СПОСОБ РЕШЕНИЯ ЗАДАЧИ ТЕСТИРОВАНИЯ КОНСОЛЬНЫХ ПРИЛОЖЕНИЙ В СРЕДЕ VISUAL STUDIO

Гладышев Ю.С., студент гр. ИТб-171, 2 курс
Научный руководитель: Асанов С.А., старший преподаватель
Кузбасский государственный технический университет
имени Т.Ф. Горбачёва
г. Кемерово

Постановка задачи.

Тестирование является одним из важных элементов при создании компьютерной программы. Тестирование можно проводить как вручную, так и с использованием средства автоматизации процесса. Основную часть автоматизированных тестов составляют модульные тесты или unit-тесты. Наиболее распространены модульные тесты для приложений с графическим интерфейсом, где можно пронаблюдать за изменением хода программы. Применение ubit-тестов в консольных приложениях затруднено, т.к. интерфейс программы чаще всего не вынесен в отдельные модули, а «перемешан» с кодом, выполняющим вычислительные задачи, а в случае простейшего консольного приложения программист вообще может обойтись только одной главной функцией.

В итоге основных инструментом тестирования становится ручное, при котором при любом внесении изменений в программу приходится заново компилировать данную сборку и открывать консоль для введения исходных данных для проверки.

Для решения данной проблемы Microsoft в 2012 году выпустили фреймворк Microsoft Fakes. Это изоляционный фреймворк, который является логическим продолжением экспериментального проекта Microsoft Moles. Microsoft Fakes позволяет изолировать часть кода, которую необходимо протестировать, заменяя остальную часть приложения заглушками и оболочками. Благодаря этим инструментам мы можем тестировать код, даже если остальная часть программы не работает/существует.

Фреймворк Fakes предлагает два варианта на выбор:

- Заглушка – это инструмент, который заменяет класс на Mock-объект (буквально: «объект-пародия», «объект-имитация» или «подстановка» – тип объектов, реализующих заданные аспекты моделируемого программного окружения). При этом приложение нужно простроить так, чтобы тестируемый компонент имел зависимости только от интерфейса, а не от других компонентов программы.
- Оболочка – меняет непосредственно скомпилированный код приложения во время выполнения, чтобы вместо вызова нужного метода запускался метод, предоставляемый тестом. Этот метод

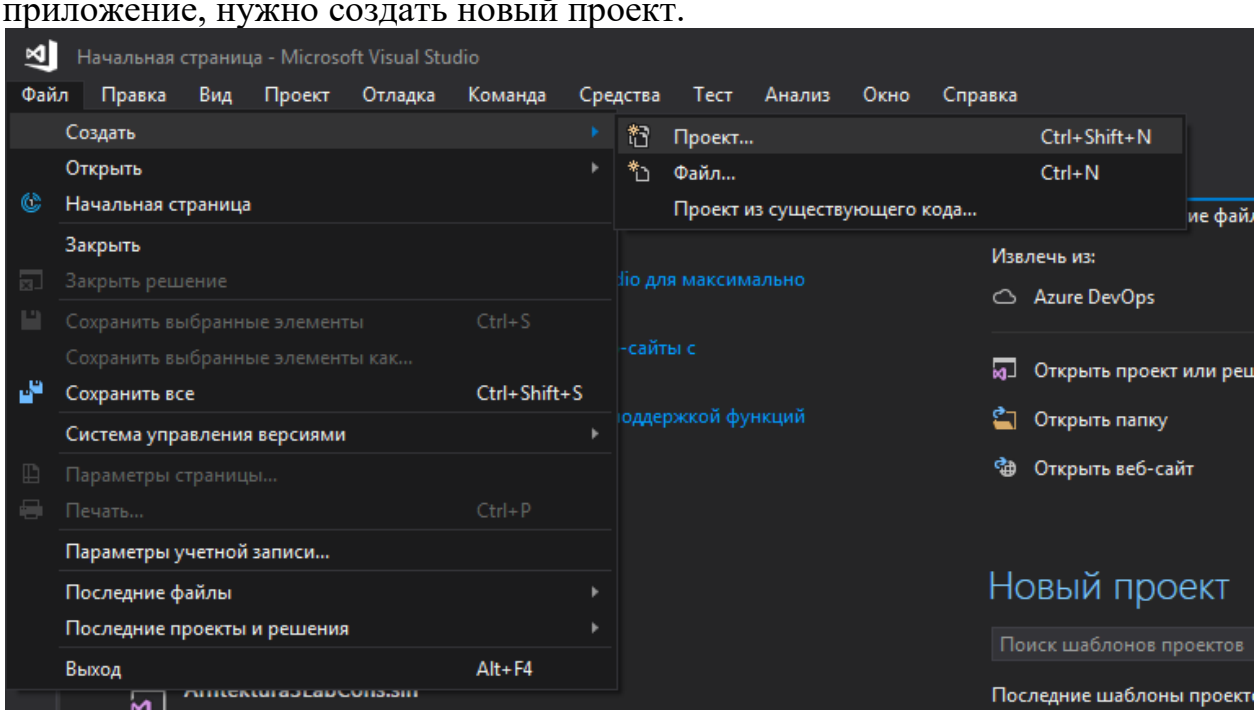
можно использовать для замены вызовов сборок, которые невозможно изменить.

В итоге, заглушки чаще используют для вызова в пределах решения Visual Studio, а оболочки используют другие сборки, с указанными ссылками на них. В части производительности оболочки выполняются медленнее из-за перезаписи кода во время выполнения, а заглушки не сказываются на производительности, выполняясь также быстро, как виртуальный метод.

Заглушки можно использовать только для реализации интерфейсов, поэтому их нельзя использовать для статических методов, не виртуальных методов и запечатанных (sealed) методов. Оболочки не могут реализовываться с интерфейсами и чистыми абстрактными методами, так как они не имеют тела метода.

Применяемый алгоритм тестирования.

Для использования оболочек, чтобы протестировать консольное приложение, нужно создать новый проект.

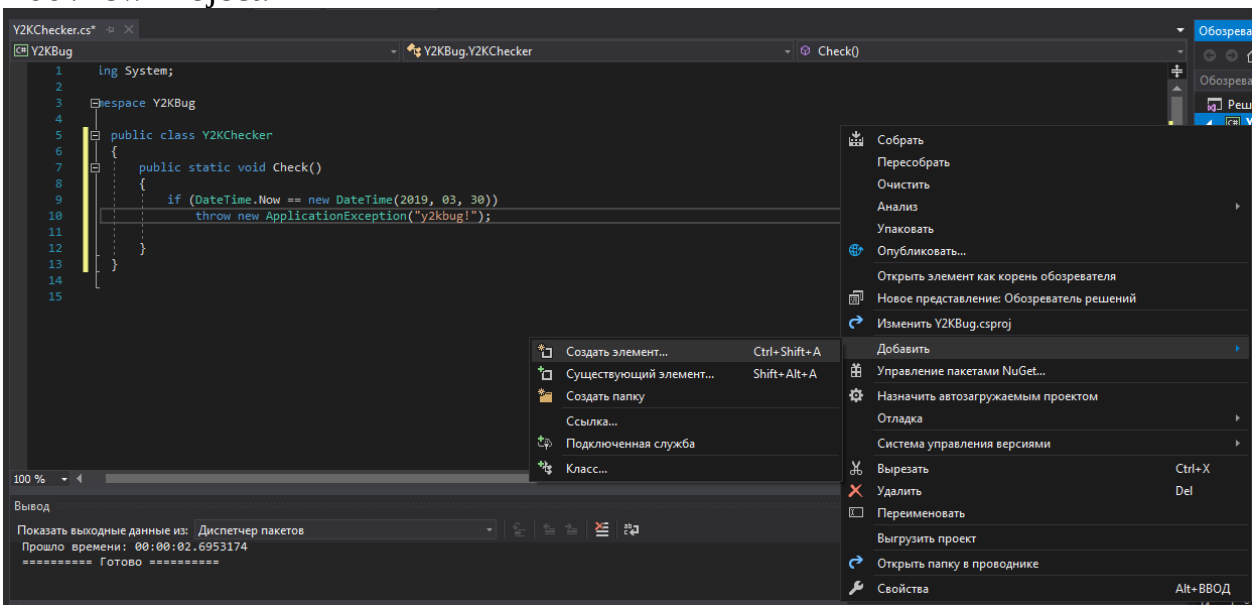


Затем зайти в Установленные\Шаблоны\Visual C#, выбрать проект библиотеки классов и назвать его Y2KBug.

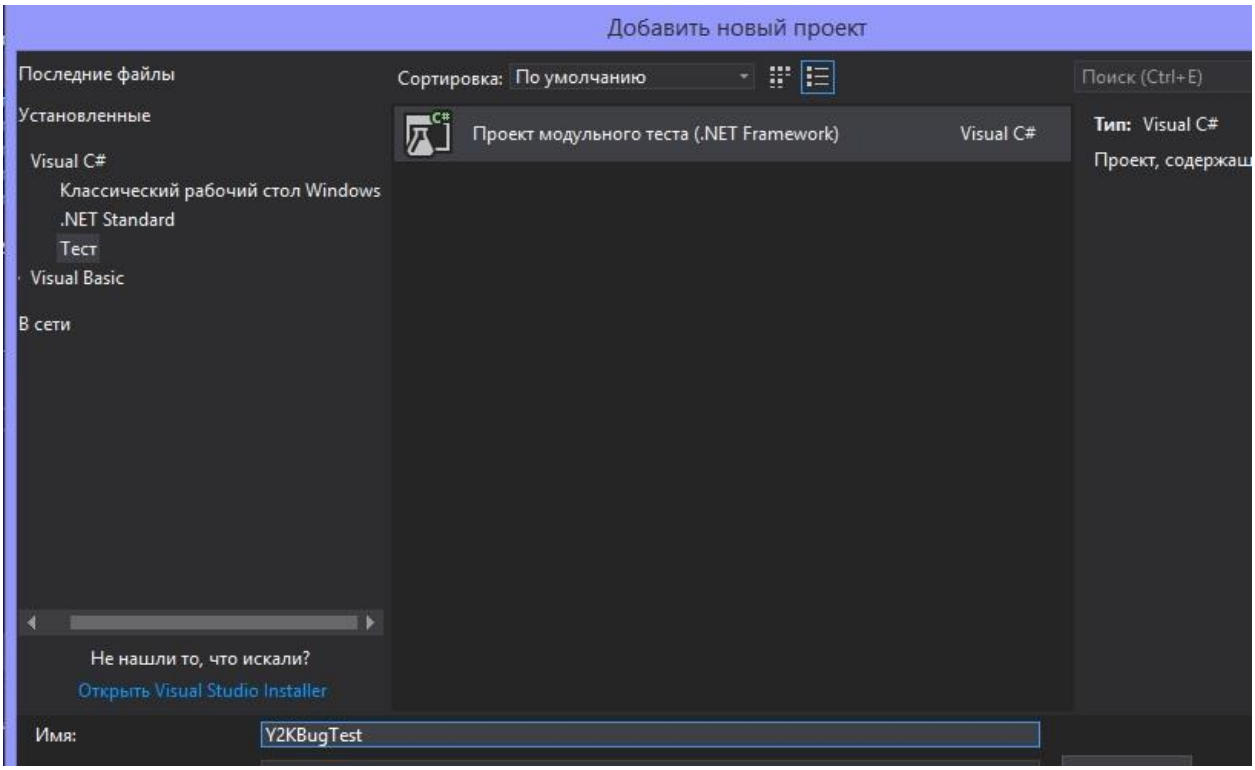
После создания проекта, он автоматически сгенерирует файл класса (Class.cs), затем переименуйте созданный файл класс в Y2KChecker.cs. Добавьте следующий код в ваш файл Y2KChecker.cs. Этот код должен проверить текущую дату с заданным параметром и, в случае совпадения, выбросить исключение.

```
Y2KChecker.cs*
Y2KBug | Y2KBug.Y2KChecker | Check()
1   ing System;
2
3   espace Y2KBug
4
5   public class Y2KChecker
6   {
7       public static void Check()
8       {
9           if (DateTime.Now == new DateTime(2019, 03, 30))
10              throw new ApplicationException("y2kbug!");
11
12      }
13  }
14
15
```

Затем нужно добавить тестовый проект в файл решения. Нажмите правой кнопкой мыши Solution “Y2KBug” в Solution Explorer и кликните на Add\New Project.

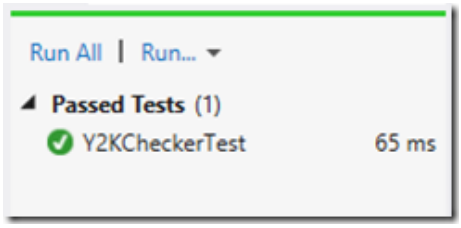


В разделе Установлено\Visual C# выберите тестовый проект Unit Test и назовите его Y2KBug.Test. Это автоматически сгенерирует файл класса с именем UnitTestProject1.cs. Переименуйте файл класса в Y2KBugTest.cs.



Для включения возможностей тестирования необходимо в решение добавить ссылки на сборки Microsoft.QualityTools.Testing.Fakes, mscorlib.4.0.0.0.Fakes и System.4.0.0.0.Fakes в папку Fakes.

Корректно пройденные тесты среда Visual Stuio традиционно маркирует зеленым цветом.



Выводы

Задачу тестирования консольных приложений решить традиционным подходом unit-тестирования крайне сложно. Однако, применение специализированных инструментов, предоставляемых фреймворком Fakes, позволяет решить данную проблему, даже несмотря на то, что изначально данный фреймворк создавался с другими целями. Применение возможностей оболочек фреймворка Fakes позволяет решить, в том числе, и задачу проведения тестирования приложений, состоящих всего из одной главной функции.

Список литературы

1. Shims and Stubs and the Microsoft Fakes Framework [Электронный ресурс] – Режим доступа: https://www.nwcadence.com/blog/shims-and-stubs-and-the-microsoft-fakes-framework – Загл. с экрана. (28.03.2019)
2. Использование платформы MSTest в модульных тестах [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-

ru/visualstudio/test/using-microsoft-visualstudio-testtools-unittesting-members-in-unit-tests?view=vs-2017 – Загл. с экрана. (28.03.2019)

3. Модульное тестирование [Электронный ресурс] – Режим доступа: http://www.intuit.ru/studies/courses/4616/209/lecture/5409 – Загл. с экрана. (28.03.2019)