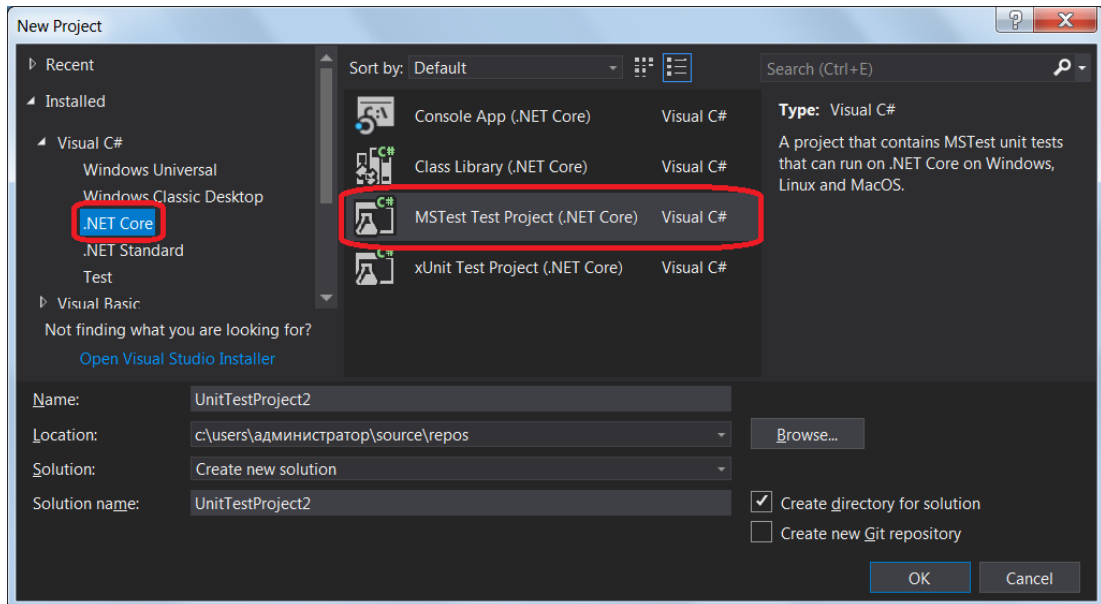
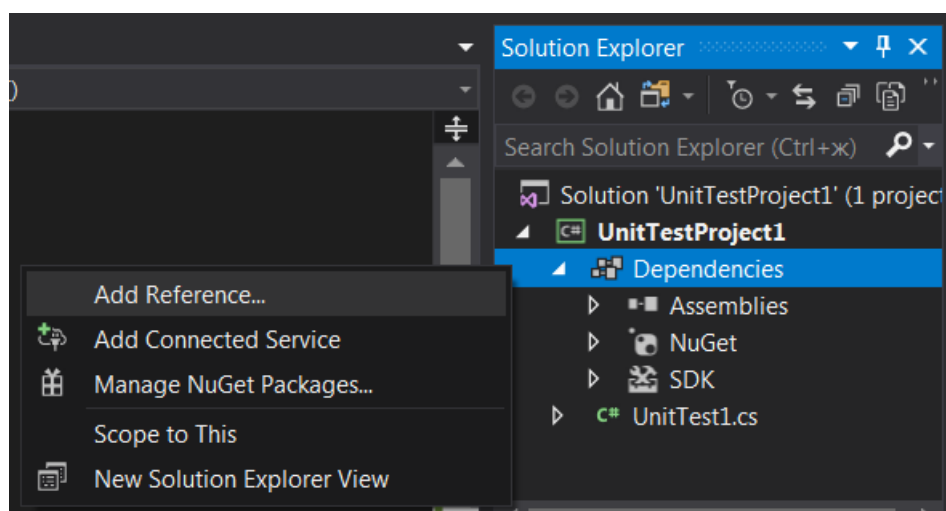


## Unit-тестирование dll-библиотеки

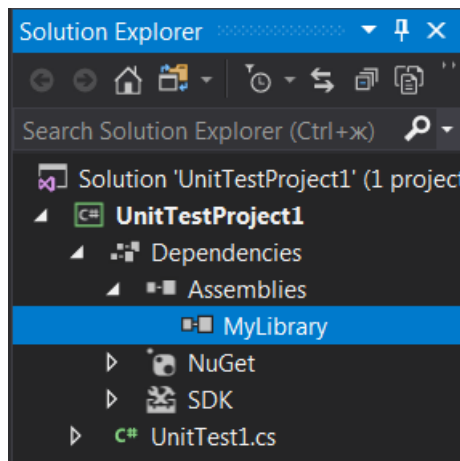
Unit-тесты представляют из себя специализированные методы, предназначенные для автоматического тестирования программного кода. В обычных приложениях юнит-тесты, как правило, включаются непосредственно в состав проекта. В случае с dll-библиотекой необходимо будет создать новый проект. Тип проекта – **MSTest Test Project**:



Далее необходимо подключить тестируемую dll-библиотеку. Выполняется это почти так же, как описано в файле «Создание dll.pdf» - правый клик на разделе **Dependencies**, выбор пункта **Add Reference** и указание dll-файла:



Подключенная библиотека появится в подразделе **Assemblies**:



Каждый тестовый метод, создаваемый внутри тестового класса, представляет собой отдельный тестовый случай. При этом простые тестовые методы строятся по одинаковой структуре:

- подготовка тестовых данных;
- выполнение некого действия над тестовыми данными;
- сравнение результата данного действия с ожидаемым результатом.

Для примера разберем данный тест **CheckLongVin**:

```
namespace UnitTestProject1
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void CheckLongVin()
        {
            string testVin = "QWE356ABNMU7908ASDJUIQ";
            Assert.AreEqual(false, MyLibrary.VinCheck.CheckVIN(testVin));
        }
    }
}
```

Название тестового метода должно отражать суть тестового случая. Здесь мы передаем в метод **CheckVIN** строку, которая *заведомо длиннее* стандартного VIN. Таким образом, строчка

```
string testVin = "QWE356ABNMU7908ASDJUIQ";
```

является входными тестовыми данными. Этот тест совсем простой, поэтому здесь выполнение действия над данными совмещено с проверкой результата в одной инструкции:

```
Assert.AreEqual(false, MyLibrary.VinCheck.CheckVIN(testVin));
```

В метод `Assert.AreEqual` передаются два параметра: первый является ожидаемым значением, второй – результатом тестируемого действия. Так как

тестовая строка изначально длиннее стандартного VIN, то ожидается, что метод вернет значение **false**.

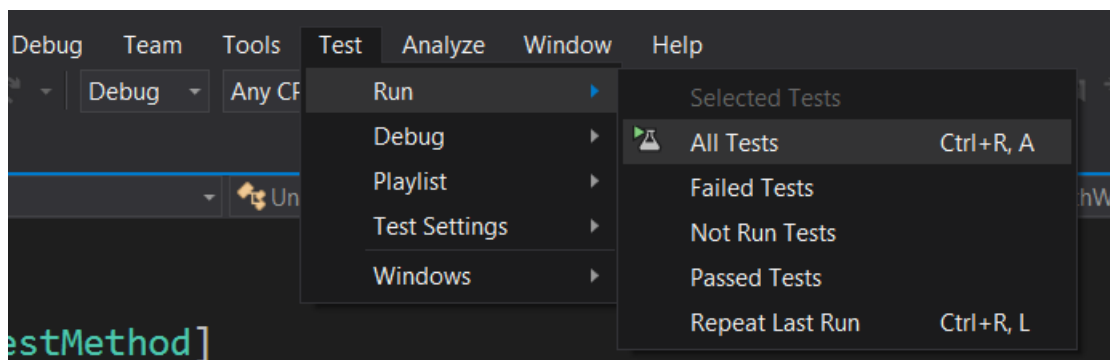
Следующий тест проверяет работу метода с заведомо короткой строкой:

```
[TestMethod]
public void CheckShortVin() {
    string testVin = "QWE356ABN";
    Assert.AreEqual(false, MyLibrary.VinCheck.CheckVIN(testVin));
}
```

В третьем методе длина тестовой строки равна 17 символам, но последние 6 символов не являются цифрами. Ожидается, что тестируемый метод **CheckVIN** также вернет значение **false**:

```
[TestMethod]
public void CheckVinWithWrongSerialNumber() {
    string testVin = "QWE356ABNYU138E95";
    Assert.AreEqual(false, MyLibrary.VinCheck.CheckVIN(testVin));
}
```

Написанные тесты запускаются командой меню **Test – Run – All Tests**:



Результаты прохождения тестов отображаются в отдельной вкладке:

