```
package OptDist;

/**
 *
 * @author Madalina
 */


public class OptionalTest {

   public void testGetCapacity()
   {
      Optional optional = new Optional("Matlab",2,2,100,15);
      assertTrue(optional.GetCapacity()==100);

   }
    public void testGetYear()
   {
      Optional optional;
      optional = new Optional("Matlab",2,2,100,15);
      assertTrue(optional.GetYear()==2);

   }

   public void testGetSemester()
   {
      Optional optional = new Optional("Matlab",2,2,100,15);
      assertTrue(optional.GetSemester()==2);

   }
```

```
public void testGetName()
{
    Optional optional = new Optional("Matlab",2,2,100,15);
    assertTrue("Matlab".equals(optional.GetName()));

}

   public void testgetID()
{
    Optional optional = new Optional("Matlab",2,2,100,15);
    assertTrue(optional.getID()== 15);

}

public void invalidSemesterTest(){ //semestru>2
   System.out.println("getSemester");
   int semester=3;
   Optional instance = new Optional ("IC",2,3,100,15);
   String expResult = null;
   float result = instance.GetSemester();
   assertEquals(expResult,result);
}

  public void invalidSemesterTest2(){ //semestru<-1
   System.out.println("getSemester");
   int semester=-1;
   Optional instance = new Optional ("IC",2,-1,100,15);
   String expResult = null;
   float result = instance.GetSemester();
   assertEquals(expResult,result);
}
```

```
  public void invalidYearTest(){ //an>3
    System.out.println("getYear");
    int year=4;
    Optional instance = new Optional ("IC",4,2,100,15);
    String expResult = null;
    float result = instance.GetYear();
    assertEquals(expResult,result);
  }

  public void invalidYearTest2(){ //an<2
    System.out.println("getYear");
    int year=1;
    Optional instance = new Optional ("IC",1,2,100,15);
    String expResult = null;
    float result = instance.GetYear();
    assertEquals(expResult,result);
  }

  public void invalidCapacityTest2(){ //capacitate <1
    System.out.println("getCapicity");
    int capacity=0;
    Optional instance = new Optional ("IC",1,2,0,15);
    String expResult = null;
    float result = instance.GetCapacity();
    assertEquals(expResult,result);
  }
   public void testInvalidName() { //numele este null
     System.out.println("getName");
     Optional instance = new Optional(" ",2,2,100,13);
     String expResult = null;
```

```
        String result = instance.getName();
        assertEquals(expResult, result);

    }

        public void testNullID() { //ID este null
        System.out.println("getID");
        Optional instance = new Optional("AG",2,2,100,0);
        String expResult = null;
        int result = instance.getID();
        assertEquals(expResult, result);

    }

    private void assertEquals(String expResult, float result) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    }

       private void assertTrue(boolean b) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    }

    private void assertEquals(String expResult, String result) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    }

}
```

```
package OptDist;

import java.util.*;
import OptDist.Student;
import OptDist.StudentAdministration;
import java.util.ArrayList;
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.*;

public class PackageTest {

    private List<Optional> optionals;

    public void testSetOptionals(){
        List<Optional> optionals1;
        Package packagesest = new Package(optionals,1,2,"pachetul1");
        packagesest.setOptionals(optionals1);

    }

    @Test
    public void GetName(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        assertTrue(packagetest.getName() == "pachetul1");
    }

    @Test
    public void GetSemester(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        assertTrue(packagetest.getSemester() == 1);
    }

    @Test
    public void GetID(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        assertTrue(packagetest.ID == packagetest.getID());
    }

    @Test
    public void GetYear(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        assertTrue(packagetest.getYear() == 1);
    }

    @Test
    public void SetYear(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        packagetest.setYear(2);
```

```
        assertTrue(packagetest.getYear() == 2);
    }

    @Test
    public void SetSemester(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        packagetest.setYear(3);
        assertTrue(packagetest.getYear() == 3);
    }

    @Test
    public void SetName(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        packagetest.setName("paachetul2");
        assertTrue(packagetest.getName() == "pachetul2");
    }

    @Test
    public void SetID(){
        Package packagetest = new Package(optionals,1,2,"pachetul1");
        packagetest.setID(123);
        assertTrue(packagetest.getID()==123);
    }


}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package test;

import OptDist.Student;
import OptDist.StudentAdministration;
import java.util.ArrayList;
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.*;

public class StudentAdministrationTest {

    public StudentAdministrationTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    //daca se introduce in lista de studenti un student fara nume, prenume etc
    @Test
    public void addStudentTest1() {
        StudentAdministration instance = new StudentAdministration();
        Student s = new Student(null,null,null,null,0);
        instance.addStudent(s);

    }

     //daca se introduce in lista de studenti un student cu nume, prenume etc care nu respecta formatul
    @Test
    public void addStudentTest2() {
        StudentAdministration instance = new StudentAdministration();
        Student s = new Student("  ","9835","a6fb",null,15);  //spatii in loc de nrmatr, cifre la nume, nota>10
```

```
        instance.addStudent(s);

    }


    //daca se introduce in lista de studenti un student fara nume, prenume etc
    @Test
    public void compareTest1() {
        StudentAdministration instance = new StudentAdministration();
        Student s1 = new Student("123aa","ana","popescu","A3",7);
        Student s2 = new Student("123bb","alex","ionescu","B3",8);
        float expectedResult = 8; //ma astept ca dupa comparare functia sa imi intoarca nota mai mare
        float actualResult = instance.orderStudents().compare(s1, s2);  //!!! au declarat grade ca fiind float, dar
functia lor returneaza int ->alt scenariu de test
                                                    // daca introduci o variabila de alt tip?
        assertEquals(expectedResult, actualResult);     //compar rezultatele sa vad daca sunt egale, daca nu, pica
testul

    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author Andreea
 */
package optDist;

import OptDist.Student;
import OptDist.StudentAdministration;
import java.util.ArrayList;
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.*;

public class StudentAdministrationTest {


   //daca se introduce in lista de studenti un student fara nume, prenume etc
   @Test
   public void addStudentTestStudentNull() {
      StudentAdministration instance = new StudentAdministration();
      StudentAdministration instance2 = new StudentAdministration();
      Student s = new Student(null, null, null, null, 0);
      instance.addStudent(s);
```

```
    if (s == null)
    {
        fail("The Student should not be null");
    }
}

//testul reuseste daca studentul null nu a fost adaugat
@Test
public void addStudentTestStudentNullAdaugat() {
    StudentAdministration instance = new StudentAdministration();
    StudentAdministration instance2 = new StudentAdministration();
    Student s = new Student(null, null, null, null, 0);
    instance.addStudent(s);
    assertEquals(instance, instance2);
}

 //daca se introduce in lista de studenti un student cu nrmatricol care nu este string
@Test
public void addStudentTestNrMatricolNotString() {
    StudentAdministration instance = new StudentAdministration();
    Student s = new Student(2,"Radu","Andrei","A3",10);
    if (!(s.getNrMatricol() instanceof String))
    {
        fail("NrMatricol should be a string");
    }
    instance.addStudent(s);

}

//testul reuseste daca studentul cu nrmatricol!=string nu a fost adaugat
@Test
```

```
public void addStudentTestNrMatricolNotStringAdaugat() {
   StudentAdministration instance = new StudentAdministration();
   StudentAdministration instance2 = new StudentAdministration();
   Student s = new Student(null, null, null, null, 0);
   instance.addStudent(s);
   assertEquals(instance, instance2);
}

 //daca se introduce in lista de studenti un student cu nume care nu este string
@Test
public void addStudentTestNumeNotString() {
   StudentAdministration instance = new StudentAdministration();
   Student s = new Student("123",12,"Andrei","A3",10);
   if (!(s.getName() instanceof String))
   {
      fail("Name should be a string");
   }
   instance.addStudent(s);

}

//testul reuseste daca studentul cu nume!=string nu a fost adaugat
@Test
public void addStudentTestNumeNotStringAdaugat() {
   StudentAdministration instance = new StudentAdministration();
   StudentAdministration instance2 = new StudentAdministration();
   Student s = new Student(null, null, null, null, 0);
   instance.addStudent(s);
   assertEquals(instance, instance2);
}
```

```
 //daca se introduce in lista de studenti un student cu prenume care nu este string
@Test
public void addStudentTestPrenumeNotString() {
   StudentAdministration instance = new StudentAdministration();
   Student s = new Student("123","Radu",123,"A3",10);
   if (!(s.getSurname() instanceof String))
   {
      fail("Surname should be a string");
   }
   instance.addStudent(s);

}

//testul reuseste daca studentul cu prenume!=string nu a fost adaugat
@Test
public void addStudentTestPrenumeNotStringAdaugat() {
   StudentAdministration instance = new StudentAdministration();
   StudentAdministration instance2 = new StudentAdministration();
   Student s = new Student(null, null, null, null, 0);
   instance.addStudent(s);
   assertEquals(instance, instance2);
}

 //daca se introduce in lista de studenti un student cu grupa care nu este string
@Test
public void addStudentTestGroupNotString() {
   StudentAdministration instance = new StudentAdministration();
   Student s = new Student("123","Radu","Andrei",3,10);
   if (!(s.getGroup() instanceof String))
   {
      fail("Group should be a string");
```

```
    }
    instance.addStudent(s);

}

//testul reuseste daca studentul cu grupa!=string nu a fost adaugat
@Test
public void addStudentTestGroupNotStringAdaugat() {
    StudentAdministration instance = new StudentAdministration();
    StudentAdministration instance2 = new StudentAdministration();
    Student s = new Student(null, null, null, null, 0);
    instance.addStudent(s);
    assertEquals(instance, instance2);
}

  //daca se introduce in lista de studenti un student cu grade care nu este float
@Test
public void addStudentTestGradeNotFloat() {
    StudentAdministration instance = new StudentAdministration();
    Student s = new Student("123","Radu","Andrei","A3","8");
    if (!(s.getGrade() instanceof Float))
    {
        fail("Grade should be a string");
    }
    instance.addStudent(s);

}

//testul reuseste daca studentul cu grade!=float nu a fost adaugat
@Test
public void addStudentTestGradeNotFloatAdaugat() {
```

```
        StudentAdministration instance = new StudentAdministration();
        StudentAdministration instance2 = new StudentAdministration();
        Student s = new Student(null, null, null, null, 0);
        instance.addStudent(s);
        assertEquals(instance, instance2);
    }

    //verifica daca compare returneaza tot float
    @Test
    public void compareTestCompareReturnFloat() {

        Student s1 = new Student("123aa","ana","popescu","A3",7);
        Student s2 = new Student("123bb","alex","ionescu","B3",8);
        if (!(compare(s1, s2)) instanceof Float)
            fail("Compare does not return a float value");

    }
    //verifica daca compare returneaza nota mai mare
    @Test
    public void compareTestBiggestGrade() {

        Student s1 = new Student("123aa","ana","popescu","A3",7);
        Student s2 = new Student("123bb","alex","ionescu","B3",8);
        float expectedResult = 8;
        float actualResult = compare(s1, s2);
        if (expectedResult != actualResult)
            fail("Compare does not return the biggest value");
    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package test;
import OptDist.Student;
import OptDist.StudentAdministration;
import java.util.ArrayList;
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.*;


public class StudentTest {
    public StudentTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    //daca numele, prenumele, nrMatricol, grupa sunt nule, iar nota 0
    public void testGetNrMatricol() {
        System.out.println("getNrMatricol");
        Student instance = new Student(null,null,null,null,0);
        String expResult = null;
        String result = instance.getNrMatricol();
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

    //daca in loc de nume se introduc cifre
    @Test
    public void testGetName() {
```

```
    System.out.println("getName");
    Student instance = new Student("asaf","1234","df","fdv",10);
    String expResult = null;
    String result = instance.getName();
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}

//daca in loc de prenume se introduc spatii
@Test
public void testGetSurname() {
    System.out.println("getSurname");
    Student instance = new Student("asaf","adca","  ","fdv",9);
    String expResult = null;
    String result = instance.getSurname();
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}

//daca in loc de un format de genul "A3" se introduce altceva
@Test
public void testGetGroup() {
    System.out.println("getGroup");
    Student instance = new Student("aaa","bfd","sdsd","AAAAAAAAAAAA",8);
    String expResult = null;
    String result = instance.getGroup();
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}

//daca nota este mai mare decat 10
@Test
public void testGetGrade() {
    System.out.println("getGrade");
    Student instance = new Student("aaa","bfd","sdsd","asd",15);
    String expResult = null;
    float result = instance.getGrade();
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}

//daca anul>3
@Test
public void testGetYear() {
    System.out.println("getGrade");
    int year = 5;
    Student instance = new Student("aaa","bfd","sdsd","asd",8);
    String expResult = null;
```

```
        float result = instance.getYear();
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }


    //!!!! In clasa Student, year nu face parte din constructor
}
```

```
public void testTTL()
{
   System.out.println("setTTL"); // vom verifica daca TTL > data_curenta
   Calendar newTTL = GregorianCalendar.getInstance();
   newTTL.set(1997,6,20); // 20-mai-1997 (numerotarea lunilor se face de la 0
   FormInfo instance = new FormInfo(newTTL,null);
   fail("Invalid TTL");
}
```