**Лабораторная работа 3. TDD – Test Driven Development**

Методология разработки через тестирование. Три закона TDD:

1. Новый рабочий код пишется только после того, как будет написан модульный тест, который не проходит.
2. Вы пишете ровно такой объем кода модульного теста, какой необходим для того, чтобы этот тест не проходил (если код теста не компилируется, считается, что он не проходит).
3. Вы пишете ровно такой объем рабочего кода, какой необходим для прохождения модульного теста, который в данный момент не проходит.

**Создание UNIT-тестов в Visual Studio**

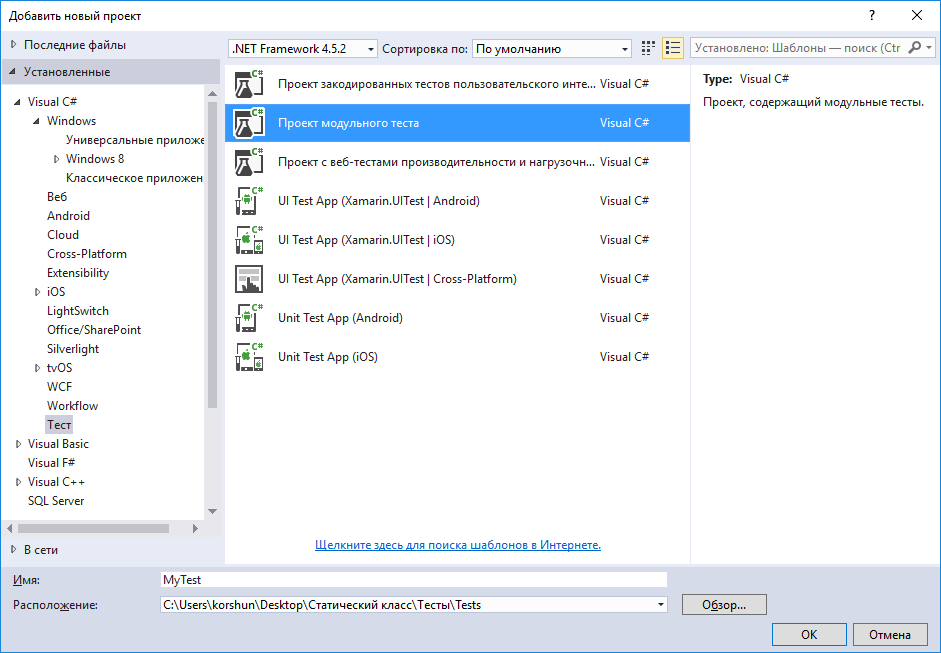
Модульное тестирование или unit-тестирование – это тестирование отдельных частей программы.

Задача тестирования состоит в том, чтобы написать дополнительный класс с методами, который будет тестировать основные классы программы. Класс для тестирования – это обычный C#-класс, который компилируется в dll-файл, но этот C#-класс должны быть размечен с помощью специальных атрибутов.

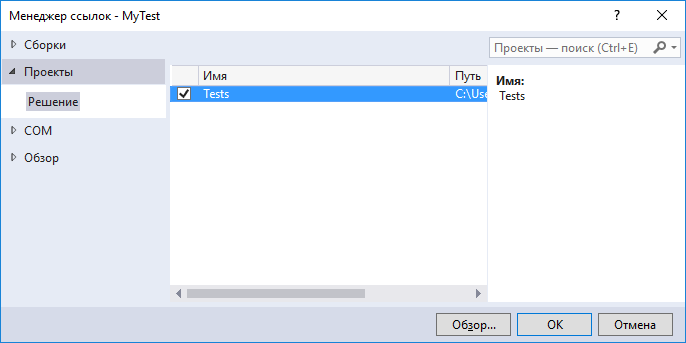
Создание теста.

1 способ. В файле класса вызвать контекстное меню и выбрать пункт Создание модульных тестов. Затем нажмите OK. Файл с тестовым классом будет создан.

2 способ. Добавьте новый проект модульного теста с именем MyTest в данное решение. Вместо префикса My запишите имя класса, к которому создаете тесты.



В тестовом проекте необходимо добавить ссылку на основной проект: Свойства проекта -> Добавить -> Ссылка.



В созданном тестовом проекте создан тестовый класс

[TestClass]

public class MyTests

{

[TestMethod]

public void TestMethod()

{

…

}

}

Обязательно объявите класс с модификатором public. В разделе ссылок тестового проекта добавьте ссылку на проект с проверяемым классом. В тестовый класс добавьте ссылку на основной проект, используя using.

Для создания тестов используется статический класс Assert. В нем определен набор методов, которые можно использовать в тестах. Класс Assert находится в пространстве имен Microsoft.VisualStudio.TestTools.UnitTesting.

Если метод кидает исключение, то для этого метода должно ты написано несколько тестов, проверяющих благополучный исход и исключительную ситуацию. В случае проверки исключения необходим атрибут:

[ExpectedException(typeof(ТипПроверяемогоИсключения))]

Для проверки выполнения тестов откройте панель Обозреватель тестов (меню Тест –> Окна –> Обозреватель тестов).

**Задание**. Используя три закона TDD описать класс. Тесты должны опережать описание класса (не менее 20 тестовых методов). Тесты должны покрывать весь код класса.

**Задача 1**.

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| Составить описание класса для представления дробей. Обеспечить выполнение операций сложения, вычитания умножения и деления дробей, возведение в степень n. Предусмотреть сокращение дробей при выполнении операций, вывода дроби на экран. | Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания умножения и деления комплексных чисел, возведение в степень *n* и извлечение корня *n* степени, вывода чисел на экран. |

**Задача 2**.

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения, вычитания и умножения многочленов с получением нового объекта-многочлена. Перегрузите операции индексации и операторы сравнения двух многочленов. Предусмотреть вывод на экран описания многочлена. | Составить описание класса для определения одномерных массивов целых чисел. Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта, возможность выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, вывода на экран элемента массива по заданному индексу, поиска заданного элемента. Предусмотреть вывод массива на экран. |

**Задача 3**.

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| Составить описание **статического** класса для чисел в 16-ой системе счисления (с.с). Предусмотреть конструктор для случая, когда вводится число в 10-ой с.с и в 16-ой с.с., а также проверку корректности записи числа. Сделать методы сложения и вычитания чисел в 16-ой с.с., а также побитовых операций AND и OR. | Составить описание **статического** класса для чисел в 8-ой системе счисления (с.с). Предусмотреть конструктор для случая, когда вводится число в 10-ой с.с и в 8-ой с.с., а также проверку корректности записи числа. Сделать методы сложения и вычитания чисел в 8-ой с.с., а также побитовых операций AND и OR. |