Минобрнауки России

Федеральное государственное автономное образовательное учреждение

высшего образования «Санкт-Петербургский государственный

электротехнический университет «ЛЭТИ»

им В. И. Ульянова (Ленина)»

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

**Лабораторная работа № 9**

**«Разработка интерфейса пользователя» по дисциплине**

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

Выполнил Авдеев Н. А.

Факультет КТИ

Группа № 3312

Санкт-Петербург

2024

**Цель работы:**

3накомство с правилами и классами построения парал лельных приложений в языке Java.

**Перечень методов, которые тестируются в приложении:**

import org.junit.jupiter.api.Test;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import java.io.File;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

import static org.junit.jupiter.api.Assertions.\*;

public class AppTestXml {

    // Тестирование метода checkName

    @Test

    public void testCheckName\_ValidInput() throws Exception {

        JTextField searchField = new JTextField("Espresso");

        App.checkName(searchField);  // Должен пройти без исключений

    }

    @Test

    public void testCheckName\_PlaceholderInput\_ShouldThrowMyException() {

        JTextField searchField = new JTextField("Поиск");

        Exception exception = assertThrows(App.MyException.class, () -> {

            App.checkName(searchField);

        });

        assertEquals("Вы не ввели название сорта кофе для поиска", exception.getMessage());

    }

    @Test

    public void testCheckName\_EmptyInput\_ShouldThrowNullPointerException() {

        JTextField searchField = new JTextField("");

        assertThrows(NullPointerException.class, () -> {

            App.checkName(searchField);

        });

    }

    // Тестирование загрузки из XML

    @Test

    public void testLoadFromXml() throws Exception {

        // Создаем временный XML файл с тестовыми данными

        File tempFile = File.createTempFile("test-coffee", ".xml");

        tempFile.deleteOnExit();

        String testXml = """

            <coffees>

                <coffee>

                    <type>Espresso</type>

                    <country>Colombia</country>

                    <roast>Dark</roast>

                    <grade>88</grade>

                </coffee>

                <coffee>

                    <type>Cappuccino</type>

                    <country>Brazil</country>

                    <roast>Medium</roast>

                    <grade>85</grade>

                </coffee>

            </coffees>

            """;

        Files.writeString(Paths.get(tempFile.getPath()), testXml);

        // Создаем пустую модель таблицы

        String[] columnNames = {"Сорт кофе", "Страна произрастания", "Степень обжарки", "Оценка Q-грейдера"};

        DefaultTableModel model = new DefaultTableModel(columnNames, 0);

        // Загружаем данные из XML файла

        App.loadTableFromXml(tempFile, model); // Передаем файл и модель таблицы

        // Проверяем данные в модели таблицы

        assertEquals("Espresso", model.getValueAt(0, 0));

        assertEquals("Colombia", model.getValueAt(0, 1));

        assertEquals("Dark", model.getValueAt(0, 2));

        assertEquals("88", model.getValueAt(0, 3));

        assertEquals("Cappuccino", model.getValueAt(1, 0));

        assertEquals("Brazil", model.getValueAt(1, 1));

        assertEquals("Medium", model.getValueAt(1, 2));

        assertEquals("85", model.getValueAt(1, 3));

    }

**Скриншоты, иллюстрирующие работу тестов:**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Исходные тексты генерирования отчёта:**

public static void loadTableFromXml(File file, DefaultTableModel model) {

        try {

            // Проверяем, что файл не null и существует

            if (file == null || !file.exists()) {

                JOptionPane.showMessageDialog(null, "Ошибка: файл не найден.");

                return; // Прерываем загрузку

            }

            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

            DocumentBuilder builder = factory.newDocumentBuilder();

            Document document = builder.parse(file);

            Element root = document.getDocumentElement();

            NodeList coffeeNodes = root.getElementsByTagName("coffee");

            model.setRowCount(0); // Очищаем таблицу

            for (int i = 0; i < coffeeNodes.getLength(); i++) {

                Element coffee = (Element) coffeeNodes.item(i);

                String type = coffee.getElementsByTagName("type").item(0).getTextContent();

                String country = coffee.getElementsByTagName("country").item(0).getTextContent();

                String roast = coffee.getElementsByTagName("roast").item(0).getTextContent();

                String grade = coffee.getElementsByTagName("grade").item(0).getTextContent();

                model.addRow(new Object[]{type, country, roast, grade});

            }

        } catch (Exception ex) {

            ex.printStackTrace();

            JOptionPane.showMessageDialog(null, "Ошибка при чтении XML-файла: " + ex.getMessage());

        }

    }

static void checkName(JTextField cName) throws MyException, NullPointerException {

        String sName = cName.getText();

        if (sName.contains("Поиск")) throw new MyException();

        if (sName.isEmpty()) throw new NullPointerException();

    }

**}**

**Вывод:**

Модульные тесты подтвердили корректность работы методов checkName и loadTableFromXml. Приложение успешно валидирует пользовательский ввод и корректно обрабатывает данные из XML-файлов. Это повышает надежность и стабильность приложения.

**Ссылка на репозиторий**:

https://github.com/G3rb3r228/Labs\_OOP\_AvdeevNA-3312/tree/main/Lab09\_Avdeev3312