

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Рязанский государственный радиотехнический  
университет имени В.Ф. Уткина» Рязанский станкостроительный колледж

Отчёт о практической работе

Unit Test 4

МДК 05.03

Выполнил:

Студент группы ИСП-32

Тогузов М.А.

Проверил:

Родин Е.Н.

Рязань 2025

**Задание 1:**

Напиши тесты для метода IsPalindrome. Тесты должны проверять следующие случаи:

- Строка-палиндром.
- Строка не является палиндромом.
- Пустая строка. - Строка с пробелами.
- Числовая строка.

### **Задание 2:**

Напиши тесты для метода CountVowels. Тесты должны проверять следующие случаи:

- В строке есть гласные буквы.
- В строке нет гласных букв.
- Пустая строка.
- Сочетание больших и маленьких букв.
- Специальные символы

### **Задание 3:**

Напиши тесты для метода RemoveSpaces. Тесты должны проверять следующие случаи:

- Строка с пробелами.
- Пустая строка.
- Строка без пробелов.
- Пробелы в начале и конце строки.

### **Задание 4:**

Напиши тесты для метода CapitalizeEachWord. Тесты должны проверять следующие случаи:

- Нормальная строка.
- Пустая строка.
- Строка с несколькими словами.
- Строка с одним словом.
- Строка с числами.

## Шаг 1:

Написан код

```
using System;
using System.Diagnostics;
using System.Linq;
using Microsoft.VisualStudio.TestTools.UnitTesting;

public class Calc
{
    public decimal Value { get; private set; }

    public Calc(decimal initialValue)
    {
        Value = initialValue;
    }

    public decimal Sum(decimal number)
    {
        if (Value > decimal.MaxValue - number)
            throw new OverflowException("Result exceeds decimal max value");
        return Value += number;
    }

    public decimal Mul(decimal number)
    {
        return Value *= number;
    }

    public decimal Pow(int exponent)
    {
        return Value = (decimal)Math.Pow((double)Value, exponent);
    }

    public decimal Sqrt()
    {
        if (Value < 0)
            throw new ArgumentOutOfRangeException(nameof(Value), "Нельзя извлекать корень из отрицательного числа");
        return Value = (decimal)Math.Sqrt((double)Value);
    }
}

[TestClass]
public class CalcTests
{
    [TestMethod]
    [TestCategory("MathFunctions")]
    public void Calc_Pow_Test()
    {
        Calc calc = new Calc(2);
        decimal result = calc.Pow(3);
        Assert.AreEqual(8, result);
    }

    [TestMethod]
    [TestCategory("MathFunctions")]
    public void Calc_Sqrt_Test()
    {
        Calc calc = new Calc(16);
        decimal result = calc.Sqrt();
        Assert.AreEqual(4, result);
    }

    [TestMethod]
    [TestCategory("Exceptions")]
    public void Calc_Sqrt_NegativeValue_ExceptionTest()
    {
        Calc calc = new Calc(-10);
        Assert.ThrowsException<ArgumentOutOfRangeException>(() => calc.Sqrt());
    }
}
```

```

[TestMethod]
[TestCategory("State")]
# [Cause:0]
public void Calc_StatePreservation_Test()
{
    Calc calc = new Calc(0);
    calc.Sum(0);
    calc.Mul(2);
    Assert.AreEqual(10, calc.Value);
}

[TestMethod]
[TestCategory("BoundaryValues")]
public void Calc_BoundaryValues_Test()
{
    Calc calc = new Calc(decimal.MaxValue - 1);
    decimal result = calc.Sum(1);
    Assert.AreEqual(decimal.MaxValue, result);
}

[TestMethod]
[TestCategory("Performance")]
public void Calc_Performance_Test()
{
    Calc calc = new Calc(0);
    Stopwatch sw = new Stopwatch();
    sw.Start();
    for (int i = 0; i < 1000000; i++)
    {
        calc.Sum(1);
    }
    sw.Stop();
    Console.WriteLine($"Time taken: {sw.ElapsedMilliseconds} ms");
    Assert.IsTrue(sw.ElapsedMilliseconds < 5000);
}

//Task 13
public class StringUtils
{
    # [Cause:1] # 1/1 passing
    public static bool IsPalindrome(string input)
    {
        if (string.IsNullOrEmpty(input)) return false;
        string reversed = new string(input.Reverse().ToArray());
        return input.Equals(reversed, StringComparison.OrdinalIgnoreCase);
    }

    # [Cause:3] # 1/1 passing
    public static int CountVowels(string input)
    {
        if (input == null) throw new ArgumentNullException(nameof(input));
        char[] vowels = { 'a', 'e', 'i', 'o', 'u' };
        return input.ToLower().Count(c => vowels.Contains(c));
    }

    # [Cause:4] # 1/1 passing
    public static string RemoveSpaces(string input)
    {
        if (input == null) throw new ArgumentNullException(nameof(input));
        return input.Replace(" ", "");
    }

    # [Cause:1] # 1/1 passing
    public static string CapitalizeEachWord(string input)
    {
        if (input == null) throw new ArgumentNullException(nameof(input));
        return string.Join(" ", input.Split(' ').Select(word => word.Length > 0 ? char.ToUpper(word[0]) + word.Substring(1) : ""));
    }
}

[TestClass]
# [Cause:0]
public class StringUtilsTests
{
    [TestMethod]
    # [Cause:0]
    public void IsPalindrome_Test()
    {
        Assert.IsTrue(StringUtils.IsPalindrome("madam"));
        Assert.IsFalse(StringUtils.IsPalindrome("hello"));
        Assert.IsFalse(StringUtils.IsPalindrome(""));
        Assert.IsTrue(StringUtils.IsPalindrome("A man a plan a canal Panama".Replace(" ", "")));
    }

    [TestMethod]
    # [Cause:0]
    public void CountVowels_Test()
    {
        Assert.AreEqual(2, StringUtils.CountVowels("hello"));
        Assert.AreEqual(0, StringUtils.CountVowels("bcdfg"));
        Assert.AreEqual(5, StringUtils.CountVowels("aeiou"));
    }

    [TestMethod]
    # [Cause:0]
    public void RemoveSpaces_Test()
    {
        Assert.AreEqual("hello", StringUtils.RemoveSpaces("h e l l o "));
        Assert.AreEqual("", StringUtils.RemoveSpaces(" "));
        Assert.AreEqual("test", StringUtils.RemoveSpaces("test"));
    }

    [TestMethod]
    # [Cause:0]
    public void CapitalizeEachWord_Test()
    {
        Assert.AreEqual("Hello World", StringUtils.CapitalizeEachWord("hello world"));
        Assert.AreEqual("Test", StringUtils.CapitalizeEachWord("test"));
        Assert.AreEqual("123 456", StringUtils.CapitalizeEachWord("123 456"));
    }
}

```

Рисунок 1-3 - Код программы

## Шаг 2 :

### Тест всех методов

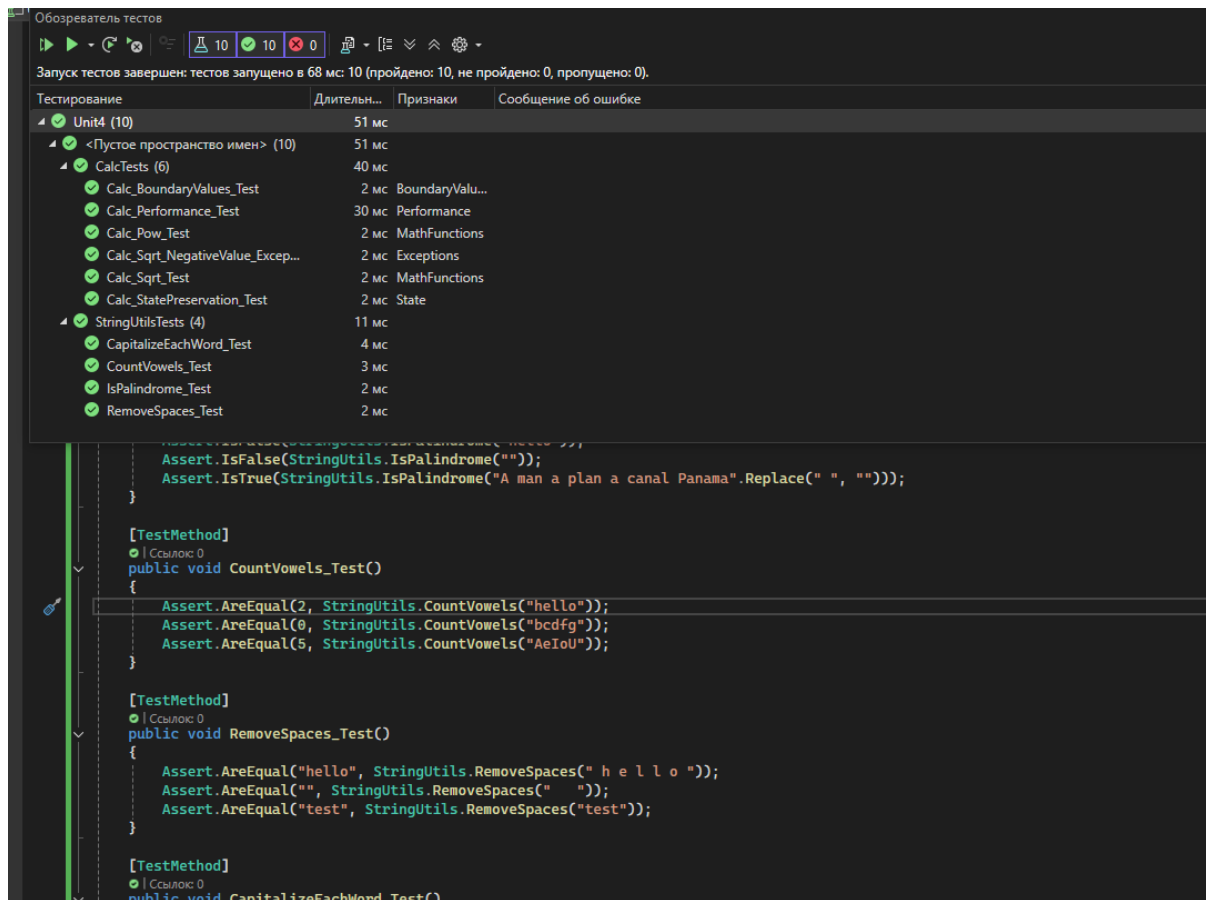


Рисунок 4 - Тест

**Вывод:** Мы получили новые навыки тестирования в Visual Studio