

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И  
ОПТИКИ

Кафедра Программных Систем

**Практическая работа №1**  
**на тему:**

**«Создание и запуск модульных тестов »**

Выполнила:  
Загряжская Наталия Ильинична  
Группа:  
К4120

Санкт-Петербург  
2017

**Цель работы:** Использование тестов для разработки.

**Ход работы:**

### Упражнение 1.Использование тестов для разработки метода класса.

В этом упражнении мы разработали тестируемый метод с помощью среды тестирования модулей. Для этого были выполнены следующие действия:

1. Создан проект библиотеки классов Visual C# - MyMath.
2. Добавлен в решение тестовый проект.
3. Листинг кода теста:

```
[TestMethod]
public void BasicRooterTest()
{
    // Create an instance to test:
    Rooter rooter = new Rooter();
    // Define a test input and output value:
    double expectedResult = 2.0;
    double input = expectedResult * expectedResult;
    // Run the method under test:
    double actualResult = rooter.SquareRoot(input);
    // Verify the result:
    Assert.AreEqual(expectedResult, actualResult,
        delta: expectedResult / 100);
}
```

4. Сгенерируем новый тип для класса Rooter.

5. Выполним тесты.

6. Внесем следующие изменения:

```
public double SquareRoot(double input)
{
    return input / 2;
}
```

4. Проверим тесты (Рисунок 1).

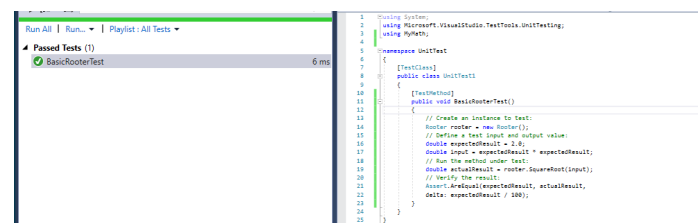


Рисунок 1 — Результат прохождения теста

## Упражнение 2. Добавление новых тестов

### 1. Листинг:

```
[TestMethod]
public void RooterValueRange()
{
    // Create an instance to test:
    Rooter rooter = new Rooter();
    // Try a range of values:
    for (double expectedResult = 1e-8;
        expectedResult < 1e+8;
        expectedResult = expectedResult * 3.2)
    {
        RooterOneValue(rooter, expectedResult);
    }
}
```

### 2. Результат изменения программы (Рисунок 2):

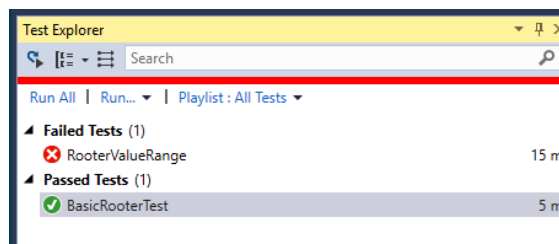


Рисунок 2 — Результат теста

### 3. Внесем еще изменения в SquareRoot. Листинг:

```
public double SquareRoot(double input)
{
    double result = input;
    double previousResult = -input;
    while (Math.Abs(previousResult - result) > result / 1000)
    {
        previousResult = result;
        result = result - (result * result - input) / (2 * result);
    }
    return result;
}
```

Результат прохождения модульных тестов изображен на Рисунке 3.

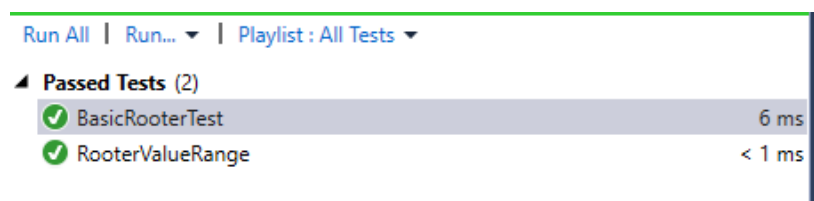


Рисунок 3 — Успешное прохождение тестов

#### 4. Добавим следующий метод:

```
[TestMethod]
public void RooterTestNegativeInputx()
{
    Rooter rooter = new Rooter();
    try
    {
        rooter.SquareRoot(-10);
    }
    catch (ArgumentOutOfRangeException e)
    {
        return;
    }
    Assert.Fail();
}
```

Результат изображен на Рисунке 4.

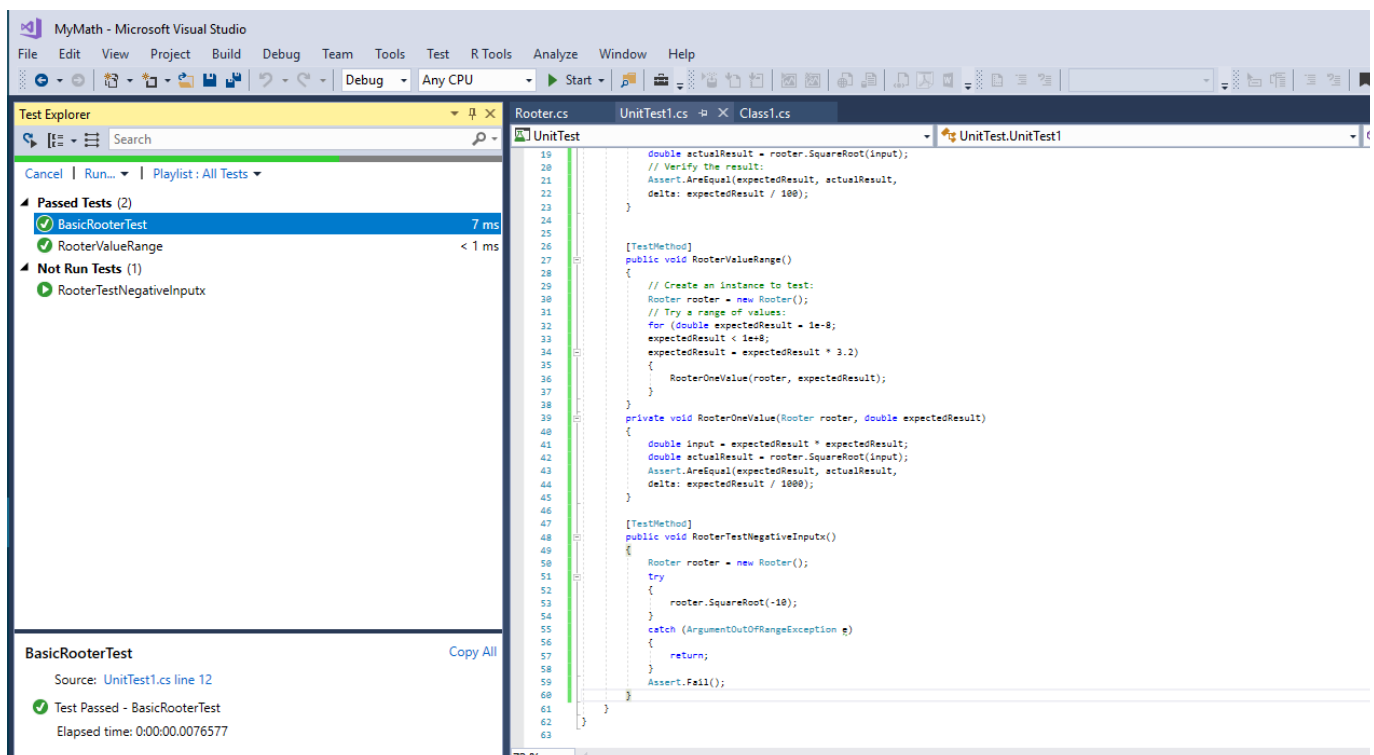


Рисунок 4 — Результат прохождения тестов после изменений в коде программы

Один из тестов заиклился. Внесем изменения в коде и добавим условие прохождения теста.

Результат прохождения проверки после Исправления метода в классе MyMath.Rooter (Рисунок 5).

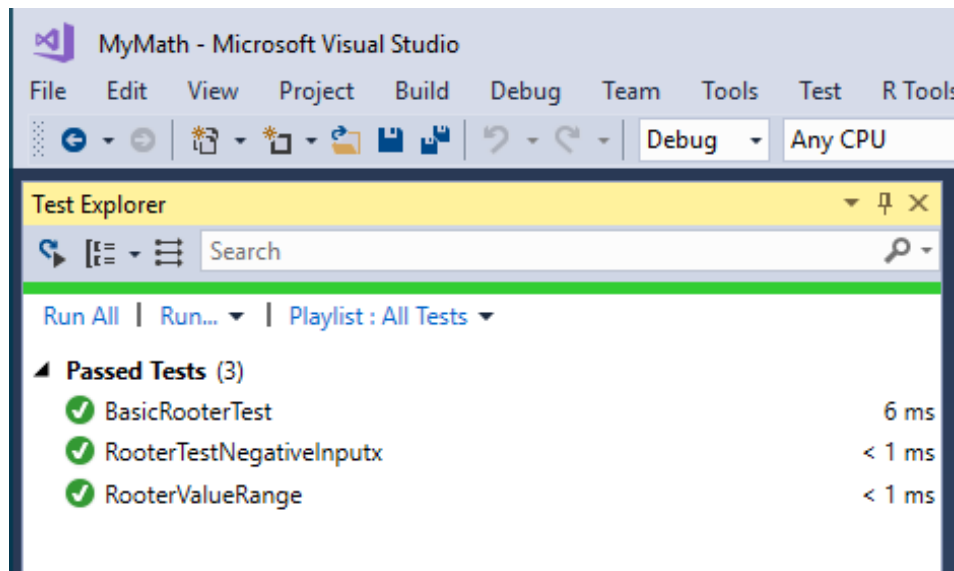


Рисунок 5 — Успешное прохождение всех тестов

Выводы:

В результате проделанной работы были изучены принципы работы с UnitTest в среде VisualStudio.