

Практика по модульному тестированию, снова стековый калькулятор

Юрий Литвинов
y.litvinov@spbu.ru

28.02.2025

TestCaseData (NUnit)

```
public class StackTest
```

```
{
```

```
    [TestCaseSource(nameof(Stacks))]
```

```
    public void StackShouldNotEmptyAfterPush(IStack stack)
```

```
{
```

```
        stack.Push(1);
```

```
        Assert.IsFalse(stack.IsEmpty());
```

```
}
```

```
private static IEnumerable<TestCaseData> Stacks
```

```
    => new TestCaseData[]
```

```
{
```

```
    new TestCaseData(new ArrayStack()),
```

```
    new TestCaseData(new ListStack()),
```

```
};
```

```
}
```

Ещё хороший приём

```
private static IEnumerable<TestCaseData> Stacks {  
    get {  
        var stacks = new List<IStack>()  
            { new ArrayStack(), new ListStack() };  
        var data = new List<int>() { 1, 2, 3 };  
        var result = new List<TestCaseData>();  
        foreach (var stack in stacks) {  
            foreach (var item in data) {  
                result.Add(new TestCaseData(item, stack));  
            }  
        }  
        return result;  
    }  
}
```

Более простые случаи

```
[TestCase(12, 3, 4)]  
[TestCase(12, 2, 6)]  
public void DivideTest(int n, int d, int q)  
    => Assert.AreEqual(q, n / d);
```

Или даже

```
[TestCase(12, 3, ExpectedResult=4)]  
[TestCase(12, 2, ExpectedResult=6)]  
public int DivideTest(int n, int d)  
    => return n / d;
```

NUnit «constraint model»

```
public void DivideTest(int n, int d, int q)  
    => Assert.That(n / d, Is.EqualTo(q)); Assert.AreEqual(q, );
```

Что ещё можно: [https:](https://docs.nunit.org/articles/nunit/writing-tests/constraints/Constraints.html)

[//docs.nunit.org/articles/nunit/writing-tests/constraints/Constraints.html](https://docs.nunit.org/articles/nunit/writing-tests/constraints/Constraints.html)

```
Assert.That(2.3, Is.GreaterThan(2.0).And.LessThan(3.0));  
Assert.That(array, Has.Member(3));  
Assert.That(array, Has.No.Member(5));  
Assert.That(collection, Is.Ordered.Ascending.By("A")  
    .Then.Descending.By("B"));  
Assert.That(phrase, Does.StartWith("Make"));
```

Задача

Реализовать модульные тесты к стековому калькулятору с прошлой практики.

- ▶ Проверять работоспособность на корректных и некорректных строках
- ▶ Проверять работу в случае использования разных реализаций стеков
- ▶ При этом не содержать дублирующегося кода