МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы №4

по дисциплине “Тестирование Программного Обеспечения”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил: доц. каф.

«Информационные системы»

Строганов В. А.

Севастополь

2022

**ЛАБОРАТОРНАЯ РАБОТА №4**

**«Исследование способов интеграционного тестирования программного обеспечения»**

**Цель работы**

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

**Задачи**

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

**Ход работы**

Для тестирования взаимодействия классов Matrix и calculation разработаем спецификацию тестового случая:

1) Названия взаимодействующих классов: Matrix, calculation.

2) Название теста: Test.

3) Описание теста: тест проверяет возможность создания объекта типа matriza в классе Marix, его инициализацию, использование его при вызове конструктора класса calculation для необходимых расчетов и дальний вывод результатов.

4) Начальные условия: матрица пуста, результирующий массив аналогично.

5) Ожидаемый результат: целочисленная квадратная матрица заполнена элементами. Определено произведение элементов в тех строках, которые не содержат отрицательных элементов. На данной основе был разработан тест – Test, который содержит вызов обоих классов. Этот класс содержит:

- Matrix A, где создаются объекты класса Matrix. Здесь происходит инициализация результирующего массива, запись данных в матрицу.

- calculation(A.matriza,A.massiv,A.N), где производятся расчеты произведения строк, и выбор подходящих.

- A.pechat(), где производится вывод результата на экран и запись в log-файл.

**Текст программы:**

#include <iostream>

#include<Windows.h>

#include <fstream>

using namespace std;

class calculation

{

public:

calculation(int\*\* matriza, int\* massiv, int N)

{

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

{

massiv[i] \*= matriza[i][j];

if (matriza[i][j] < 0)

{

massiv[i] = -1;

break;

}

}

}

};

class Matrix

{

public:

Matrix()

{

initialization();

read();

}

int N;

int\* massiv = new int[N];

int\*\* matriza = new int\*[N];

initialization()

{

cout << "Введите положительное количество целых элементов маcсива = ";

cin >> N;

for (int i = 0; i < N; i++)

massiv[i] = 1;

for (int i = 0; i < N; i++)

matriza[i] = new int[N];

}

read()

{

cout << "Введите значения " << N \* N << " элементов матрицы :\n";

for (int i = 0; i < N; i++)

for (int j = 0; j < N; j++)

cin >> matriza[i][j];

}

pechat()

{

ofstream Test;

Test.open("Test.log");

for (int i = 0; i < N; i++)

{

if (massiv[i] >= 0)

{

cout << "Произведение элементов в " << i << " строкe = " << massiv[i] << endl;

Test << "Произведение элементов в " << i << " строкe = " << massiv[i] << endl;

}

else

{

cout << "Строка " << i << " содержит отрицательные элементы\n";

Test << "Строка " << i << " содержит отрицательные элементы\n";

}

}

Test.close();

}

};

void Test(){

Matrix A;

calculation(A.matriza,A.massiv,A.N);

A.pechat();

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Test();

}

Таким образом, в ходе выполнения теста создается объект A класса matrix, содержащий

размерность матрицы, матрицу и вектор. Затем в конструктор класса расчетов calculation передаются необходимые аргументы класса A, после выполнения которого получаем вектор massiv[N] с решением задачи. В конце производится вызов метода pechat() печати и записи в log-файл.

Результат взаимодействия классов Matrix и calculation (рисунок 1).

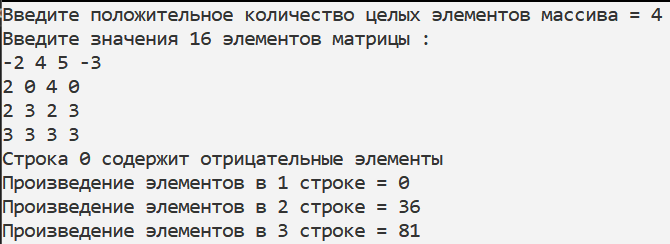


Рисунок 1 – Тест взаимодействия классов

Запись расчетов в log-файл (рисунок 2).

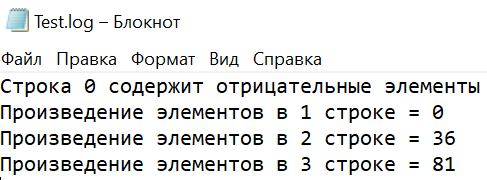


Рисунок 2 – Содержимое файла Test.log

На основе полученных результатов вычислений можно сказать об успешном выполнении интеграционного тестирования программного обеспечения.

.

**Выводы**

В ходе лабораторной работы были исследованы основные принципы интеграционного тестирования программного обеспечения: интеграционное тестирование представляет собой тестирование взаимодействующих модулей, объединяет от двух и более объектов тестируемой системы. Были приобретены практические навыки организации интеграционных тестов для объектно-ориентированных программ на примере двух классов: класс создания матрицы, ввода в нее данных, печати, а также класса, осуществляющего расчеты.