

#### 4 Лабораторная работа №4

### Исследование способов интеграционного тестирования программного обеспечения

#### 4.1. Цель работы

Исследовать основные принципы интеграционного тестирования программного обеспечения. Приобрести практические навыки организации интеграционных тестов для объектно-ориентированных программ.

#### 4.2. Постановка задачи

Был выдан вариант 14.

Выбрать в качестве тестируемого взаимодействие двух или более классов, спроектированных в лабораторных работах №1 – 4. Составить спецификацию тестового случая. Реализовать тестируемые классы. Выполнить тестирование с выводом результатов на экран и сохранением в log-файл. Проанализировать результаты тестирования, сделать выводы.

#### 4.3. Ход выполнения работы

##### 4.3.1. Спецификация тестового случая

В соответствии с задачей была составлена следующая спецификация:

- Названия взаимодействующих классов: Matrix; MatrixColumn.
- Название теста: test\_replaceColumnData.
- Описание теста: тест проверяет метод replaceColumnWith — функцию замены данных указанного столбца в матрице. В тесте метод вызывается с двумя параметрами: 1 — номер заменяемого столбца и [0, 0, 0, 0] — массив новых данных

- Начальные условия: матрица  $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 78 & 11 \\ 1 & 100 & 5 \end{pmatrix}$ .
- Ожидаемый результат: матрица  $\begin{pmatrix} 0 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 78 & 11 \end{pmatrix}$ .

### 4.3.2. Текст программы

На основе спецификации был создан тестовый драйвер `MatrixIntegrateTest`, реализующий следующие методы:

- В конструктор тестового драйвера передаются начальная матрица и список потоков, в которые будут записаны результаты каждого теста.
- Сами методы, реализующие тестовые случаи.
- Метод `dump`, выводящий текущее содержимое матрицы в поток консольного вывода или файловый. Формат вывода: матрица до изменения, команда, матрица после изменения.

Полный текст драйвера представлен в листинге 1.

#### Листинг 1 — Тестовый драйвер `MatrixIntegrateTest`

```
package org.cory7666.softwaretestingexample;
import java.io.PrintStream;
import java.util.List;
import org.cory7666.softwaretestingexample.task1.Matrix;

public class MatrixIntegrateTest implements AutoCloseable
{
    private final List<PrintStream> streams;
    private final Matrix input;
    public MatrixIntegrateTest (Matrix input, List<PrintStream> streams)
    { this.streams = streams; this.input = input; }
    public MatrixIntegrateTest test (int column, int[] newData)
    {
        writeString("=== Начало теста. ===-");
        writeString("До изменения:"); dump();
        writeString(String.format("Замена данных в позиции %d.", column));
        input.replaceColumnWith(column, newData);
        writeString("После изменения:"); dump();
        writeString("=== Конец теста. ===-\n");
        return this;
    }
    private void dump ()
    { var matrixString = input.toString(); writeString(matrixString); }
    private void writeString (String x)
    { streams.forEach(stream -> stream.println(x)); }
    @Override
    public void close () throws Exception
    { streams.forEach(x -> x.close()); }
```

}

### 4.3.3. Тестовый запуск драйвера и анализ полученных данных

После написания кода программа была скомпилирована и запущена. Рисунок 4.1 демонстрирует вывод тестового драйвера.

```
> java -jar integrate-test.jar $(cat lab4-args)
-=== Начало теста. ===-
До изменения:
1      2      3
1      4      5
2      78     11
1      100    5
Замена данных в позиции 0.
После изменения:
0      2      3
0      4      5
0      78     11
0      100    5
-=== Конец теста. ===-
```

Рисунок 4.1 — Результаты запуска тестового драйвера

Полученные данные были проанализированы и сравнены со спецификацией. Результаты тестирования полностью соответствуют ожидаемым результатам.

### Выводы по результатам работы

При выполнении данной лабораторной работы были получены навыки составления интеграционных тестов. Было выделено, что интеграционные тесты следует применять для тестирования взаимодействия двух или нескольких модулей.