**Лабораторная работа 3**

Создание Unit-теста в MS Visual Studio — C#

Студент Венедиктов Никита

ФПМИ, ПИ, 2 курс, 13 группа

Ссылка на **Github** с кодом(ветка lab3): <https://github.com/NikitaVen/Testing>

/\*Для чего нужны Unit-test-ы\*/**Unit-test-ы** позволяют проверить на корректность отдельные модули исходного кода программы, протестировать программу модульно. Они упрощают поиск ошибок в коде и проверку на то, не привели ли новые изменения кода к его регрессии.

//Ход работы:

Код файла *Program.cs,* содержащий тестируемую функцию Min:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace MinApp

{

public class Program

{

public static int Min(int a, int b, int c)

{

int min = a;

if (min > b) min = b;

if (min > c) min = c;

return min;

}

static void Main(string[] args)

{

Console.WriteLine("Demo of Unit-testing in C#.");

}

}

}

/\*Метод(функция) Min объявлен public и static, чтобы его можно было тестировать и использовать в целом без создания объекта класса Program\*/

/\*В файле тестирующей программы *UnitTest1.cs* создаем 3 Unit-test-а TestMin1, TestMin2, TestMin3\*/

Код файла *UnitTest1.cs* тестирующей программы:

using System;

using System.Text;

using System.Collections.Generic;

using System.Linq;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using MinApp;

namespace TestMinApp

{

[TestClass]

public class UnitTest1

{

private TestContext testContextInstance;

public TestContext TestContext

{

get

{

return testContextInstance;

}

set

{

testContextInstance = value;

}

}

[TestMethod]

public void TestMin1()

{

int min;

min = Program.Min(3, 4, 5);

Assert.AreEqual(3, min);

}

[TestMethod]

public void TestMin2()

{

int min;

min = Program.Min(10, 3, -3);

Assert.AreEqual(-3, min);

}

[TestMethod]

public void TestMin3()

{

int min;

min = Program.Min(10, 2+1, 5);

Assert.AreEqual(3, min);

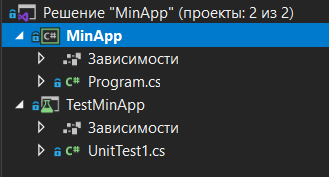
}

}

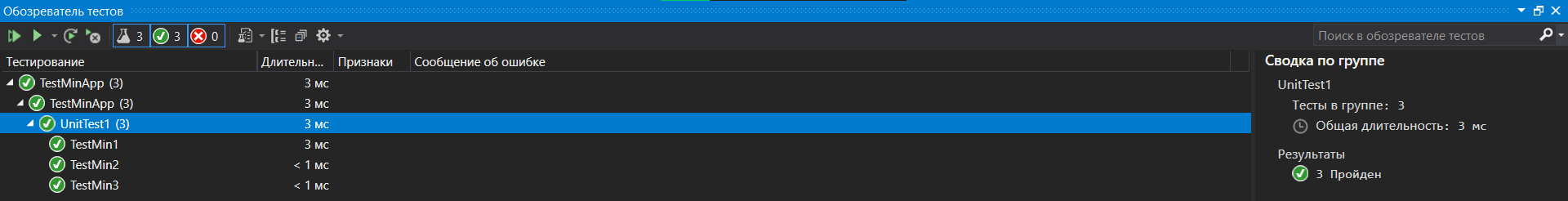
}

/\*В контекстном меню вызываем команду “Add reference” и выбираем проект MinApp. Тесты запускаются с помощью специального меню команд “Test”\*/

Структура проектов MinApp и TestMinApp



Скриншот “Обозревателя тестов”, показаны результаты всех тестов и их время работы:



Скриншот “Вспомогательной информации о тестах” в файле Program.cs тестируемой программы:



***Вывод***: в ходе выполнения данной лабораторной работой нами была протестирована программа на языке C# с помощью Unit-Test-ов. В качестве IDE была использована Microsoft Visual Studio. Был использован класс Assert, позволяющий проверить корректность работы функции.