

Доповідь

На тему: «Фреймворк xUnit.net»

Підготував студент 3-го курсу, групи ТТП-31

Микитюк Максим

1. Огляд

xUnit.net це безкоштовний інструмент з відкритим кодом, призначений для платформи .NET та створений авторами оригінального NUnit. Він має ліцензію від Apache, працює з Xamarin, ReSharper, CodeRush, та TestDriven.NET

На даний момент, це найновіший із поточних фреймворків тестування для платформи .NET. Основною причиною його створення було бажання авторів створити нову, покращену платформу, замість нарощування функціоналу на існуючий NUnit.

2. Приклади використання

```
using System;
using Xunit;
using Validators.Password;

namespace PasswordValidatorTests
{
    public class ValidityTest
    {
        [Theory]
        [InlineData("ThisIsapassword!", true)]
        [InlineData("thisIsapassword123!", true)]
        [InlineData("Abc$123456", true)]
        [InlineData("Th1s!", false)]
        [InlineData("thisIsAPassword", false)]
        [InlineData("thisisapassword#", false)]
        [InlineData("THISISAPASSWORD123!", false)]
        [InlineData("", false)]
        public void ValidatePassword(string password, bool expectedResult)
        {
            //Arrange
            var passwordValidator = new PasswordValidator();

            //Act
            bool isValid = passwordValidator.IsValid(password);

            //Assert
            Assert.Equal(expectedResult, isValid);
        }
    }
}
```

```

using Hw1.Exercisel;
using Hw1.Tests.Tools;
using Xunit;

#pragma warning disable CA1707 // Test name can contains underscores
namespace Hw1.Tests.Exercisel
{
    Ссылка: 0
    public class PrimeNumbersTests
    {
        [Theory]
        [InlineData(new string[] { "5", "7" }, "5;7;", 0)]
        [InlineData(new string[] { "0", "10" }, "2;3;5;7;", 0)]
        [InlineData(new string[] { "-10", "10" }, "2;3;5;7;", 0)]
        [InlineData(new string[] { "10", "-10" }, "2;3;5;7;", 0)]
        [InlineData(new string[] { "104", "112" }, "107;109;", 0)]
        [InlineData(new string[] { "112", "104" }, "107;109;", 0)]
        Ссылка: 0
        public void Run_WithValidRange_ReturnsPrimeNumbers(string[] args, string primeNumbers, int successCode)
        {
            using var output = ConsoleOutputInterceptor.InterceptOutput();
            var app = new PrimeNumbersApplication();

            var returnCode = app.Run(args);
            Assert.Equal(successCode, returnCode);

            var outputStr = output.ToString().NormalizeOutput(trimNewLineEnding: true);
            Assert.Equal(primeNumbers, outputStr);
        }

        [Theory]
        [InlineData(null, -1)]
        [InlineData(new string[] { "" }, -1)]
        [InlineData(new string[] { "1,2", "" }, -1)]
        [InlineData(new string[] { "1.2", "3,4" }, -1)]
        [InlineData(new string[] { "1", "two" }, -1)]
        [InlineData(new string[] { "one", "two" }, -1)]
        [InlineData(new string[] { "one", "two", "1", "2" }, -1)]
        Ссылка: 0
        public void Run_WithInvalidArgs_ReturnsError(string[] args, int errorCode)
        {
            var app = new PrimeNumbersApplication();
            var returnCode = app.Run(args);
            Assert.Equal(errorCode, returnCode);
        }
    }
}

```

3. Архітектура та структура фреймворку

Структура xUnit.net в цілому схожа на типову структуру сімечства xUnit (зокрема й JUnit).

Основні компоненти:

- Модуль, який виконує тестування – програма, яка безпосередньо виконує тести і відображає дані про їх виконання
- Сценарії – варіанти тестування
- Конфігурації тестування – набір умов та станів прописаний для запуску тестів
- Набори тестів – групи тестів, які мають спільну конфігурацію
- Форматування результатів тестування

- Твердження – функція, яка перевіряє стан та поведінку об'єкта, якого тестують

4. Альтернативи

Разом з xUnit.net основними інструментами тестування для платформи .NET є MSTest та NUnit.

Платформа xUnit відрізняється від NUnit зокрема відсутністю тегів Test та TestFixture, а такі атрибути як SetUp і TearDown більше не підтримуються. Причина цих змін в дублюванні коду, якого автори фреймворку вирішили уникнути через виклик конструктора замість SetUp та використання інтерфейсу IDisposable замість TearDown.

Це робить xUnit на порядок зручнішим інструментом при роботі з Selenium. Він значно потужніший та легший в розширюванні.

І якщо порівнювати з альтернативними фреймворками, в розширюваності і заключається основна перевага xUnit.net: використання атрибуту Fact замість Test дозволило імплементувати в нього непараметризовані тести, в той час як Theory реалізував параметризовані.

В NUnit та MSTest атрибут TestClass містить в собі клас із тестами, що доволі незручно, якщо порівнювати із xUnit. Він, на противагу старим фреймворкам, може самостійно визначити де знаходяться тестові методи, не зважаючи на місцезнаходження тестів

5. Переваги та недоліки

Окрім всього вищеназваного, із переваг важливо зазначити сучасність і постійну підтримку, а з недоліків відчутно меншу кількість матеріалів по ньому (якщо порівнювати з NUnit), та прив'язаність до конкретних платформ.

Таблиця сумісності з різними платформами:

Test Runner Compatibility

	xUnit.net	xunit.console	MSBuild	TestDriven.NET	Visual Studio ²	Devices
Desktop Class Library	1.9.2+	✓ ^{3a}	✓ ^{3a}	✓ ^{3a}	✓ ^{3b} (2012+)	
.NET Core (Windows, Linux, OS X)	2.2+				✓ ^{3b} (2017+)	
Universal Windows Platform	2.1+				✓ ^{3b} (2015+)	✓ ^{3c}
Xamarin MonoAndroid ¹	2.0+					✓ ^{3c}
Xamarin iOS Unified ¹	2.0+					✓ ^{3c}
↓ <i>The project types below have been officially deprecated</i> ↓						
Universal Application (Windows 8.1, Windows Phone 8.1)	2.0 - 2.1				✓ ^{3b} (2013+)	✓ ^{3c}
Windows Phone 8 (Silverlight)	2.0 - 2.1					✓ ^{3c}
Xamarin MonoTouch (iOS Classic) ¹	2.0 - 2.1					✓ ^{3c}

6. Підсумок

xUnit.net – найсучасніший фреймворк тестування для .NET, і в порівнянні з NUnit та MSTest він може бути більш оптимальним варіантом, особливо в зручності та ефективності кодування.