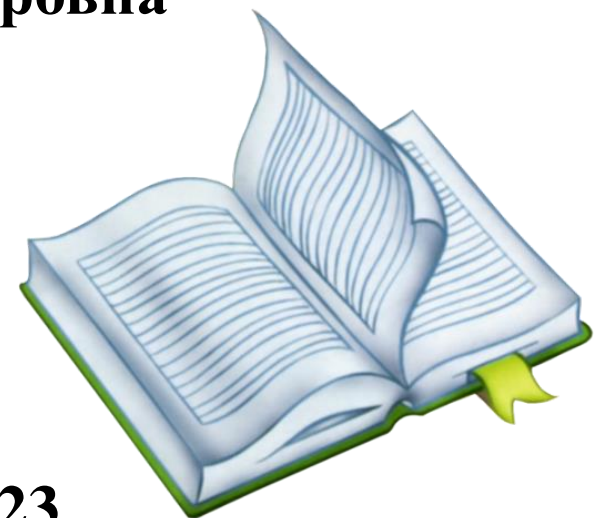


**ГПОУ «Ленинск-Кузнецкий политехнический техникум»**

# **Основы алгоритмизации и программирования**

## **Модульные тесты**

**Преподаватель Щеглова Алена Александровна  
Теоретическое занятие  
для студентов II курса**



**Ленинск-Кузнецкий, 2023**

# Актуализация

**Найти ошибки и определить результат и принцип работы программы**

```
class Foo {  
public int S1(int a, int b)  
    { return a + b; }  
static void Main(string[] args) {  
int a= Convert.ToInt32(Console.ReadLine());  
int b= Convert.ToInt32(Console.ReadLine());  
int res = new Program().S1(a, b);  
}
```

**Создать программу, которая включает в себя две функции, которые вычисляют площади и периметр квадрата. передает их значение в главную функцию программу и выводит на экран, значение стороны квадрата вводит пользователь и это значение является параметром функции**

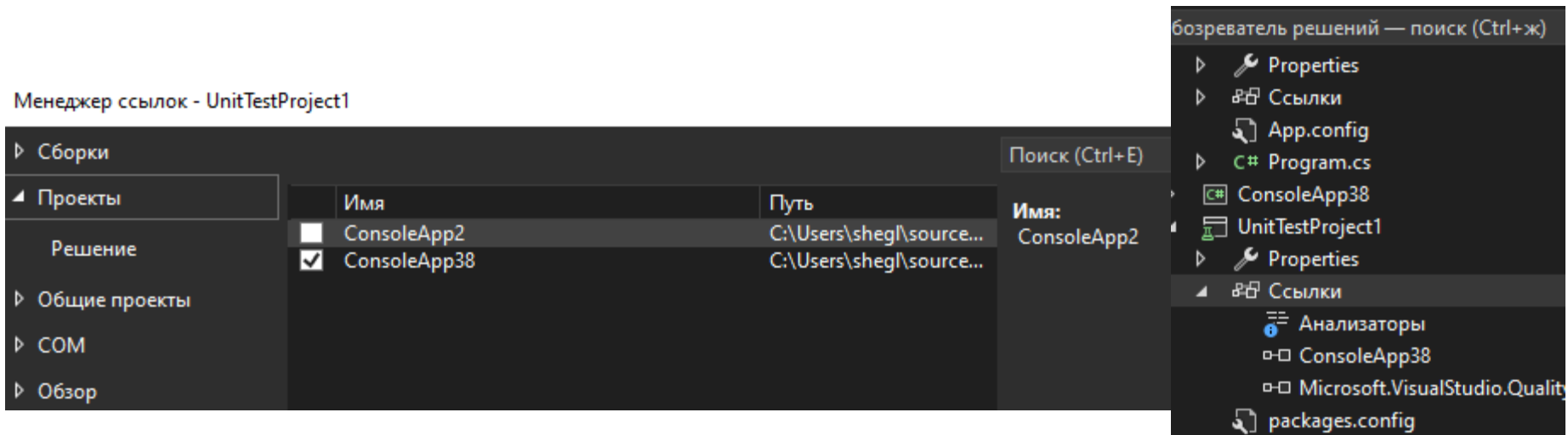
# Модульный тест

**Модульное тестирование - метод тестирования, с помощью которого тестируются отдельные модули, чтобы определить, есть ли какие-либо проблемы самим разработчиком.**

**Цель: изолировать каждую единицу системы для выявления, анализа и исправления дефектов.**

# Создание модульного теста

1. **Файл/ Добавить/ Создать проект/ Проект модульного теста (.NET Framework)**
2. **В Обозревателе решений выбрать Ссылки модульного теста, открыть контекстное меню и выбрать Добавить ссылку:**
  - **в разделе Проект выбрать название своего консольного приложения**
  - **в разделе Расширения выбрать Microsoft.VisualStudio.Quality Tools.UnitTestFramework, версия 10.0.0.0**



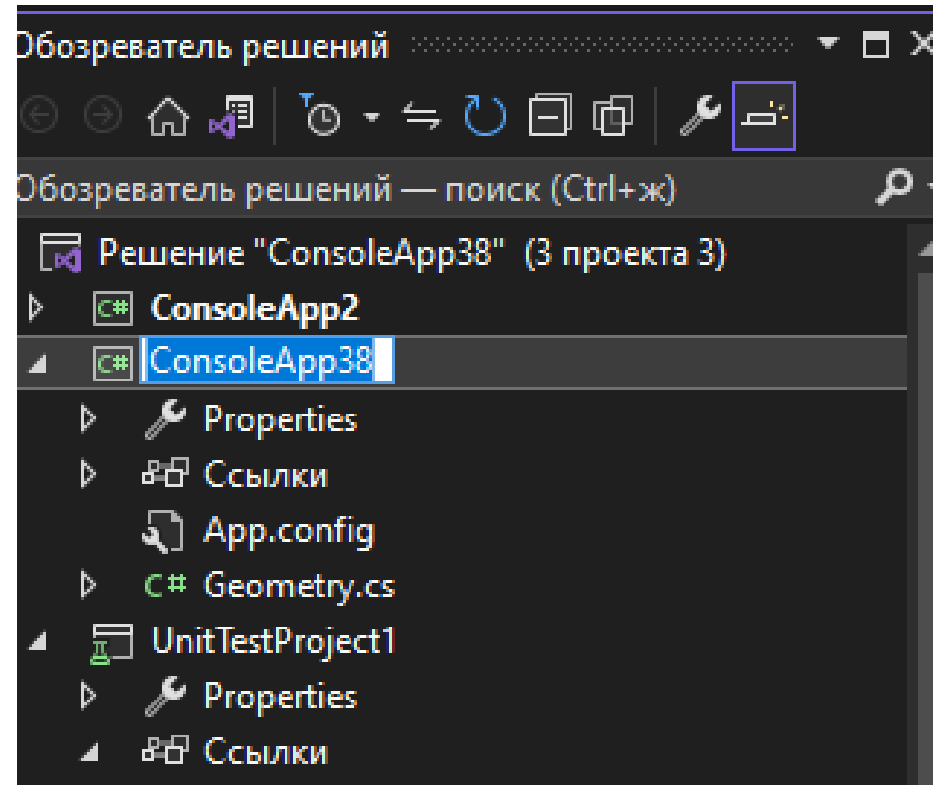
# Создание модульного теста

В коде теста необходимо подключить с помощью директивы `using` имя тестируемого проекта:

`using имя проекта;`

Например:

`using ConsoleApp38;`



# Создание модульного теста

- TestClassAttribute

Используется для идентификации тестовых классов

- TestMethodAttribute

Используется для идентификации тестовых методов

```
[TestClass]
public class UnitTest1
{
    [TestMethod]
    public void TestMethod1()
    {
    }
}
```

# Создание модульного теста

[TestClass]

```
public class UnitTest1{
```

[TestMethod]

```
public void имя_метода{
```

```
}
```

**Если тестовый класс содержит несколько методов:**

[TestClass]

```
public class UnitTest1{
```

[TestMethod]

```
public void имя_метода{
```

```
    [TestMethod]
```

```
public void имя_метода1{
```

[TestMethod]

```
public void имя_метода2{
```

```
}
```

# Создание модульного теста

**Например: тест для функции**

```
class Foo {  
public int S1(int a, int b)  
{ return a + b; }
```

**Тестирующий метод содержит три необходимых компонента:**

**1. Исходные данные: входные значения и ожидаемый результат:**

```
int имя_значений для_параметра = 3; // исходные данные  
int имя_значений для_параметра = 2; // исходные данные  
int имя_переменной_ожидаемый_результат = 3;
```



# Создание модульного теста

**Например: тест для функции**

```
class Foo { public int S1(int a, int b) { return a + b; } }
```

**Тестирующий метод содержит три необходимых**

**компонента:**

**2. Код, вычисляющий значение с помощью тестируемого метода (вызов функции):**

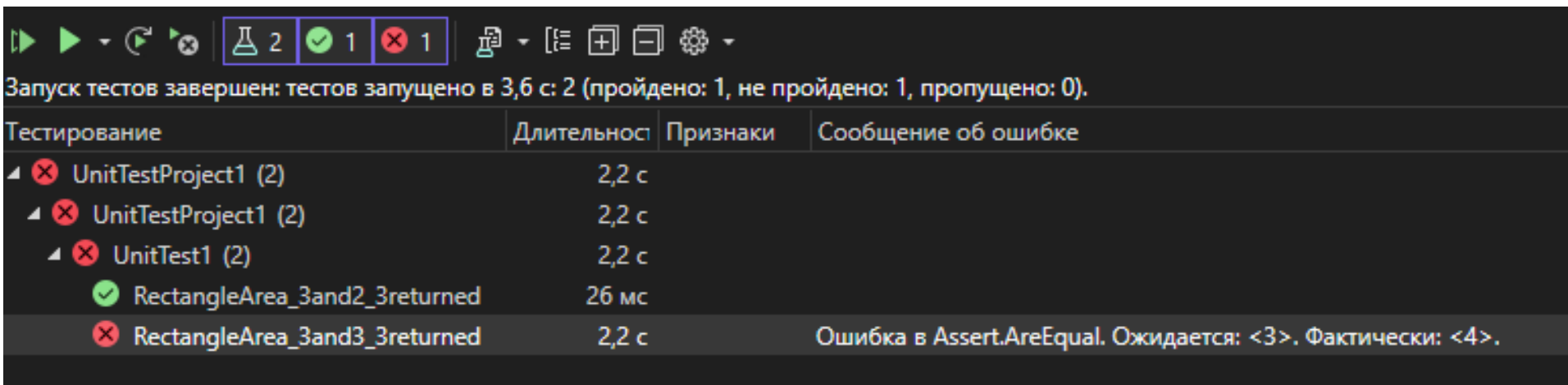
```
Имя_класса переменная1 = new Имя_класса();  
int фактический_результат= переменная1.Имя_функции  
(параметры); // получение значения с помощью  
тестируемого метода
```

**3. Код, сравнивающий ожидаемый результат с полученным.**

```
Assert.AreEqual(expected, actual); // сравнение ожидаемого  
результата с полученным
```

# Запуск модульных тестов на выполнение

1. Сборка/Собрать решение
2. Тест/Запуск тестов



Запуск тестов завершен: тестов запущено в 3,6 с: 2 (пройдено: 1, не пройдено: 1, пропущено: 0).

Тестирование	Длительность	Признаки	Сообщение об ошибке
✘ UnitTestProject1 (2)	2,2 с		
✘ UnitTestProject1 (2)	2,2 с		
✘ UnitTest1 (2)	2,2 с		
✔ RectangleArea_3and2_3returned	26 мс		
✘ RectangleArea_3and3_3returned	2,2 с		Ошибка в Assert.AreEqual. Ожидается: <3>. Фактически: <4>.