3 Лабораторная работа №3
Исследование способов модульного тестирования программного обеспечения

3.1. Цель работы

Исследовать основные подходы к модульному тестированию программного обеспечения. Приобрести практические навыки составления модульных тестов для объектно-ориентированных программ.

3.2. Постановка задачи

Был выдан вариант 14.

Выбрать в качестве тестируемого один из классов, спроектированных в лабораторной работе №1. Составить спецификацию тестового случая для одного из методов выбранного класса. Реализовать тестируемый класс и необходимое тестовое окружение. Выполнить тестирование с выводом результатов на экран и сохранением в log-файл. Проанализировать результаты тестирования, сделать выводы.

3.3. Ход выполнения работы

3.3.1. Спецификация тестового случая

Было принято решение протестировать класс MyLine.

Для тестируемого класса был определён тестовый случай: проверяется правильность работы метода containsThreeDigitNumber() - определить, содержит ли переданная строка трёхзначные числа. В тесте подаются значения, определённые в лабораторной работе №1.

3.3.2.Текст программы

На основе спецификации был создан тестовый драйвер LineModuleTest, реализованный в виде отдельного класса, вызывающего public-методы тестируемого класса. Текст драйвера представлен в листинге 1.

Листинг 1 — Тестовый драйвер LineModuleTest

```
package org.cory7666.softwaretestingexample;

import java.io.PrintStream;
import java.util.Collection;

import org.cory7666.softwaretestingexample.task3.MyLine;

public class LineModuleTest implements AutoCloseable
{
  private int testNumCounter;
  private final Collection<PrintStream> outputStreams;
  public LineModuleTest (Collection<PrintStream> outputStreams)
  {
    this.testNumCounter = 0;
    this.outputStreams = outputStreams;
  }
  public LineModuleTest test (String testCase, boolean expected)
  {
    ++testNumCounter;
    boolean result = new MyLine(testCase).containsThreeDigitNumber();
    outputStreams.forEach(writer -> {
      try
      {
        writer.println(String.format("-===        Тест %2d        ===-", testNumCounter));
        writer
          .println(
            String
              .format(
                "Строка: \"%s\". Ожидаемый результат: %b. Получено: %b.",
                testCase,
                expected,
                result));
        writer.println(String.format("-=== Завершение теста %2d ===-\n", testNumCounter));
      }
      catch (Exception ex) {}
    });
    return this;
  }
  @Override
  public void close () throws Exception
  {
    for (var stream : outputStreams)
    {
      try
      {
        stream.close();
      }
      catch (Exception ex)
      {}
    }
  }
}
```

3.3.3.Тестовый запуск драйвера и анализ полученных данных

После написания кода программа была скомпилирована и запущена. Рисунок 3.1 демонстрирует вывод тестового драйвера.

```
❯ java -jar out.jar
-===        Тест  1        ===-
Строка: "34". Ожидаемый результат: false. Получено: false.
-=== Завершение теста  1 ===-

-===        Тест  2        ===-
Строка: "Было подобрано число 33.". Ожидаемый результат: false. Получено: false.
-=== Завершение теста  2 ===-

-===        Тест  3        ===-
Строка: "101 далматинец". Ожидаемый результат: true. Получено: true.
-=== Завершение теста  3 ===-

-===        Тест  4        ===-
Строка: "2022 год". Ожидаемый результат: false. Получено: false.
-=== Завершение теста  4 ===-

-===        Тест  5        ===-
Строка: "13 литров 00509 миллилитров". Ожидаемый результат: true. Получено: true.
-=== Завершение теста  5 ===-

-===        Тест  6        ===-
Строка: "11 точка 0000900". Ожидаемый результат: true. Получено: true.
-=== Завершение теста  6 ===-
```

Рисунок 3.1 — Результат запуска тестового драйвера

Полученные данных были проанализированы и сравнены со спецификацией. Данные полностью соответствуют спецификации.

Выводы по результатам работы

При выполнении данный лабораторной работы были получены навыки составления модульных тестов. Было выделено, что модульные тесты необходимо применять при тестировании модуля программы, или, в случае этой работы, при тестировании класса.