Атрибуты NUnit:

* [**SetUp**] - этот атрибут мы можем применять к методу так же как и атрибут [Test], он вызывает метод каждый раз перед запуском тестов в вашем классе.
* [**TearDown**] - этот атрибут будет вызывать метод один раз после выполнения каждого теста.
* [**TextFixtureSetUp**], [**TextFixtureTearDown**] - позволяют установить состояние один раз перед запуском всех тестов или после запуска тестов.
* [**ExpectedException**] - сообщение об исключении предоставляется в качестве параметра атрибута.
* [**Ignore**] - применяется к методам, которые не будут выполнятся при запуске всех юнит-тестов.

Нет смысла получать от метода логическое значение, потому что вызов метода должен вызвать исключение.

Вы можете настроить тесты для работы под конкретные категории испытаний, таких как медленных и быстрых тестов. Сделать это вы можете с помощью атрибута [Category].

Вы можете создать несколько категорий тестов в коде, а затем выбрать конкретную категорию для запуска в среде NUnit.

**Проверка юнит-тестов**

Класс Assert проверяет условия, использующие утверждения "истина/ложь", в процессе модульных тестов

Методы-утверждения класса Assert:

* Равенства–AreEqual();
* Идентичности –AreSame(), AreNotSame();
* Сравнения –Greater();
* Типа–IsInstanceOfType();
* Условия–IsTrue(), IsFalse(), IsNotNull();

**Способы проверки правильности юнит-теста**

* AreEqual()-сравнение двух аргументов.
* AreSame() -проверяет, ссылаются ли переменные на одну и ту же область памяти.
* Contains() -используется для проверки объектов, содержащихся в коллекции или массиве.
* Greater(a, b) -проверяет, является ли один объект больше, чем другой (a > b).
* Less(a, b) -Проверяет, является ли один объект меньше, чем другой (a < b).
* InInstanceOf(typeof(“hello”, string)) -метод для проверки типов объектов.

**Проверка конкретных условий**

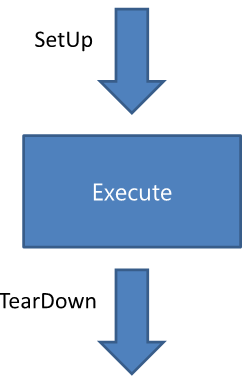
* IsTrue(2 + 2 == 4) –проверка истинности логической конструкции.
* IsFalse(2 + 2 == 5) –проверка ложи логической конструкции.
* IsNan(double a) –проверка является ли объект не числом.
* IsEmpty(“”) –хранит ли объект пустое значение.

**Создание юнит-теста**

* [TestFixture]–атрибут помечает класс, который будет содержать юнит-тесты.
* [Test]–метод декорированный данным атрибутом считается юнит-тестом и будет распознаваться средой тестирования NUnit.

**Настройка и очистка состояния**

* [SetUp]–атрибутом помечается метод, который вызывается перед запуском каждого юнит-теста.
* [TearDown]–атрибутом помечается метод, который вызывается после завершения каждого юнит-теста.
* [TextFixtureSetUp]–атрибут определяет метод, который выполнится один раз перед запуском всех юнит-тестов в классе с тестами.
* [TextFixtureTearDown]–атрибут определяет метод, который выполнится один раз по завершению всех юнит-тестов в классе с тестами.



**Проверка ожидаемых исключений**

[ExpectedException] - атрибут указывает, что юнит-тест должен выбросить исключение определенного типа.

В этом юнит-тесте нет смысла использовать методы класса Assert, так как проверка юнит-теста будет заключатся в проверке наличия исключения при выполнении.

**Игнорирование тестов**

[Ignore] - применяется к методам, которые не будут выполнятся при запуске всех юнит-тестов.

**Установка категорий тестов**

С помощью атрибута [Category] можно распределить тесты по различным категориям для упрощения анализа результатов.

Вы можете создать несколько категорий тестов в коде, а затем выбрать конкретную категорию для запуска в среде NUnit.