**Приступая к работе с модульным тестированием**

Используйте Visual Studio, чтобы определить и запустить модульные тесты для обеспечения работоспособности кода, нужного объема протестированного кода, а также для обнаружения ошибок и сбоев, прежде чем с ними столкнутся клиенты. Выполняйте модульные тесты регулярно, чтобы обеспечить правильную работу вашего кода.

В этой статье код использует C# и C++, иллюстрации представлены на C#, однако концепции и функции применимы к языкам .NET, C++, Python, JavaScript и TypeScript.

**Создание модульных тестов.**

В этом разделе описывается создание проекта модульного теста.

1. Откройте проект, который хотите протестировать в Visual Studio.

В целях демонстрации примера модульного теста в этой статье тестируется простой проект консоли C# или C++ "Hello World" с именем **HelloWorld** (**HelloWorldCore** в C#). Пример кода для такого проекта выглядит следующим образом:

namespace HelloWorldCore

public class Program

{

public static void Main()

{

Console.WriteLine("Hello World!");

}

}

1. Выберите узел решения в **обозревателе решений**. Затем в верхней строке меню выберите **Файл** > **Добавить** > **Новый проект**.
2. В диалоговом окне нового проекта найдите проект модульного теста, который хотите использовать.
3. Введите **test** в поле поиска, чтобы найти шаблон проекта модульного теста для тестовой среды, которую вы хотите использовать, например MSTest (C#) или проект **Native Unit Test** (C++), и выберите его.
4. Начиная с Visual Studio 2017 версии 14.8, языки .NET включают встроенные шаблоны для NUnit и xUnit. Для C++ в этом примере выберите проект **Собственный модульный тест**, в котором используется платформа Microsoft Native Unit Test Framework. (Чтобы использовать другую среду тестирования C++, см. [Написание модульных тестов для C/C++](https://docs.microsoft.com/ru-ru/visualstudio/test/writing-unit-tests-for-c-cpp?view=vs-2022)). Сведения о настройке тестового проекта на Python см. в статье [Настройка модульного тестирования для кода Python](https://docs.microsoft.com/ru-ru/visualstudio/python/unit-testing-python-in-visual-studio?view=vs-2022).

**Совет**

Только для C#: вы можете создавать проекты модульных тестов из кода, используя более быстрый метод. Дополнительные сведения см. в разделе [**Создание проектов модульных тестов и методов теста**](https://docs.microsoft.com/ru-ru/visualstudio/test/unit-test-basics?view=vs-2022#create-unit-test-projects-and-test-methods-c). Чтобы использовать этот метод для .NET Core или .NET Standard, требуется Visual Studio 2019.

1. На следующем рисунке показано модульный тест MSTest, поддерживаемый в .NET.

Нажмите **Далее**, выберите имя для тестового проекта и нажмите **Создать**.

Проект добавляется в решение.

1. В проекте модульного тестирования добавьте ссылку на проект, который вы хотите протестировать, щелкнув правой кнопкой мыши **Ссылки** или **Зависимости**, после чего выбрав **Добавить ссылку** или **Добавить ссылку на проект**.
2. Выберите проект, содержащий код, который будет тестироваться, и нажмите **OK**.
3. Добавьте код в метод модульных тестов.

Так вы можете использовать следующий код, выбрав правильную вкладку документации, которая соответствует вашей тестовой среде: MSTest, NUnit или xUnit (поддерживается только в .NET) или C ++ Microsoft Native Unit Test Framework.

* + [MSTest](https://docs.microsoft.com/ru-ru/visualstudio/test/getting-started-with-unit-testing?view=vs-2022&tabs=dotnet%2Cmstest#tabpanel_2_mstest)
  + [NUnit](https://docs.microsoft.com/ru-ru/visualstudio/test/getting-started-with-unit-testing?view=vs-2022&tabs=dotnet%2Cmstest#tabpanel_2_nunit)
  + [xUnit](https://docs.microsoft.com/ru-ru/visualstudio/test/getting-started-with-unit-testing?view=vs-2022&tabs=dotnet%2Cmstest#tabpanel_2_xunit)
  + [Платформа Microsoft Native Unit Test Framework](https://docs.microsoft.com/ru-ru/visualstudio/test/getting-started-with-unit-testing?view=vs-2022&tabs=dotnet%2Cmstest#tabpanel_2_msunittest)

C#Копировать

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System.IO;

using System;

namespace HelloWorldTests

{

[TestClass]

public class UnitTest1

{

private const string Expected = "Hello World!";

[TestMethod]

public void TestMethod1()

{

using (var sw = new StringWriter())

{

Console.SetOut(sw);

HelloWorldCore.Program.Main();

var result = sw.ToString().Trim();

Assert.AreEqual(Expected, result);

}

}

}

}

**Запуск модульных тестов**

1. Откройте [обозреватель тестов](https://docs.microsoft.com/ru-ru/visualstudio/test/run-unit-tests-with-test-explorer?view=vs-2022).

Откройте обозреватель тестов, выбрав **Тест** > **Обозреватель тестов** в верхней строке меню (или нажмите клавиши **CTRL** + **E**, **T**).

1. Запустите модульные тесты, нажав **Запустить все** (или нажмите клавиши **CTRL** + **R**, **V**).

После завершения зеленый флажок указывает, что тест пройден. Красный значок "x" указывает на сбой теста.

**Совет**

Используйте [**обозреватель тестов**](https://docs.microsoft.com/ru-ru/visualstudio/test/run-unit-tests-with-test-explorer?view=vs-2022) для запуска модульных тестов из встроенной платформы тестирования (MSTest) или сторонней платформы тестирования. Вы можете группировать тесты по категориям, фильтровать список тестов, а также создавать, сохранять и запускать списки воспроизведения тестов. Кроме того, с его помощью можно выполнять отладку тестов и анализировать производительность тестов и покрытие кода.