

Создание и демонстрация простого юнит-теста на C#

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Беликов Андрей Геннадьевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается и описывается разработка простого юнит-теста. Приложение разрабатывается на языке программирования C# с помощью IDE Visual Studio. Практическим результатом является разработанный юнит-тест.

Ключевые слова: юнит-тест, C#, библиотека, visual studio

Creating and demonstrating a simple unit test in C#

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Belikov Andrey Gennadievich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses and describes the development of a simple unit test. The application is developed in the C# programming language using the Visual Studio IDE. The practical result is a developed unit test.

Keywords: unit test, C#, library, visual studio

До запуска приложения в производство, когда оно станет доступно пользователям, важно убедиться, что данное приложение функционирует, как и должно, что в нем нет ошибок. Для проверки приложения возможно использовать различные схемы и механизмы тестирования. Одним из таких механизмов являются юнит-тесты. Юнит-тесты позволяют быстро и автоматически протестировать отдельные компоненты приложения независимо от остальной его части. Не всегда юнит-тесты могут покрыть весь код приложения, но тем не менее они позволяют существенно уменьшить количество ошибок уже на этапе разработки.

Цель данной статьи создать и продемонстрировать юнит-тест в среде разработки Visual Studio на языке программирования C#.

В своей работе Н. Н. Додобоев, О. И. Кукарцева, Я. А. Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления основных видов и классификаций языков программирования [1]. В.М. Ицкисон, М.Х. Ахин провели апробацию подхода на системе статического анализа, а также на системе извлечения и поиска шаблонов исходного кода показала, что он позволяет существенно сократить затраты на выполнение регрессионного тестирования [2]. В статье Д.Н. Петрова, Т.Б. Чистяковой, Д.В. Алтуниной была рассмотрена проблематика оценки качества корпоративных веб-ресурсов. Обоснована разработка системы автоматизированного тестирования корпоративных веб-приложений [3].

Создаем проект и называем его « NUnit Test» см. рисунок 1.

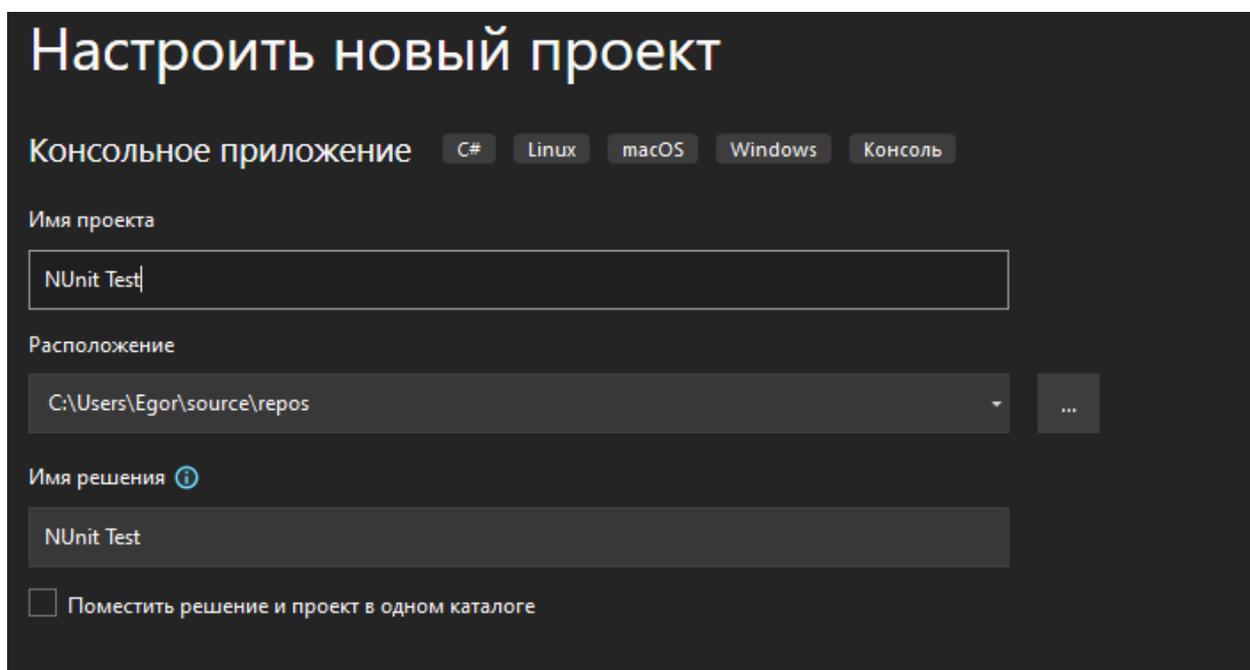


Рис. 1. Создание проекта

Для демонстрации тестов, необходимо создать класс с функцией, которую необходимо протестировать см. рисунок 2-3.

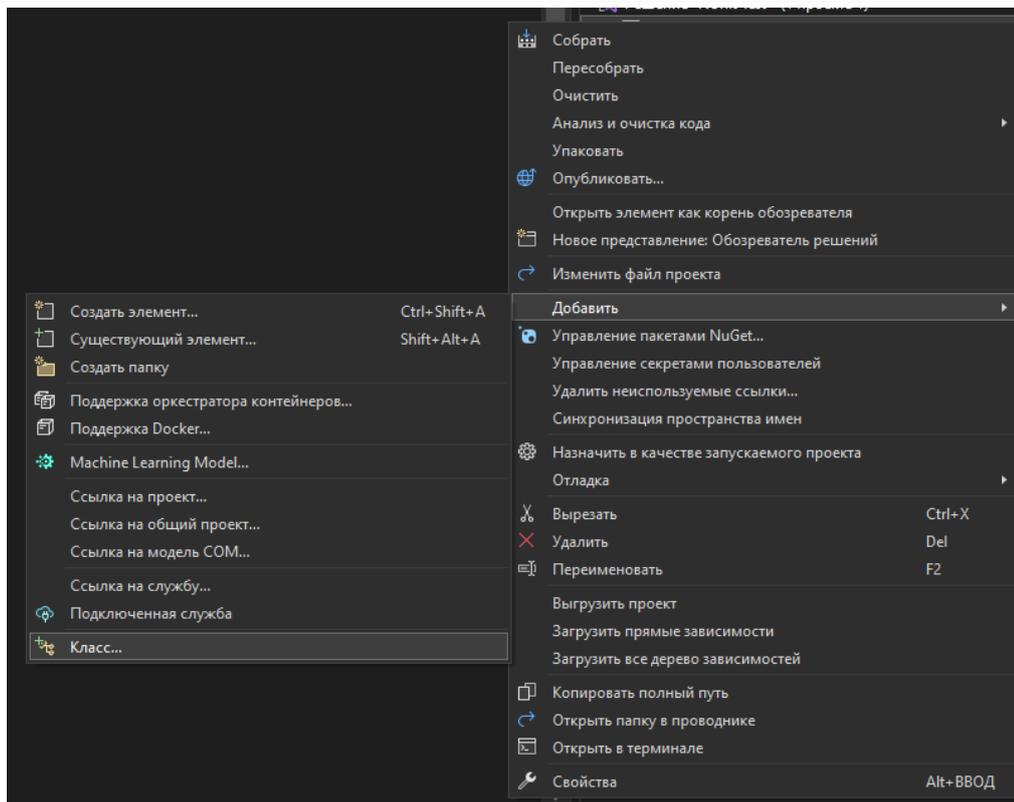


Рис. 2. Создание класса

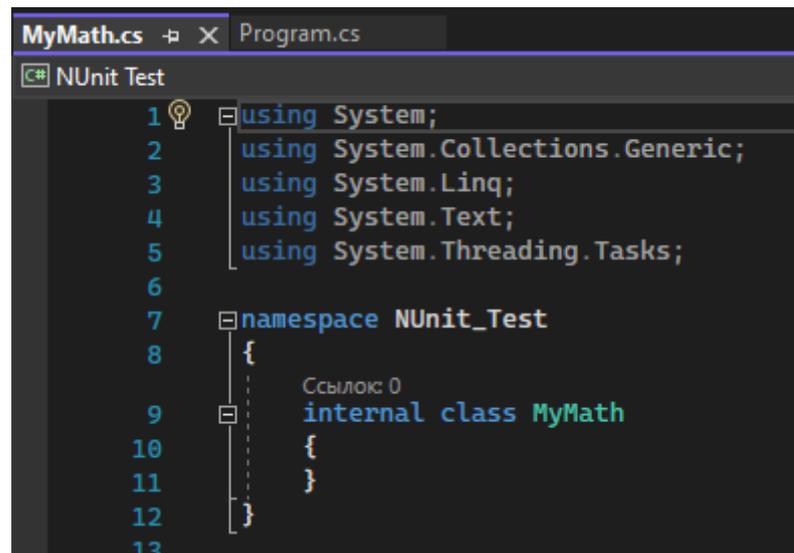


Рис. 3. Создание класса

В данном примере программируем два легких метода с двумя переменными на различные арифметические операции см. рисунок 4.

```
Ссылка: 0
internal class MyMath
{
    Ссылка: 0
    public int Add(int a, int b)
    {
        return a + b * 2;
    }

    Ссылка: 0
    public int Sub(int a, int b)
    {
        return a - b;
    }
}
```

Рис. 4. Методы в классе

Теперь необходимо установить библиотеку «NUnit» из «Manage Nuget Packages» в проект. «NUnit Test» - это среда модульного тестирования с открытым исходным кодом см. рисунок 5-6.

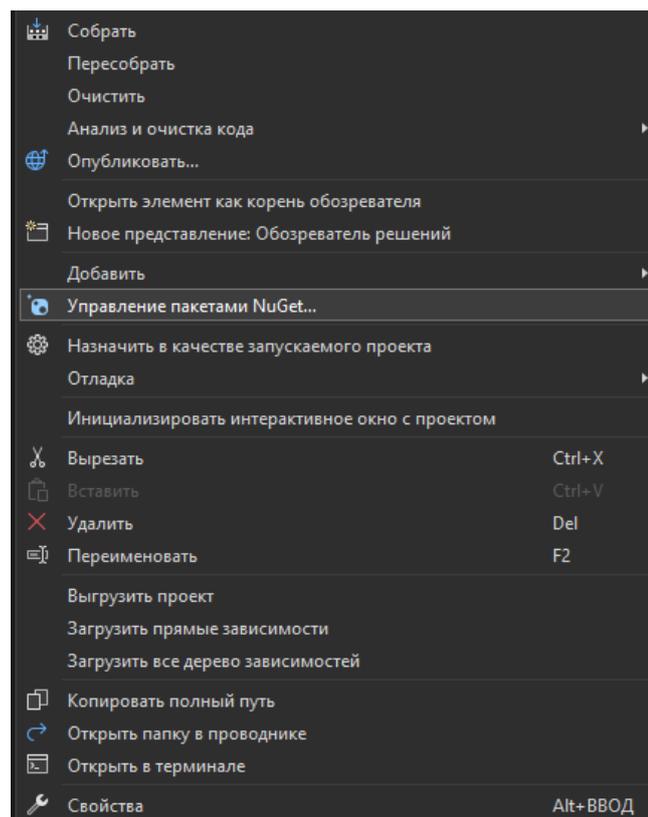


Рис. 5. Запуск «Manage Nuget Packages»

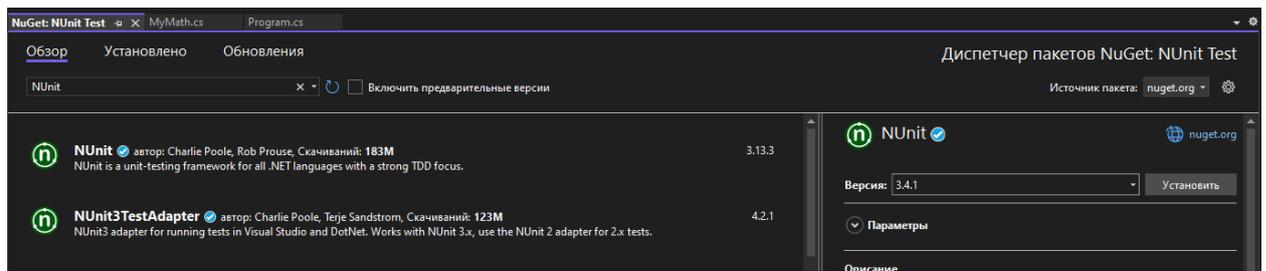


Рис.6. Установка библиотеки «NUnit»

Создаем еще один класс, подключаем библиотеку и создаем два метода с тестовыми значениями см. рисунок 7.

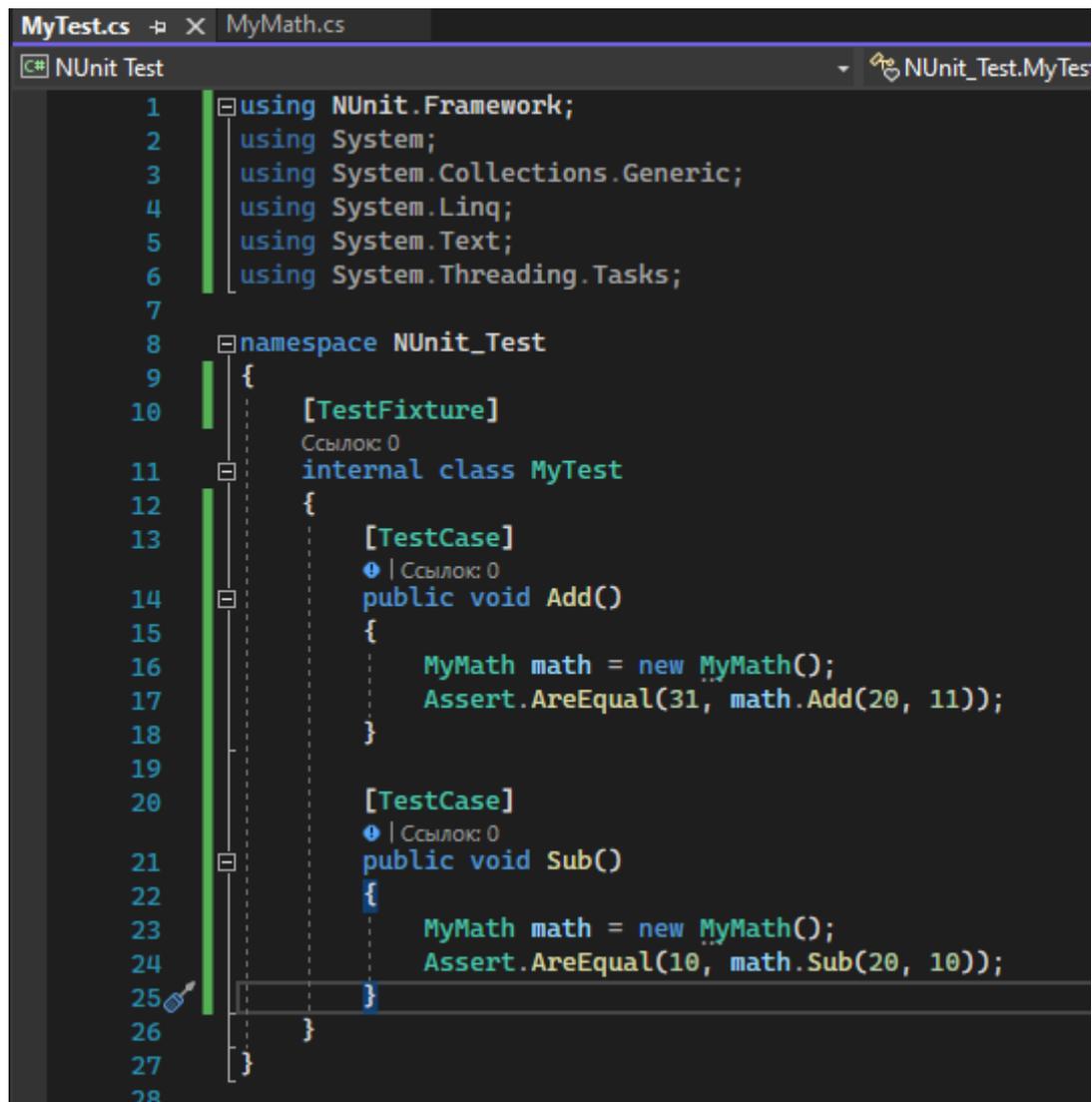


Рис.7. Создания класса с тестами

Теперь запускаем тестирования через специальную вкладку «Тест» см. рисунок 8-9.

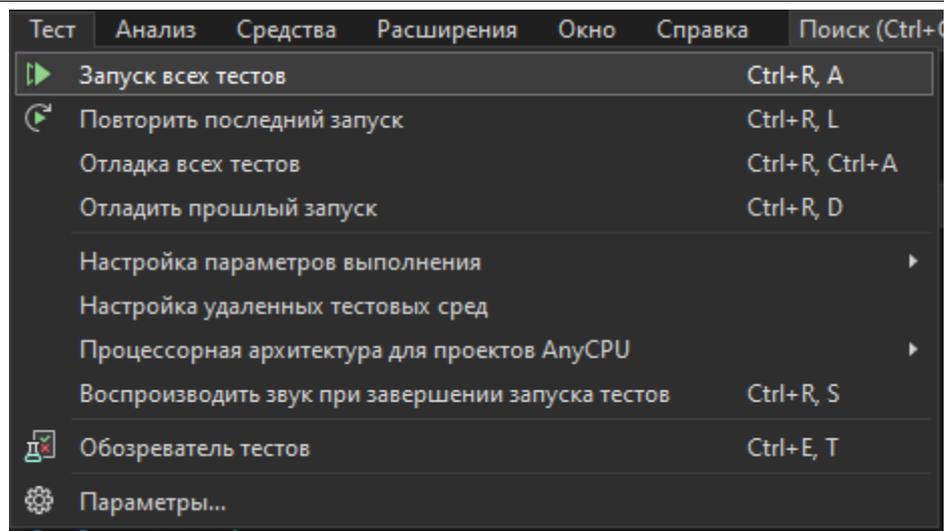


Рис.8. Запуск тестирования

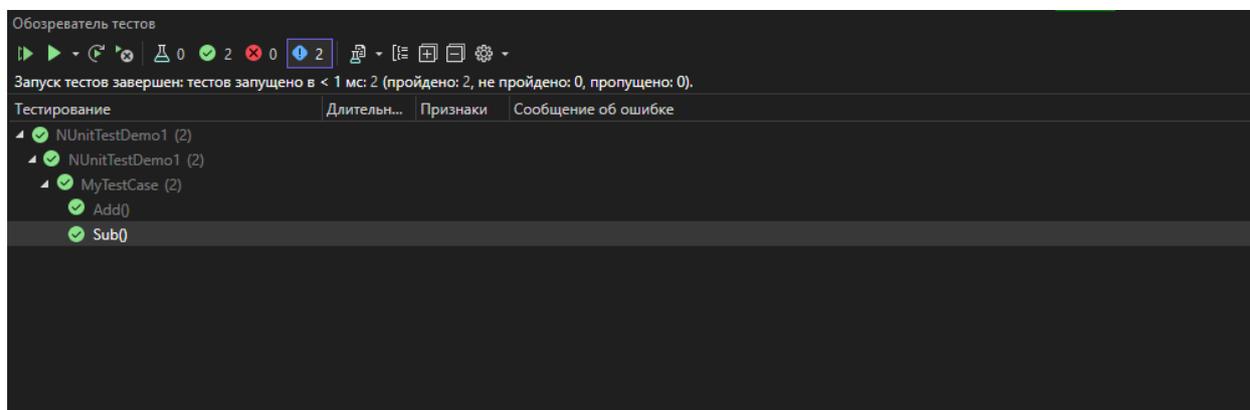


Рис.9. Результат тестов

В данной статье были проанализированы существующие аналоги и методы разработки, а также выбрана среда разработки. Для реализации поставленной задачи отлично подошла разработка с помощью Visual Studio и языка программирования C#. Такой выбор заметно упростил разработку проекта, так как в интернете имеется достаточное кол-во документации. Во время разработки был полученный ценный опыт работы с этим средством разработки.

В итоге был написан и продемонстрирован простой юнит-тест. Данное консольное приложение имеет потенциал к развитию, например, добавление новых функций.

Библиографический список

1. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования // Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
2. Ицыксон В.М., Ахин М.Х. Технологии выборочного регрессионного тестирования // Моделирование производственных процессов и развитие информационных систем. 2011. С. 248-251.

3. Петров Д.Н., Чистякова Т.Б., Алтунина Д.В. Автоматизированное тестирование корпоративных веб-приложений // Приоритетные направления развития науки и технологий. 2022. С. 166-169.