

СКИТ – Лабораториска вежба 4

Со користење на pitclipse (pitest) и junit да се направат тестови за да се постигне 100% Mutation Coverage на мутантите генерирани од pitclipse (pitest) за дадениот изворен код.

Дадени се следниве класи:

Student.java

```
public class Student {  
  
    private String name;  
    private String surname;  
    private Course course;  
    private int age;  
  
    public Student(String name, String surname, Course course, int age) {  
        this.name = name;  
        this.surname = surname;  
        this.course = course;  
        this.age = age;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getSurname() {  
        return surname;  
    }  
}
```

```

    public void setSurname(String surname) {
        this.surname = surname;
    }

    public Course getCourse() {
        return course;
    }

    public void setCourse(Course course) {
        this.course = course;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public String toString() {
        return "Student{" +
            "name='" + name + '\'' +
            ", surname='" + surname + '\'' +
            ", course=" + course +
            ", age=" + age +
            '}';
    }
}

```

Conference.java

```

import java.util.ArrayList;
import java.util.List;

public class Conference {

    public static double TICKET_PRICE = 10.90;
    public static double EMT_DISCOUNT = 0.5;
    public static double WEB_DISCOUNT = 0.3;

    private List<Student> attendees;
    private int capacity;
}

```

```

public Conference(int capacity) {
    this.attendees = new ArrayList<>(capacity);
    this.capacity = capacity;
}

public double calculateTotalPricePaid() {
    double price = 0;

    for (Student p : attendees) {
        if (p.getCourse().name().equals("EMT"))
            price += (1 - EMT_DISCOUNT) * TICKET_PRICE;
        else if (p.getCourse().name().equals("WEB"))
            price += (1 - WEB_DISCOUNT) * TICKET_PRICE;
        else
            price += TICKET_PRICE;
    }
    return price;
}

public boolean addAttendeeToConference(Student student) {
    if ((this.attendees.size() < capacity) || tripleCapacity()) {
        this.attendees.add(student);
        return true;
    } else return false;
}

public boolean tripleCapacity() {
    if (this.capacity * 3 > 10000)
        return false;
    this.capacity *= 3;
    return true;
}

public List<Student> getAttendees() {
    return attendees;
}

public int getCapacity() {
    return capacity;
}
}

```

Во класата `ConferenceTest.java` дадена е имплементацијата т.е. тестовите со кои е постигната 100% Mutation Coverage на мутантите генерирани за дадениот изворен код.

ConferenceTest.java

```
import org.junit.Before;
import org.junit.Test;

import java.util.List;

import static org.junit.Assert.*;

public class ConferenceTest {

    private static final double TICKET_PRICE = 10.90;
    private static final double EMT_DISCOUNT = 0.5;
    private static final double WEB_DISCOUNT = 0.3;

    private Conference conference;
    private Student student1;
    private Student student2;
    private Student student3;

    @Before
    public void setUp() {
        conference = new Conference(2);
        student1 = new Student("John", "Doe", Course.EMT, 20);
        student2 = new Student("Jane", "Doe", Course.WEB, 21);
        student3 = new Student("Jim", "Beam", Course.OS, 22);
    }

    @Test
    public void testCalculateTotalPricePaidWithWebCourse() {
        Conference conference = new Conference(100);
        conference.addAttendeeToConference(new Student("WebStudent",
"LastName", Course.WEB, 20));
        double totalPrice = conference.calculateTotalPricePaid();
        assertEquals(10.90 * 0.7, totalPrice, 0.01); // WEB_DISCOUNT = 0.3
    }
}
```

```

@Test
public void testAddAttendeeToConferenceWithinCapacity() {
    boolean result1 = conference.addAttendeeToConference(student1);
    boolean result2 = conference.addAttendeeToConference(student2);
    List<Student> attendees = conference.getAttendees();

    assertTrue(result1);
    assertTrue(result2);
    assertEquals(2, attendees.size());
    assertTrue(attendees.contains(student1));
    assertTrue(attendees.contains(student2));
}

@Test
public void testAddAttendeeToConferenceExceedCapacity() {
    conference.addAttendeeToConference(student1);
    conference.addAttendeeToConference(student2);
    boolean result3 = conference.addAttendeeToConference(student3);
    List<Student> attendees = conference.getAttendees();

    assertTrue(result3);
    assertEquals(3, attendees.size());
    assertTrue(attendees.contains(student1));
    assertTrue(attendees.contains(student2));
    assertTrue(attendees.contains(student3));
    assertEquals(6, conference.getCapacity());
}

@Test
public void testCalculateTotalPricePaid() {
    conference.addAttendeeToConference(student1); // EMT discount
    conference.addAttendeeToConference(student2); // WEB discount
    conference.addAttendeeToConference(student3); // no discount
    double expectedPrice = (1 - EMT_DISCOUNT) * TICKET_PRICE +
        (1 - WEB_DISCOUNT) * TICKET_PRICE +
        TICKET_PRICE;

    double actualPrice = conference.calculateTotalPricePaid();
    assertEquals(expectedPrice, actualPrice, 0.01);
}

@Test
public void testTripleCapacityNotExceedLimit() {
    Conference smallConference = new Conference(3333);
    boolean result = smallConference.tripleCapacity();

    assertTrue(result);
    assertEquals(9999, smallConference.getCapacity());
}

```

```

@Test
public void testTripleCapacityExceedLimit() {
    Conference largeConference = new Conference(4000);
    boolean result = largeConference.tripleCapacity();

    assertFalse(result);
    assertEquals(4000, largeConference.getCapacity()); // (4000 * 3 =
12000)
}

@Test
public void testGetCapacity() {
    int capacity = conference.getCapacity();
    assertEquals(2, capacity);
}

@Test
public void testAddAttendeeToConferenceExceedingCapacity() {
    Conference largeConference = new Conference(3340);
    for (int i = 0; i < 3340; i++) {
        largeConference.addAttendeeToConference(new Student("Student" +
i, "LastName" + i, Course.OS, 20));
    }
    boolean result = largeConference.addAttendeeToConference(new
Student("Overflow", "Student", Course.OS, 20));
    assertFalse(result); // (3340 * 3 = 10020)
}

@Test
public void testTripleCapacityWithinLimit() {
    Conference conference = new Conference(2000);
    boolean result = conference.tripleCapacity();
    assertTrue(result);
    assertEquals(6000, conference.getCapacity());
}

// TESTS FOR STUDENT CLASS

@Test
public void testStudentGetName() {
    assertEquals("John", student1.getName());
}

@Test
public void testStudentSetName() {
    student1.setName("Jane");
    assertEquals("Jane", student1.getName());
}

@Test
public void testStudentGetSurname() {
    assertEquals("Doe", student1.getSurname());
}

```

```
@Test
public void testStudentSetSurname() {
    student1.setSurname("Smith");
    assertEquals("Smith", student1.getSurname());
}

@Test
public void testStudentGetCourse() {
    assertEquals(Course.EMT, student1.getCourse());
}

@Test
public void testStudentSetCourse() {
    student1.setCourse(Course.WEB);
    assertEquals(Course.WEB, student1.getCourse());
}

@Test
public void testStudentGetAge() {
    assertEquals(20, student1.getAge());
}

@Test
public void testStudentSetAge() {
    student1.setAge(25);
    assertEquals(25, student1.getAge());
}

@Test
public void testStudentToString() {
    String expected = "Student{name='John', surname='Doe', course=EMT,
age=20}";
    assertEquals(expected, student1.toString());
}
}
```

Cover: ConferenceTest

✓

Tests passed: 23 of 23 tests – 52 ms

✓

ConferenceTest (homework) 52 ms

testAddAttendeeToConferenceExceedLimit 25 ms

testAddAttendeeToConferenceWithDis 0 ms

testTripleCapacityExceedLimit 0 ms

testAddAttendeeToConferenceExceed 8 ms

testTripleCapacityWithinLimit 0 ms

testTripleCapacityEdgeCase 0 ms

testStudentSetSurname 0 ms

testAddAttendeeToConferenceWithinC 0 ms

testStudentSetName 0 ms

testTripleCapacityNotExceedLimit 0 ms

testAddAttendeeToConferenceExceed 0 ms

testCalculateTotalPricePaidWithWebC 0 ms

testStudentGetAge 0 ms

testGetAttendees 0 ms

testCalculateTotalPricePaid 0 ms

testStudentToString 19 ms

testStudentSetAge 0 ms

testGetCapacity 0 ms

testCalculateTotalPricePaidForWebDis 0 ms

testStudentGetSurname 0 ms

testStudentSetCourse 0 ms

testStudentGetCourse 0 ms

testStudentGetName 0 ms

C:\Users\pc\.jdk\bin\java.exe ...
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
homework\...
exclude patterns:
exclude annotations patterns:
.*Generated.*
Class transformation time: 0.0274624s for 524 classes or 5.2409160305343516E-5s per class
Process finished with exit code 0

Project pitest_hw C:\Users\pc\Desktop\pitest_hw\pitest_hw

pitest_hw

src

homework: 100% classes, 100% lines covered

Conference: 100% methods, 100% lines covered

ConferenceTest: 100% methods, 100% lines covered

Course: 100% methods, 100% lines covered

Student: 100% methods, 100% lines covered

pitest_hw.iml

External Libraries

Scratches and Consoles

Coverage: ConferenceTest

Element	Class, %	Method, %	Line, %
homework	100% (4/4)	100% (43/43)	100% (135/135)

Coverage: ConferenceTest

Element

Class, %

Method, %

Line, %

homework

Course

Student

Conference

ConferenceTest

100% (4/4)

100% (1/1)

100% (1/1)

100% (1/1)

100% (1/1)

100% (43/43)

100% (2/2)

100% (10/10)

100% (7/7)

100% (24/24)

100% (135/135)

100% (6/6)

100% (14/14)

100% (25/25)

100% (90/90)

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
3	100% <div>159/159</div>	46% <div>34/74</div>	46% <div>34/74</div>

Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage	Test Strength
homework	3	100% <div>159/159</div>	46% <div>34/74</div>	46% <div>34/74</div>

Pit Test Coverage Report

Package Summary

homework

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
3	100% <div>159/159</div>	46% <div>34/74</div>	46% <div>34/74</div>

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
Conference.java	100% <div>26/26</div>	100% <div>23/23</div>	100% <div>23/23</div>
ConferenceTest.java	100% <div>114/114</div>	13% <div>6/46</div>	13% <div>6/46</div>
Student.java	100% <div>19/19</div>	100% <div>5/5</div>	100% <div>5/5</div>

Conference.java

```
1  package homework;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Conference {
7
8      public static double TICKET_PRICE = 10.90;
9      public static double EMT_DISCOUNT = 0.5;
10     public static double WEB_DISCOUNT = 0.3;
11
12     private List<Student> attendees;
13     private int capacity;
14
15     public Conference(int capacity) {
16         this.attendees = new ArrayList<>(capacity);
17         this.capacity = capacity;
18     }
19
20     public double calculateTotalPricePaid() {
21         double price = 0;
22
23         for (Student p : attendees) {
24             if (p.getCourse().name().equals("EMT"))
25                 price += (1 - EMT_DISCOUNT) * TICKET_PRICE;
26             else if (p.getCourse().name().equals("WEB"))
27                 price += (1 - WEB_DISCOUNT) * TICKET_PRICE;
28             else
29                 price += TICKET_PRICE;
30         }
31         return price;
32     }
33
34     public boolean addAttendeeToConference(Student student) {
35         if ((this.attendees.size() < capacity) || tripleCapacity()) {
36             this.attendees.add(student);
37             return true;
38         } else return false;
39     }
40
41     public boolean tripleCapacity() {
42         if (this.capacity * 3 > 9999)
43             return false;
44         this.capacity *= 3;
45         return true;
46     }
47
48     public List<Student> getAttendees() {
49         return attendees;
50     }
51
52     public int getCapacity() {
53         return capacity;
54     }
55 }
```

Mutations

24	1. negated conditional → KILLED
25	1. Replaced double addition with subtraction → KILLED 2. Replaced double multiplication with division → KILLED 3. Replaced double subtraction with addition → KILLED
26	1. negated conditional → KILLED
27	1. Replaced double addition with subtraction → KILLED 2. Replaced double multiplication with division → KILLED 3. Replaced double subtraction with addition → KILLED
29	1. Replaced double addition with subtraction → KILLED
31	1. replaced double return with 0.0d for homework/Conference::calculateTotalPricePaid → KILLED
35	1. negated conditional → KILLED 2. changed conditional boundary → KILLED 3. negated conditional → KILLED
37	1. replaced boolean return with false for homework/Conference::addAttendeeToConference → KILLED
38	1. replaced boolean return with true for homework/Conference::addAttendeeToConference → KILLED
42	1. negated conditional → KILLED 2. Replaced integer multiplication with division → KILLED 3. changed conditional boundary → KILLED
43	1. replaced boolean return with true for homework/Conference::tripleCapacity → KILLED
44	1. Replaced integer multiplication with division → KILLED
45	1. replaced boolean return with false for homework/Conference::tripleCapacity → KILLED
49	1. replaced return value with Collections.emptyList for homework/Conference::getAttendees → KILLED
53	1. replaced int return with 0 for homework/Conference::getCapacity → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- homework.ConferenceTest.testGetAttendees(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testAddAttendeeToConferenceWithinCapacity(homework.ConferenceTest) (3 ms)
- homework.ConferenceTest.testAddAttendeeToConferenceExceedCapacity(homework.ConferenceTest) (12 ms)
- homework.ConferenceTest.testTripleCapacityWithinLimit(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testTripleCapacityNotExceedLimit(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testTripleCapacityExceedLimit(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testTripleCapacityEdgeCase(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testCalculateTotalPricePaid(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testAddAttendeeToConferenceExceedingCapacity(homework.ConferenceTest) (3 ms)
- homework.ConferenceTest.testAddAttendeeToConferenceExceedMaxCapacity(homework.ConferenceTest) (17 ms)
- homework.ConferenceTest.testCalculateTotalPricePaidForWebDiscount(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testAddAttendeeToConferenceWithDifferentCourse(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testCalculateTotalPricePaidWithWebCourse(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testGetCapacity(homework.ConferenceTest) (2 ms)

Student.java

```
1  package homework;
2
3  public class Student {
4
5      private String name;
6      private String surname;
7      private Course course;
8      private int age;
9
10     public Student(String name, String surname, Course course, int age) {
11         this.name = name;
12         this.surname = surname;
13         this.course = course;
14         this.age = age;
15     }
16
17     public String getName() {
18 1     return name;
19     }
20
21     public void setName(String name) {
22         this.name = name;
23     }
24
25     public String getSurname() {
26 1     return surname;
27     }
28
29     public void setSurname(String surname) {
30         this.surname = surname;
31     }
32
33     public Course getCourse() {
34 1     return course;
35     }
36
37     public void setCourse(Course course) {
38         this.course = course;
39     }
40
41     public int getAge() {
42 1     return age;
43     }
44
45     public void setAge(int age) {
46         this.age = age;
47     }
48
49     @Override
50     public String toString() {
51 1     return "Student{" +
52         "name='" + name + '\'' +
53         ", surname='" + surname + '\'' +
54         ", course=" + course +
55         ", age=" + age +
56         '}';
57     }
58 }
```

Mutations

18	1. replaced return value with "" for homework/Student::getName → KILLED
26	1. replaced return value with "" for homework/Student::getSurname → KILLED
34	1. replaced return value with null for homework/Student::getCourse → KILLED
42	1. replaced int return with 0 for homework/Student::getAge → KILLED
51	1. replaced return value with "" for homework/Student::toString → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- homework.ConferenceTest.testStudentGetCourse(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testCalculateTotalPricePaidForWebDiscount(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testStudentSetCourse(homework.ConferenceTest) (3 ms)
- homework.ConferenceTest.testCalculateTotalPricePaidWithWebCourse(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testCalculateTotalPricePaid(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testStudentGetName(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testStudentSetName(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testStudentGetSurname(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testStudentSetSurname(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testStudentGetAge(homework.ConferenceTest) (1 ms)
- homework.ConferenceTest.testStudentSetAge(homework.ConferenceTest) (2 ms)
- homework.ConferenceTest.testStudentToString(homework.ConferenceTest) (15 ms)

Изработил:
Бојан Ристов (211151)