

## Лабораторная работа № 2

### «Консольный string-калькулятор»

<http://osherove.com/tdd-kata-1/>

*Описание задания:* необходимо создать простой string-калькулятор с методом **int CalcResult (string numbers)**. Метод может принимать 0, 1, 2 или несколько чисел в виде текстовой строки с разделителем между числами, и возвращает их сумму. Для пустой строки "" метод вернёт **0**, для строки "1" вернёт **1**, для строки "1,2" вернёт **3**, и т.д.

#### Прежде чем начать:

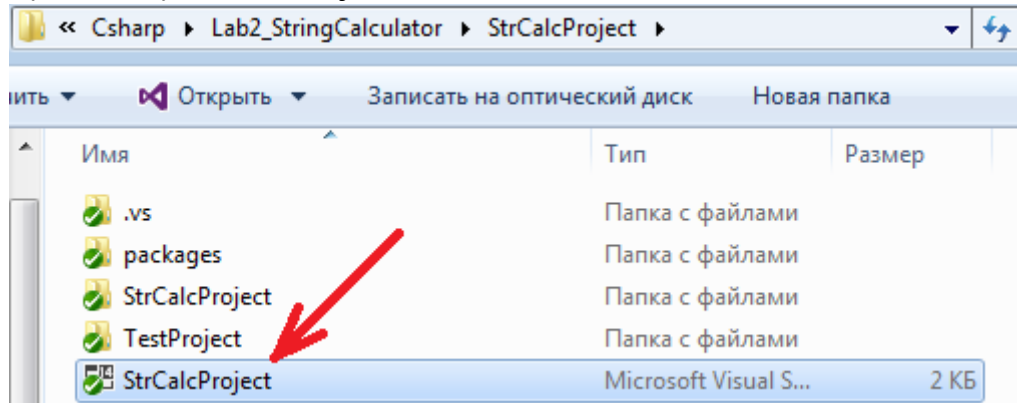
В лабораторной работе необходимые тесты, класс и необходимые методы с сигнатурами уже написаны. Студенту необходимо только заполнить (написать с нуля) тело каждого метода необходимым функционалом, и запускать выполнение тестов. Вносить изменения в тесты и сигнатуры методов запрещено. После того, как все тесты будут пройдены, необходимо вернуться к главному методу **Main**, и написать его функционал.

\n - знак переноса строки, \t - знак табуляции, и другие – это специальные комбинации символов в коде программы, выполняющие роль функциональных клавиш (Enter, Tab).

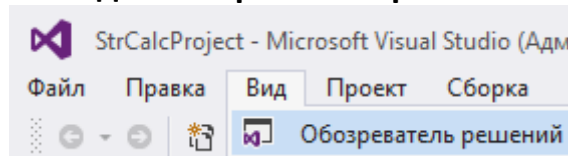
Старайтесь не читать вперед. Выполняйте одно задание (один пункт) за раз. Смысл заключается в том, чтобы научиться работать постепенно. Убедитесь, что вы проверяете только правильные входные данные. Нет необходимости проверять наличие недопустимых входных данных.

## Описание проекта String-калькулятор

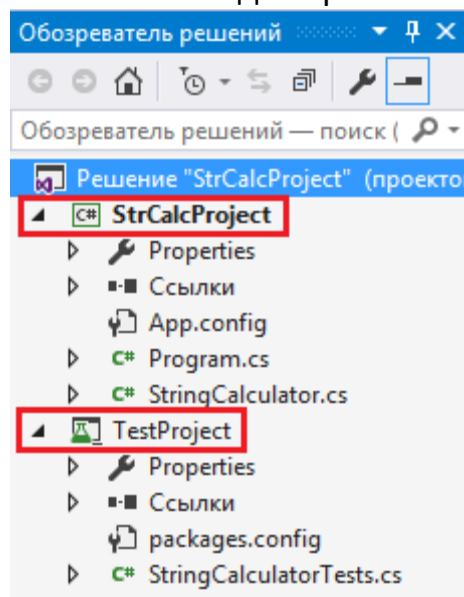
1. В Проводнике перейти в папку **d:\КомпОбрдан\Csharp\Lab2\_StringCalculator\StrCalcProject** и запустить файл решения (solution) **StrCalcProject.sln**



2. Если после запуска проекта, в Visual Studio нет окна **Обозревателя решений**, то в главном меню выбрать **Вид – Обозреватель решений**



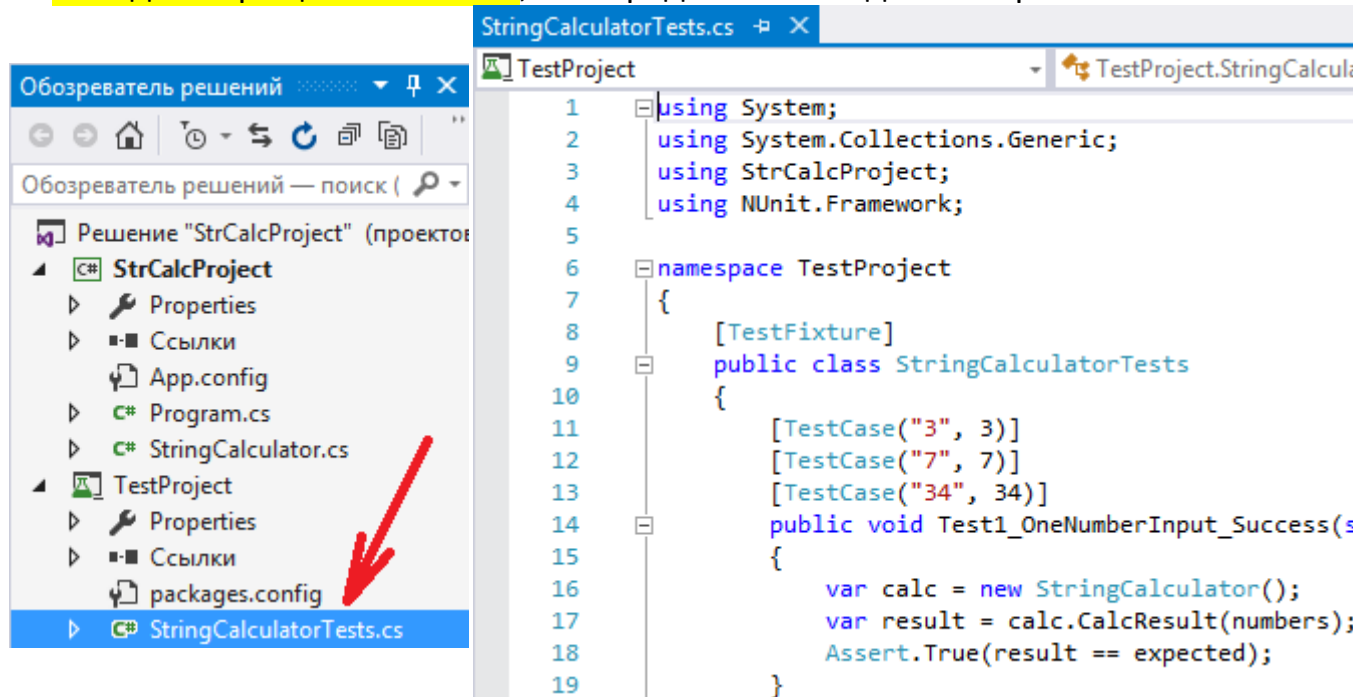
3. Решение **StrCalcProject** включает в себя два проекта: **StrCalcProject** и **TestProject**



4. Проект **StrCalcProject** является основным проектом, код для методов которого студент должен написать самостоятельно. **Сигнатуры методов (имя метода, количество параметров, типы параметров, порядок параметров, модификаторы параметров) уже заданы, вносить изменения в сигнатуры запрещено. Студент должен писать код только в теле методов.**

Проект **TestProject** является тестовым (вспомогательным) проектом, и предназначен для тестирования проекта **StrCalcProject**. Проект **TestProject** является полностью готовым, вносить изменения в его код запрещено.

5. В окне **Обозревателя решений** дважды кликнуть по классу **StringCalculatorTest** проекта **TestProject**. Откроется вкладка с кодом методов данного класса, **данные методы запрещено изменять**, они предназначены для тестирования.



Рассмотрим первый метод `Test1_OneNumberInput_Success` данного класса:

```
[TestCase("3", 3)]
[TestCase("7", 7)]
[TestCase("34", 34)]
public void Test1_OneNumberInput_Success(string numbers, int expected)
{
    var calc = new StringCalculator();
    var result = calc.CalcResult(numbers);
    Assert.True(result == expected);
}
```

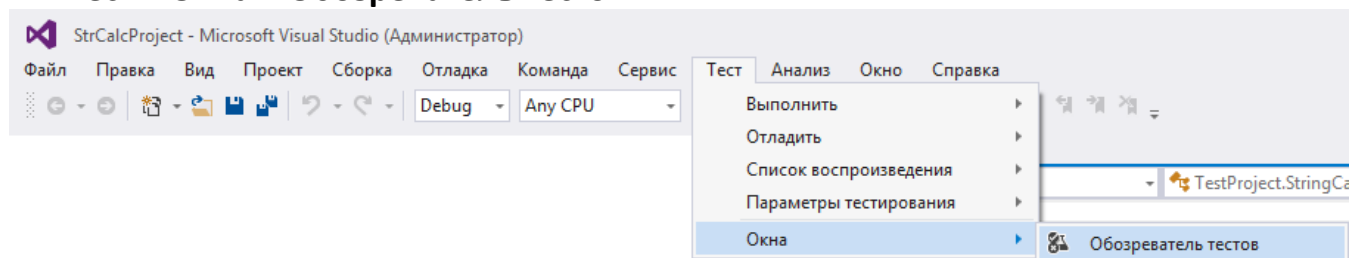
Для удобства восприятия имя метода состоит из трёх частей: порядковый номер теста `Test1`, примерное описание метода (что метод делает) `OneNumberInput` (т.е. одно число на вход), и что ожидаем от данного метода `Success`.

Данный метод ничего не возвращает, поэтому имеет тип `void`. На вход данный метод принимает два параметра: первый параметр `numbers` типа `string` (текстовая строка), и второй параметр `expected` типа `int` (неотрицательное целое число).

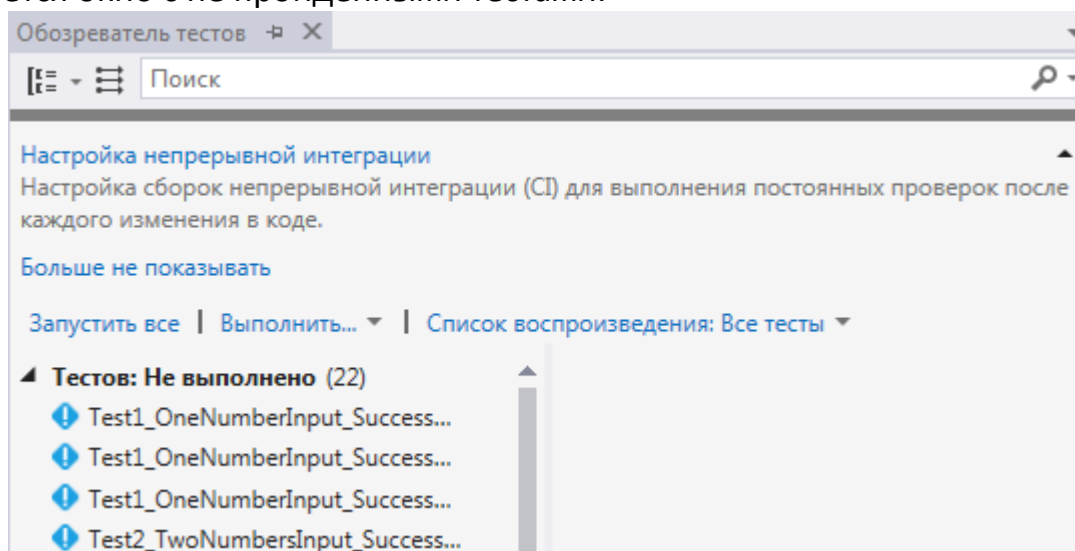
Двум параметрам метода соответствуют фактические данные, передаваемые в метод, и находящиеся в строках `TestCase` перед методом. Для первой строки `TestCase` параметру `numbers` соответствует значение `"3"`, а параметру `expected` соответствует значение `3`.

Таким образом в данном случае с помощью одного метода можно провести три теста, т.е. проверить 3 пары данных вход/результат.

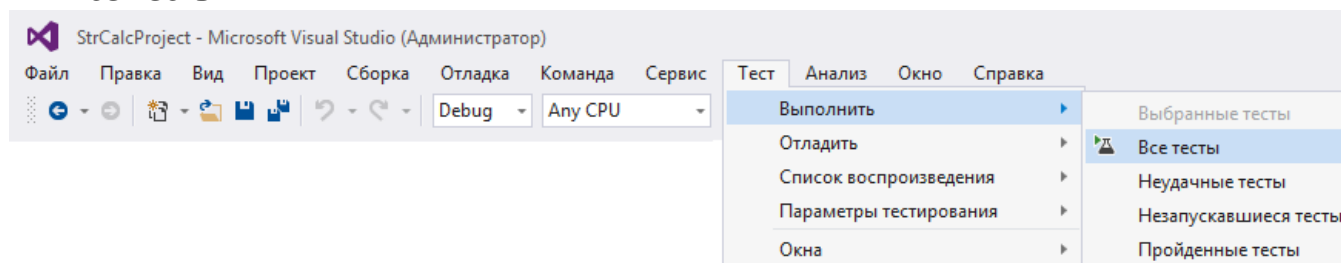
6. Чтобы открыть окно **Обозревателя тестов**, необходимо в главном меню выбрать **Тест – Окна – Обозреватель тестов**



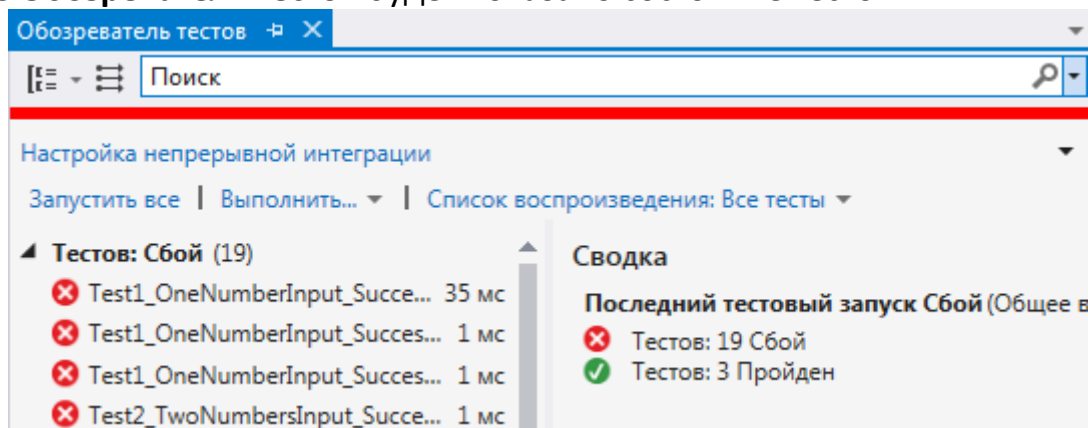
7. Откроется окно с не пройденными тестами.



8. Для запуска тестов необходимо в главном меню выбрать **Тест – Выполнить – Все тесты**

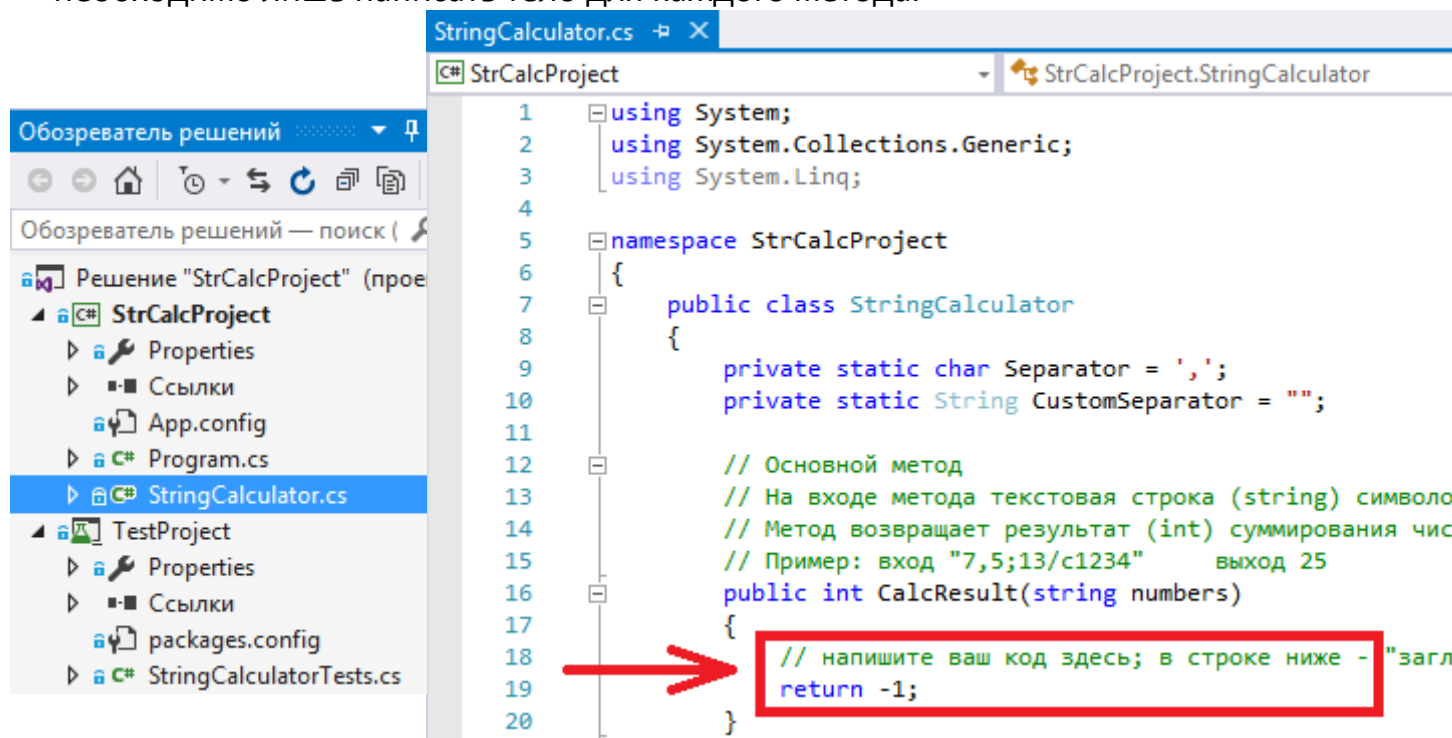


9. В окне **Обозревателя тестов** будет показано состояние тестов



Тесты необходимо запускать каждый раз, когда необходимо проверить, правильно ли написан функционал очередного метода.

10. Далее в окне **Обозревателя решений** дважды кликнуть по классу **StringCalculator** проекта **StrCalcProject**. Откроется вкладка с кодом данного класса. **Сигнатуры методов уже написаны, менять их запрещено**. Студенту необходимо лишь написать тело для каждого метода.



## Задания

1. Начните с самого простого теста пустой строки. Метод CalcResult принимает на входе пустую строку "", и возвращает на выходе число **0**.
2. Перейдите к тесту с одним числом на входе. Метод CalcResult принимает на входе строку с числом, например "1", и возвращает на выходе число **1**.
3. Затем перейдите к тесту с двумя числами на входе. Метод CalcResult принимает на входе строку с числами, например "1,2", и возвращает на выходе число **3**.
4. После этого перейдите к тесту с произвольным количеством чисел на входе. Метод CalcResult принимает на входе строку с числами, например "1,2,3,4", и возвращает на выходе число **10**.
5. Если во входной строке среди чисел есть хотя бы одно отрицательное, например "34,7,-2,7", то метод CalcResult вернёт на выходе **-1**.
6. Если во входной строке среди чисел есть хотя бы одно большее или равное 1000, например "4,7,2,1000,5", то метод CalcResult проигнорирует эти числа, и вернёт на выходе сумму оставшихся чисел, т.е. **18**.
7. Разрешить метод CalcResult для обработки новых разделителей между числами (вместо запятых). Метод принимает на входе, например, строку : "1\n2,3", и возвращает на выходе число **6**. Следующий разделитель НЕ подходит: "1,\n" (не нужно это доказывать - это просто уточнение)
8. Поддержка комплексных разделителей. Начало строки будет содержать отдельную строку, которая выглядит следующим образом  
"/ [разделитель]\n[числа]", например строка  
"/f  
3f1f7"  
на входе метода CalcResult, вернёт результат **11**. По умолчанию разделитель ';', а первая строка необязательна. Все существующие (предыдущие) сценарии по-прежнему должны поддерживаться.
9. Если во входной строке существуют чётные числа, то все они умножаются на 2. Метод CalcResult на входе получает, например, "1,2,3,4", и возвращает на выходе **16**. После выполнения данного теста, все предыдущие тесты должны перестать выполняться.