

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
«Модульное тестирование приложения»
по дисциплине «Объектно - ориентированное
программирование на языке Java»

Выполнил Сапронов К.Д.

Факультет КТИ

Группа № 3311

Подпись преподавателя _____

Санкт-Петербург

2024 г

Цель работы

Знакомство с технологией модульного тестирования Java-приложений с использованием системы JUnit.

Описание задания

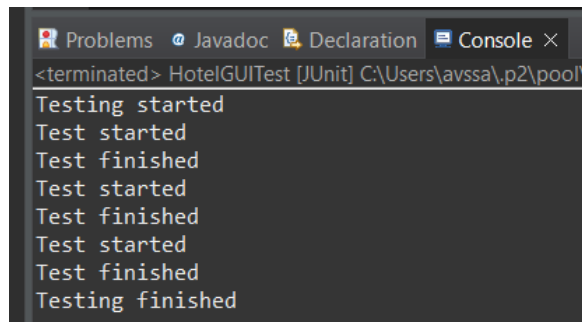
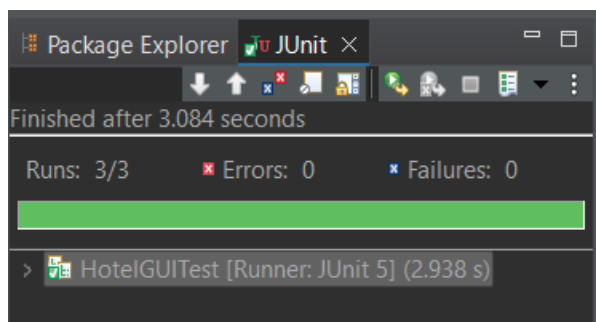
1. Проанализировать классы приложения из л/р №3 и определить, какие методы необходимо протестировать.
2. Написать JUnit-тесты для выбранных методов.
3. Запустить тесты и проверить их выполнение.

Разработанные методы

Для этой работы был разработан отдельный класс HotelGUITest, в котором определены методы:

1. testAddGuest() – проверяет добавление новой строки в таблицу с помощью проверки изменения количества строк и соответствия содержимого последней строки и добавляемых данных.
2. testSortByName() – проверяет сортировку по имени с помощью проверки данных до и после сортировки.
3. testException() – проверяет корректность работы программы при возникновении исключительной ситуации, проверяя выбрасывание верного исключения в такой ситуации.

Выполнение тестов



Ссылки

https://drive.google.com/drive/folders/1SkmitiMaArA7aWjd8Q5cThXVNP8ws_N4?usp=drive_link

В этой папке будут находиться все лабораторные работы

В папке lab9 находятся этот отчет, видеоотчет и папка lab09, в которой находятся файлы проекта и документация javadoc.

Текст класса

```
package lab09;

import org.junit.*;

import javax.swing.table.DefaultTableModel;

public class HotelGUITest {

    @Test
    public void testAddGuest() {
        GUI gui = new GUI();
        gui.buildAndShowGUI();

        DefaultTableModel model = gui.tableModel;
        int initialRowCount = model.getRowCount();

        model.addRow(new Object[]{"Alice Brown", "305", "12.12.2024", "20.12.2024"});

        Assert.assertEquals(initialRowCount + 1, model.getRowCount());

        Object[] lastRow = {
            model.getValueAt(initialRowCount, 0),
            model.getValueAt(initialRowCount, 1),
            model.getValueAt(initialRowCount, 2),
            model.getValueAt(initialRowCount, 3)
        };
        Assert.assertArrayEquals(new Object[]{"Alice Brown", "305", "12.12.2024",
"20.12.2024"}, lastRow);
    }

    @Test
    public void testSortByName() {
        GUI gui = new GUI();
        gui.buildAndShowGUI();

        DefaultTableModel model = gui.tableModel;

        // Проверяем начальное состояние таблицы
        String firstGuestNameBeforeSort = (String) model.getValueAt(0, 0);
        String secondGuestNameBeforeSort = (String) model.getValueAt(1, 0);

        // Сортируем таблицу по имени
        gui.sortTable(0);

        // Проверяем, что таблица отсортирована
        String firstGuestNameAfterSort = (String) model.getValueAt(0, 0);
        String secondGuestNameAfterSort = (String) model.getValueAt(1, 0);
    }
}
```

```

    Assert.assertTrue(firstGuestNameAfterSort.compareTo(secondGuestNameAfterSort) <=
0);

    Assert.assertNotEquals(firstGuestNameBeforeSort, firstGuestNameAfterSort);
}

@Test(expected = RuntimeException.class)
public void testException() {
    throw new RuntimeException("Error encountered");
}
@BeforeClass
public static void allTestsStarted() {
    System.out.println("Testing started");
}
@AfterClass
public static void allTestsFinished() {
    System.out.println("Testing finished");
}
@Before
public void testStarted() {
    System.out.println("Test started");
}
@After
public void testFinished() {
    System.out.println("Test finished");
}
}

```