# 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;
    }
}
```

## 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 double calculateTotalPricePaid() {
        this.attendees.add(student);
public boolean tripleCapacity() {
public List<Student> getAttendees() {
public int getCapacity() {
```

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

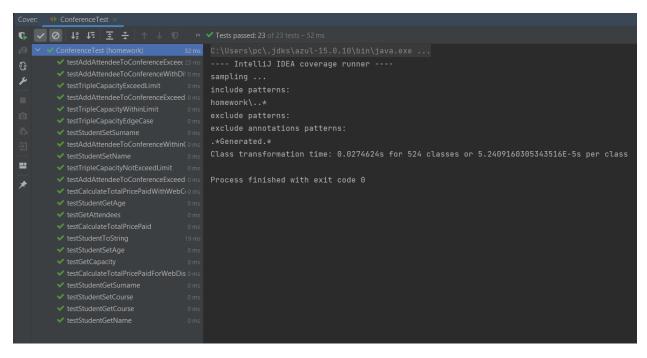
## ConferenceTest.java

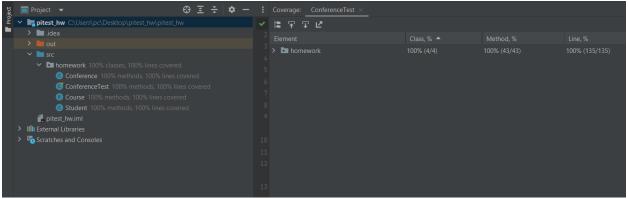
```
public void setUp() {
```

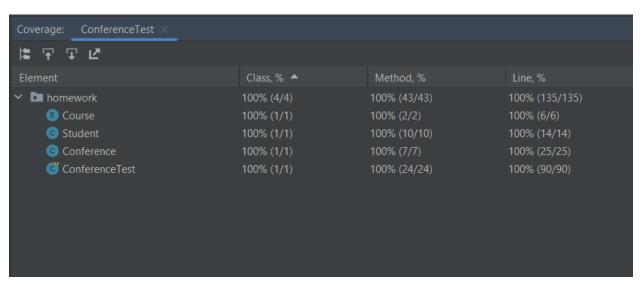
```
public void testAddAttendeeToConferenceWithinCapacity() {
    boolean result2 = conference.addAttendeeToConference(student2);
    List<Student> attendees = conference.getAttendees();
    assertTrue(result1);
    assertTrue(result2);
    assertTrue(attendees.contains(student1));
   assertTrue(attendees.contains(student2));
public void testAddAttendeeToConferenceExceedCapacity() {
    conference.addAttendeeToConference(student2);
    boolean result3 = conference.addAttendeeToConference(student3);
    List<Student> attendees = conference.getAttendees();
    assertTrue(result3);
   assertTrue(attendees.contains(student3));
public void testCalculateTotalPricePaid() {
    conference.addAttendeeToConference(student2); // WEB discount
   double actualPrice = conference.calculateTotalPricePaid();
   assertEquals(expectedPrice, actualPrice, 0.01);
    assertTrue(result);
```

```
public void testTripleCapacityExceedLimit() {
   assertFalse(result);
public void testGetCapacity() {
public void testAddAttendeeToConferenceExceedingCapacity() {
        largeConference.addAttendeeToConference(new Student("Student" +
   assertFalse(result); // (3340 * 3 = 10020)
public void testTripleCapacityWithinLimit() {
public void testStudentGetName() {
public void testStudentSetName() {
public void testStudentGetSurname() {
```

```
public void testStudentSetSurname() {
public void testStudentGetCourse() {
public void testStudentSetCourse() {
public void testStudentGetAge() {
public void testStudentSetAge() {
   String expected = "Student{name='John', surname='Doe', course=EMT,
   assertEquals(expected, student1.toString());
```







# **Pit Test Coverage Report**

## **Project Summary**

Number of Classes	s I	Line Coverage		utation Coverage	Test Strength		
3	100%	159/159	46%	34/74	46%	34/74	

### Breakdown by Package

Name Number of Classes	Line Coverage		Mutati	on Coverage	Test Strength	
homework 3	100%	159/159	46%	34/74	46%	34/74

# **Pit Test Coverage Report**

# **Package Summary**

### homework

Number of Classe	s I	Line Coverage		utation Coverage	Test Strength		
3	100%	159/159	46%	34/74	46%	34/74	

## **Breakdown by Class**

Name	Line Coverage		Mut	ation Coverage	Test Strength		
Conference.java	100%	26/26	100%	23/23	100% [	23/23	
ConferenceTest.java	100%	114/114	13%	6/46	13% [	6/46	
Student.java	100%	19/19	100%	5/5	100% [	5/5	

# Conference.java

```
1
   package homework;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Conference {
7
        public static double TICKET_PRICE = 10.90;
8
        public static double EMT_DISCOUNT = 0.5;
9
10
   public static double WEB_DISCOUNT = 0.3;
11
12
        private List<Student> attendees;
13
        private int capacity;
14
15
        public Conference(int capacity) {
           this.attendees = new ArrayList<>(capacity);
16
17
            this.capacity = capacity;
18
19
20
        public double calculateTotalPricePaid() {
21
           double price = 0;
22
23
            for (Student p : attendees) {
241
                if (p.getCourse().name().equals("EMT"))
                    price += (1 - EMT_DISCOUNT) * TICKET_PRICE;
253
261
                else if (p.getCourse().name().equals("WEB"))
                    price += (1 - WEB_DISCOUNT) * TICKET_PRICE;
27 3
                else
28
291
                    price += TICKET_PRICE;
30
31 1
            return price;
32
33
34
        public boolean addAttendeeToConference(Student student) {
35 3
            if ((this.attendees.size() < capacity) || tripleCapacity()) {
                this.attendees.add(student);
36
37 1
                return true;
381
            } else return false;
39
40
41
        public boolean tripleCapacity() {
423
           if (this.capacity * 3 > 9999)
431
                return false;
44 1
            this.capacity *= 3;
451
            return true;
46
47
48
        public List<Student> getAttendees() {
491
            return attendees;
50
51
52
        public int getCapacity() {
531
            return capacity;
54
55 }
```

#### Mutations

- 24 1. negated conditional → KILLED
  - Replaced double addition with subtraction → KILLED
- Replaced double multiplication with division → KILLED
  - Replaced double subtraction with addition → KILLED
- 26 1. negated conditional → KILLED
  - Replaced double addition with subtraction → KILLED
- 27 2. Replaced double multiplication with division → KILLED
  - 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
  - negated conditional → KILLED
- 35 2. changed conditional boundary → KILLED
  - negated conditional → KILLED
- replaced boolean return with false for homework/Conference::addAttendeeToConference → KILLED
- 38 1. replaced boolean return with true for homework/Conference::addAttendeeToConference → KILLED
  - negated conditional → KILLED
- 42 2. Replaced integer multiplication with division → KILLED
  - 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\_RĒTURNS
- 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.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;
       private int age;
8
9
10 public Student(String name, String surname, Course course, int age) {
    this.name = name;
11
12
   this.surname = surname;
13
    this.course = course;
14
   this.age = age;
15 }
16
17
       public String getName() {
181
        return name;
19
20
21
       public void setName(String name) {
22
           this.name = name;
23
24
25
       public String getSurname() {
261
           return surname;
27
28
       public void setSurname(String surname) {
29
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
       public String toString() {
50
       return "Student{" +
51 1
                  "name='" + name + '\'' +
52
                  ", surname="" + surname + "\" +
53
                  ", course=" + course +
54
                  ", age=" + age +
55
                  .).3
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 RĒTURNS
- PRIMITIVE RETURNS
- TRUE\_RETŪRNS
- 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)