MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY «KHARKIV POLYTECHNICAL INSTITUTE»

DEPARTMENT OF SOFTWARE ENGINEERIG AND MANAGEMENT INFORMATION TECHNOLOGIES

REPORT
LABOROTORY WORK №5
«TESTING OF CODE»

Completed by a student
From CS-221v
Shuliopov Yehor Ruslanovych

Examine by
Associate professor
Lyutenko Iryna Viktorivna

KHARKIV 2021

Goal: Learning basic principles of testing C++ code

Tasks:

1. Study principles of using functions in C++.

2. Study Exception Handling in C++.

3. Modify the code from lab 2 according to 1 and 2 tasks.

4. Implement unit testing for developed program.

5. Make all necessary actions on github.com. Show the iteration where you made updating information.

6. Prepare the report of the work.

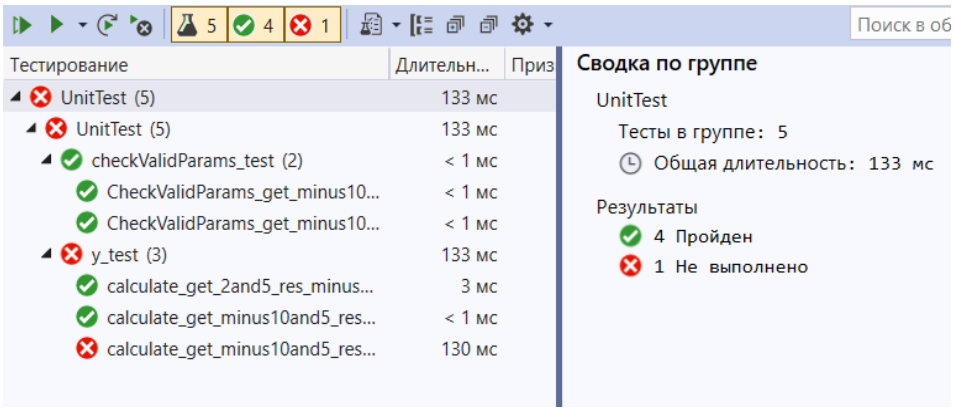
Progress

Several testing methods have been developed, both for the data corresponding to the logic of the task and for the deliberately false ones.

```cpp
#include <iostream>
#include <fstream>
using namespace std;

void checkValidInput(double x_1, double x_2, double step, int n) {
    if (cin.fail()) {
        cin.clear();
        while (cin.get() != '\n');

        throw "Incorrect input";
    }
    else if ((n < 1) || (step <= 0) || (x_1 > x_2)) {
        throw "input correct data";
    }
}

double y_1(double x_1, int n) {

    double y = 0; int i = 0;
    if (x_1 != 0) {
        for (i; i <= (n - 1); i++) {
            int j = 0;
            for (j; j <= (n - 1); j++) {
                y += 1 / (x_1 - i + x_1 * j);
            }
        }
        return y;
    }
    else {
        cout << "division by zero (x) " << x_1 << endl;
    }
}


double y_2(double x_1, int n) {

    double y = 1; int i = 1;
    for (i; i <= n; i++) {
        y *= (1 / x_1 - 1 / i);
    }
    return y;
}

double calculate(double x_1, double x_2, double step, int n)
{
    double y;
    while (x_1 <= x_2) {
        if (x_1 <= 0) {
            y = y_1(x_1, n);
        }
        else {
            y = y_2(x_1, n);

        }
        return y;
        x_1 += step;
    }
}
```

```cpp
int main() {

    setlocale(LC_ALL, "");
    bool outp_file;

tryAgain:
    try {
        double x_1;
        cout << "x_1 : ";
        cin >> x_1;

        double x_2;
        cout << "x_2 : ";
        cin >> x_2;


        double step;
        cout << "step : ";
        cin >> step;

        int n;
        cout << "n : ";
        cin >> n;
        checkValidInput(x_1, x_2, step, n);

        cout << "Вывод результата в файл?\n" << "любое число - да\n" << "0 - нет\n";
        cin >> outp_file;
        ofstream outf("result.txt");


        for (; x_1 <= x_2; x_1 += step) {
            if (outp_file == false)
                cout << "x = " << x_1 << " " << "y = " << calculate(x_1, x_2, step, n) << endl;
            else
                outf << "x = " << x_1 << " " << "y = " << calculate(x_1, x_2, step, n) << endl;
        }
    }
    catch (...) {
        cout << "data must be ((n >= 1) || (step > 0) || (x_1 <= x_2))" << endl;
        goto tryAgain;
    }
    return 0;
}
```

Pic. 1.1 «program code»



Pic.1.2 «Results of tests»

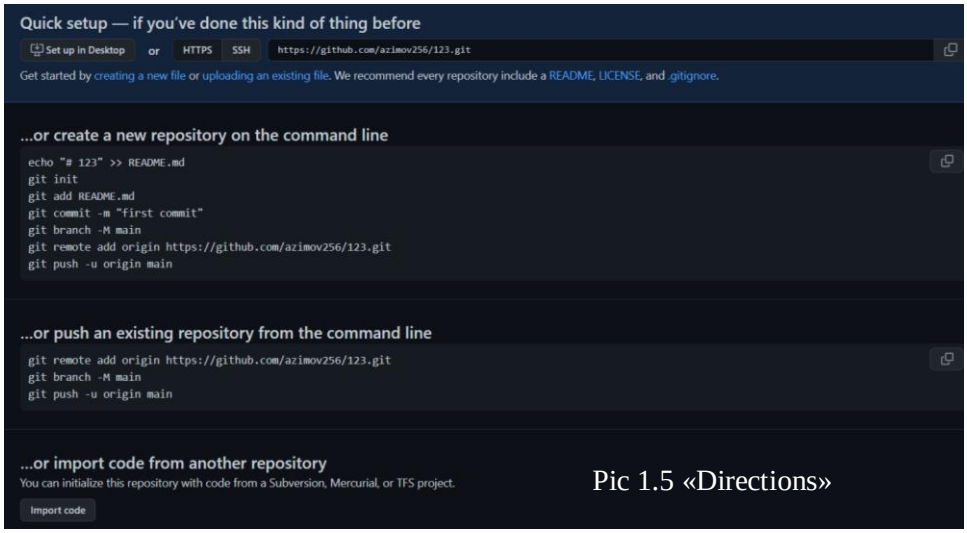
```cpp
#include "pch.h"
#include "CppUnitTest.h"
#include "C:\Users\esulu\Desktop\folders\1 курс\swe\практические лабы\лр_3_

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest
{
    TEST_CLASS(y_test)
    {
    public:

        TEST_METHOD(calculate_get_2and5_res_minus0point03125)
        {
            double x_1 = 2;
            int n = 5;
            double exp_ed = -0.03125;
            double actual = y_2(x_1, n);
            Assert::AreEqual(exp_ed, actual);
        }


        TEST_METHOD(calculate_get_minus10and5_res_minus0point1)
        {
            double x_1 = -10;
            int n = 1;
            double exp_ed = -0.1;
            double actual = y_1(x_1, n);
            Assert::AreEqual(exp_ed, actual);
        }


        TEST_METHOD(calculate_get_minus10and5_res_minus0point25)
        {
            double x_1 = -10;
            int n = 1;
            double exp_ed = -0.25;
            double actual = y_1(x_1, n);
            Assert::AreEqual(exp_ed, actual);
        }

    };
```

```cpp
TEST_CLASS(checkValidParams_test)
{
public:

    TEST_METHOD(CheckValidParams_get_minus10_20_3_5_exceptionNotThrown)
    {
        double x_1 = -10;
        double x_2 = 20;
        double step = 3;
        int n = 5;
        try {
            checkValidInput(x_1, x_2, step, n);
            Assert::IsTrue(true);
        }
        catch (...){
            Assert::Fail();
        }
    }

public:
    TEST_METHOD(CheckValidParams_get_minus10_minus11_3_5_exceptionThrown)
    {
        double x_1 = -10;
        double x_2 = -11;
        double step = 1;
        int n = 5;
        try {
            checkValidInput(x_1, x_2, step, n);
            Assert::Fail();
        }
        catch (...) {
            Assert::IsTrue(true);
        }
    }
};
```

Рис.1.3 «test methods code»

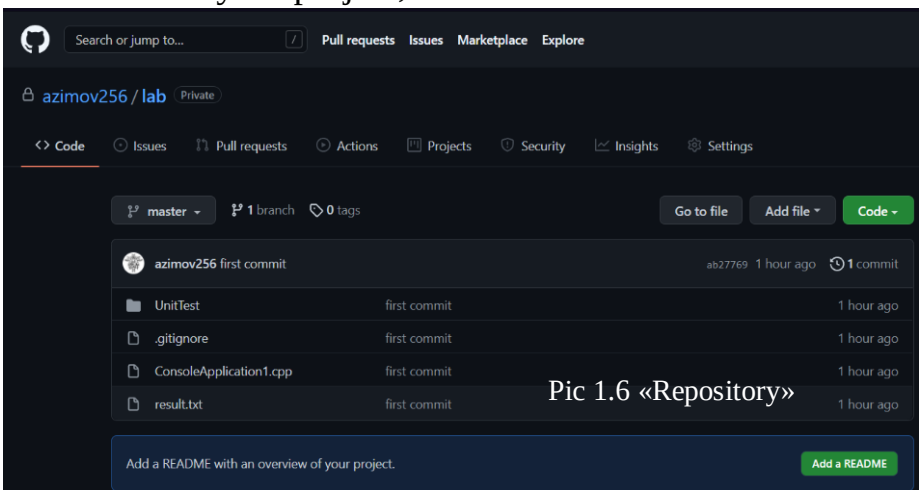
Converting code in sharing segment

1. Sign up at Github.com;
2. Download and install the “git”;
3. Create new repository at Github.com;
4. Open folder with your code and tests;
5. Create file “.ignore” and indicate in it all unwanted files;
6. Register in the file path “cmd”;
7. In command line you need to follow all ”github’s” directions;



Pic 1.5 «Directions»

8. And u unload your project;



Pic 1.6 «Repository»

9. Repository -> Settings -> Manage access -> Add people (in case of sharing using your project)

Conclusion: In the course of my work I developed practical skills of software testing in Visual Studio environments. During laboratory work, I got an idea with the methods of testing logical programs, creating a formal description of the test results conducted in real console application code.