Министерство науки и высшего образования

Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Новосибирский государственный технический университет»



Кафедра теоретической и прикладной информатики

Лабораторная работа № 3  
по дисциплине «Метрология, качество и тестирование программного обеспечения»

**Модульное тестирование программ на языке C++ в среде Visual Studio**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Факультет: | ПМИ |  |  |
| Группа: | ПМИМ-01 |  |  |
| Студент: | Ничипиенко В.А. |  |  |
| Вариант: | 6 |  |  |
| Преподаватель: | Зайцев М.Г. |  |  |

Новосибирск

2021

1. **Цель работы**

Сформировать практические навыки разработки тестов и модульного тестирования на языке С++ с помощью средств автоматизации Visual Studio.

1. **Задание**

* Разработайте на языке С++ класс, содержащий набор функций в соответствии с вариантом задания.
* Разработайте тестовые наборы данных по критерию С2 для тестирования функций класса.
* Протестировать функции с помощью средств автоматизации модульного тестирования Visual Studio.
* Провести анализ выполненного теста и, если необходимо отладку кода.
* Написать отчёт о результатах проделанной работы.

|  |  |
| --- | --- |
| **Вариант**  **№** | **Описание функции** |
| 6(1) | Функция упорядочивает значения переменных x,y,z в порядке убывания их значений, так чтобы x>=y>=z. |
| Функция получает два положительных целых числа a и b Вычисляет и возвращает их наибольший общий делитель. |
| Функция получает целое числа a. Формирует и возвращает целое число b из значений чётных разрядов целого числа a. Например: a = 12345, b = 24. |
| Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает сумму нечётных значений компонентов массива, лежащих выше главной диагонали |

1. **Управляющий граф программы**

*Функция 1*

1. tuple<int, int, int> MyClass::ArrangeNumbers(int x, int y, int z){
2. int first = min(x, min(y, z));
3. int second = max(max(min(x, y), min(y, z)), min(x, z));
4. int third = max(max(x, y), z);
5. return tuple<int, int, int>(third, second, first);}

*Функция 2*

1. int MyClass::FindGreatestCommonDivisor(int a, int b){
2. int div;
3. if (a == b)
4. {return a;}
5. int d = a - b;
6. if (d < 0)
7. {d = -d;
8. div = FindGreatestCommonDivisor(a, d);}
9. else div = FindGreatestCommonDivisor(b, d);
10. return div;}

*Функция 3*

1. int MyClass::FormNumber(int a)
2. {string result = "";
3. string inputStr = to\_string(a);
4. for (int i = 0; i < inputStr.size(); i++)
5. if ((int)inputStr[i] % 2 == 0)
6. result += inputStr[i];
7. return stoi(result);}

*Функция 4*

1. float MyClass::SumOddElemintsTopMatrix(int(\*numbers)[3], int rowsCount){
2. float result = 0.0f;
3. int columnsCount = sizeof(\*numbers) / sizeof(\*numbers[0]);
4. for (int i = 0; i < rowsCount; i++)
5. for (int j = i + 1; j < columnsCount; j++)
6. if (numbers[i][j] % 2 != 0)
7. result += numbers[i][j];
8. cout << endl;
9. return result;}

1. **Тестовые наборы данных для тестирования функций класса**

*Функция 1*

|  |  |
| --- | --- |
| № теста | Входные данные |
| 1 | (1, 3, 2) |
| 2 | (1, 1, 1) |
| 3 | (1, 2, 3) |

*Функция 2*

|  |  |
| --- | --- |
| № теста | Входные данные |
| 1 | (5, 20) |
| 2 | (11, 7) |

*Функция 3*

|  |  |
| --- | --- |
| № теста | Входные данные |
| 1 | (1, 2, 3, 4, 5, 6, 7, 8, 9) |
| 2 | (1, 3, 5, 7, 9) |

*Функция 4*

|  |  |
| --- | --- |
| № теста | Входные данные |
| 1 | (1, 2, 3,  4, 5, 6,  7, 8, 9) |
| 2 | (1, 2, 2,  4, 5, 6,  7, 8, 9) |

1. **Исходные тексты программ**

*Текст программы MyClass.h*

#pragma once

#include <tuple>

using namespace std;

class MyClass

{

public:

static tuple<int, int, int> ArrangeNumbers(int x, int y, int z);

static int FindGreatestCommonDivisor(int a, int b);

static int FormNumber(int a);

static float SumOddElemintsTopMatrix(int(\*)[3], int);

};

*Текст программы MyClass.cpp*

#include "MyClass.h"

#include <iostream>

#include <string>

tuple<int, int, int> MyClass::ArrangeNumbers(int x, int y, int z)

{

int first = min(x, min(y, z));

int second = max(max(min(x, y), min(y, z)), min(x, z));

int third = max(max(x, y), z);

return tuple<int, int, int>(third, second, first);

}

int MyClass::FindGreatestCommonDivisor(int a, int b)

{

int div;

if (a == b)

{

return a;

}

int d = a - b;

if (d < 0)

{

d = -d;

div = FindGreatestCommonDivisor(a, d);

}

else

{

div = FindGreatestCommonDivisor(b, d);

}

return div;

}

int MyClass::FormNumber(int a)

{

string result = "";

string inputStr = to\_string(a);

for (int i = 0; i < inputStr.size(); i++)

{

if ((int)inputStr[i] % 2 == 0)

{

result += inputStr[i];

}

}

return stoi(result);

}

float MyClass::SumOddElemintsTopMatrix(int(\*numbers)[3], int rowsCount)

{

float result = 0.0f;

int columnsCount = sizeof(\*numbers) / sizeof(\*numbers[0]);

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

{

for (int j = i + 1; j < columnsCount; j++)

{

if (numbers[i][j] % 2 != 0)

{

result += numbers[i][j];

}

}

cout << endl;

}

return result;

}

*Текст программы ProjectTests.cpp*

#include "CppUnitTest.h"

#include "C:\Users\Виталий\source\repos\TestProject\TestProject\MyClass.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace ProjectTests

{

TEST\_CLASS(ProjectTests)

{

public:

TEST\_METHOD(METHOD1)

{

tuple<int, int, int> expected(3, 3, 1);

tuple<int, int, int> actual = MyClass::ArrangeNumbers(1, 3, 3);

Assert::AreEqual(get<0>(expected), get<0>(actual));

Assert::AreEqual(get<1>(expected), get<1>(actual));

Assert::AreEqual(get<2>(expected), get<2>(actual));

}

TEST\_METHOD(METHOD2)

{

int expected = MyClass::FindGreatestCommonDivisor(5, 20);

int actual = 5;

Assert::AreEqual(expected, actual);

}

TEST\_METHOD(METHOD3)

{

int c[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };

int rowsCount = sizeof(c) / sizeof(c[0]);

float actual = MyClass::SumOddElemintsTopMatrix(c, 3);

float expected = 3.0f;

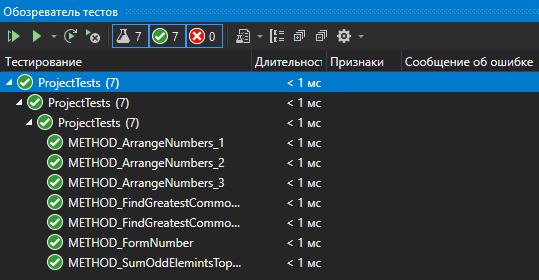
Assert::AreEqual(expected, actual);

}

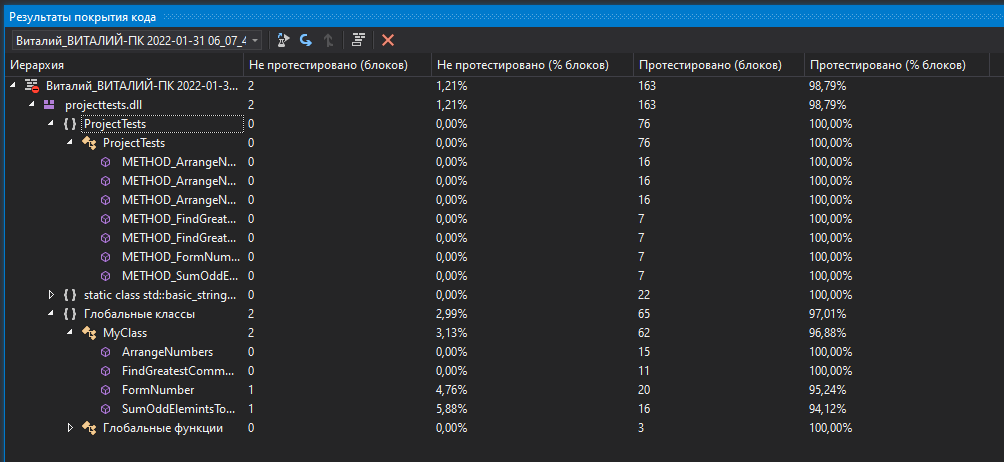
};

}

1. **Результаты выполнения модульных тестов**

****

1. **Результаты покрытия разработанного кода тестами**

****

1. **Выводы по выполненной работе**

В ходе выполнения работы были сформированы практические навыки разработки тестов и модульного тестирования на языке С++ с помощью средств автоматизации Visual Studio.