Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический университет

“ЛЭТИ” им.В.И.Ульянова (Ленина) »

Кафедра МОЭВМ

**ОТЧЕТ**

**по лабораторно-практической работе № 9**

**по дисциплине «Объектно - ориентированное программирование на языке Java»**

Выполнил Баймухамедов Р. Р.

Преподаватель Павловский М.Г

Факультет КТИ

Группа № 3311

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024 г

**Цель работы**

Знакомство с технологией модульного тестирования Java-приложения с использование системы JUnit.

**Описание задания**

1. Создать новый проект, который будет дублировать проект одной из предыдущих лабораторных работ.

2. Проанализировать классы приложения и определить, какие методы необходимо протестировать.

3. Написать JUnit-тесты для выбранных методов.

4. Запустить тесты и снять с экрана скриншоты, иллюстрирующие выполнение тестов.

5. Сгенерировать документацию с помощью Javadoc и просмотрите ее в браузере.

**Репозиторий**

<https://github.com/brick1ng5654/course-2/tree/main/OOP/lab_09>

В этом репозитории находятся исходные файлы проекта:

В /src/edu/java/lab09 находится файл CinemaList.java

В /doc находится документация, сгенерированная JavaDoc

Ссылка на видеоотчёт

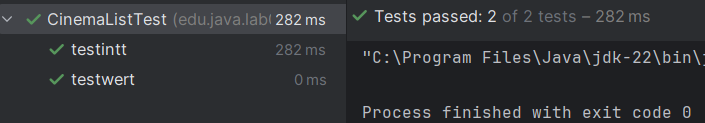
<https://youtu.be/xlqYBVKc06I>

**Перечень методов, которые тестируются в приложении**

В приложении будут тестироваться методы intt и wert класса CinemaList. Они будут тестироваться методами testintt и testwert класса CinemaListTest. Тестирующий метод testintt буте проверять в методе intt строку getWordToFind на наличие строки "Ключевое слово". Метод testwert буте проверять в методе wert соответствие переменной fileNameOpen на начальное null значение.

**Работоспособность приложения и скриншоты, иллюстрирующие выполнение тестов**

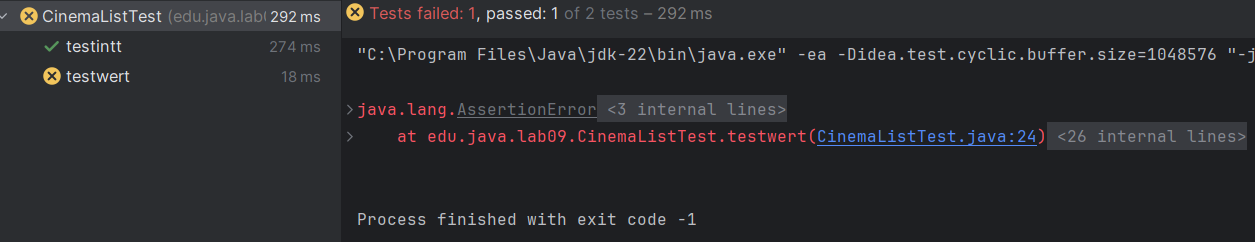
Запустим CinemaListTest и получим следующее



Пример

В примере 1 все тесты выполнены успешно

Теперь изменим метод wert значение переменной fileNameOpen с null на “fail” и в результате мы должны получить ошибку. Проверим это



Пример

Как можно заметить результат соответствует нашему ожиданию, значит программа работает как надо.

**Вывод**

Во время выполнения данной лабораторной работы, были получены навыки в технологии модульного тестирования Java-приложения с использование системы JUnit.

**Исходные тексты классов тестов**

*Код класса тестов*

package edu.java.lab09;  
import org.junit.Assert;  
import org.junit.Before;  
import org.junit.Test;  
  
  
public class CinemaListTest {  
 private CinemaList cinemaList;  
  
 @Before  
 public void setUp() {  
 cinemaList = new CinemaList();  
 cinemaList.initializeComponents();  
 }  
  
 @Test  
 public void testintt() {  
 cinemaList.getWordToFind().setText("Введите данные");  
 Assert.*assertTrue*(cinemaList.intt());  
 }  
  
 @Test  
 public void testwert() {  
 Assert.*assertTrue*(cinemaList.wert());  
 }  
}

*Код тестирующих методов*

public boolean intt() {  
 return "Введите данные".equals(wordToFind.getText());  
}  
  
public boolean wert() {  
 return fileNameOpen == "fail";  
}  
  
public JTextField getWordToFind() {  
 return wordToFind;  
}  
  
public String getFileNameOpen() {  
 return fileNameOpen;  
}