu Fasada

```java
package MainApp;

import org.junit.Assert;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import org.junit.experimental.categories.Category;

import java.util.Arrays;

import static org.junit.Assert.*;

@Category(TestCategory1.class)
public class MainTest {

    @BeforeClass
    public static void setUp() {
        Main.students.addAll(Arrays.asList(TestData.students));
        Main.admins.addAll(Arrays.asList(TestData.admins));
        Main.courses.addAll(Arrays.asList(TestData.courses));
        Main.groups.addAll(Arrays.asList(TestData.groups));
    }

    @Test
    public void findStudent() {
        for (int i = 0; i < 3; i++) {
            assertEquals(TestData.students[i], Main.findStudent(TestData.studentsData[i][0]));
        }
        assertNull(Main.findStudent(TestData.studentsData[3][0]));
        assertNull(Main.findStudent(TestData.studentsData[4][0]));
    }

    @Test
    public void findCourse() {
        for (int i = 0; i < 2; i++) {
            assertEquals(TestData.courses[i], Main.findCourse(TestData.coursesData[i][1]));
        }
        assertNull(Main.findCourse(TestData.coursesData[2][1]));
    }

    @Test
    @Category(TestCategory3.class)
    public void findGroup() {
        for (int i = 0; i < 3; i++) {
            assertEquals(TestData.groups[i], Main.findGroup(TestData.groupsData[i][1]));
        }
        assertNull(Main.findGroup(TestData.groupsData[3][1]));
    }

    @Test
    public void findAdmin() {
        for (int i = 0; i < 2; i++) {
            assertEquals(TestData.admins[i], Main.findAdmin(TestData.adminsData[i][0]));
        }
        assertNull(Main.findAdmin(TestData.adminsData[2][0]));
    }
}
```

Należy wykonać zestawy testów

Interfejsy do kategorii testów

```
package MainApp;

public interface TestCategory1 {}
```

```
package MainApp;

public interface TestCategory2 {}
```

```
package MainApp;

public interface TestCategory3 {}
```

Grupy testów

```
package MainApp;

import org.junit.experimental.categories.Categories;
import org.junit.runner.RunWith;


@Categories.SuiteClasses({CourseTest.class,MainTest.class,StudentTest.class})
@RunWith(Categories.class)
@Categories.IncludeCategory(TestCategory1.class)
public class TestGroup1 {}
```

```
package MainApp;

import org.junit.experimental.categories.Categories;
import org.junit.runner.RunWith;


@Categories.SuiteClasses({CourseTest.class,MainTest.class,StudentTest.class})
```

```
@RunWith(Categories.class)
@Categories.IncludeCategory(TestCategory2.class)
public class TestGroup2 {}
```

```
package MainApp;

import org.junit.experimental.categories.Categories;
import org.junit.runner.RunWith;


@Categories.SuiteClasses({CourseTest.class,MainTest.class,StudentTest.class})
@RunWith(Categories.class)
@Categories.IncludeCategory(TestCategory1.class)
@Categories.ExcludeCategory(TestCategory3.class)
public class TestGroup3 {}
```

```
package MainApp;

import org.junit.experimental.categories.Categories;
import org.junit.runner.RunWith;


@Categories.SuiteClasses({CourseTest.class,MainTest.class,StudentTest.class})
@RunWith(Categories.class)
public class TestGroupAll {}
```