

Лабораторная работа №5

Тема: Разработка плана и проведение модульного тестирования ПО.

Цели: Сформировать умение разрабатывать план и проводить модульное тестирование ПО.

Литература: Котляров В.П. Основы современного тестирования программного обеспечения, разработанного на С#. Учебное пособие

Задание: Изучите теоретическую часть и выполните практические задания, ответьте на контрольные вопросы.

Теоретическая часть

Модульное тестирование, или юнит-тестирование (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

Для модульного тестирования необходимо использовать драйверы и заглушки.

Unit (Элемент) – наименьший компонент, который можно скомпилировать.

Драйверы – модули тестов, которые запускают тестируемый элемент.

Заглушки – заменяют недостающие компоненты, которые вызываются элементом и выполняют следующие действия:

- возвращаются к элементу, не выполняя никаких других действий;
- отображают трассировочное сообщение и иногда предлагают тестеру продолжить тестирование;
- возвращают постоянное значение или предлагают тестеру самому ввести возвращаемое значение;
- осуществляют упрощенную реализацию недостающей компоненты;
- имитируют исключительные или аварийные условия.

Для реализации юнит-тестов будем использовать NUnit.

NUnit – открытая среда юнит-тестирования приложений для .NET, которая позволяет создавать автоматические тесты. NUnit легко интегрируется с другими средствами разработки ПО, тесты NUnit можно запустить из консоли, графической оболочки, скрипта сборки, или любым другим программным способом.

Интеграция с MS Visual Studio

Сначала необходимо скачать NUnit с сайта. Затем необходимо разархивировать и запустить установщик программы.

Для того, чтобы интегрировать NUnit в Visual Studio, необходимо добавить его как внешний инструмент. Для этого в меню «Tools» необходимо выбрать пункт «External Tools». В появившемся окне необходимо указать:

- Title: название пункта меню, например, NUnit.
- Command: Путь до графического интерфейса NUnit, например, C:\Program Files (x86)\NUnit 2.5.3\bin\net-2.0\nunit.exe.
- Arguments: Аргументы передаваемые приложению, т.е. путь до сборки, например, \$(ProjectDir)bin/Debug/\$(TargetName)\$(TargetExt).
- Initial directory - Рабочая папка - \$(TargetDir).

После этого, если выбрать проект с тестами и выбрать в меню Tools пункт NUnit, стартует графическая оболочка программы.

Для того, чтобы тесты запускались при запуске отладки проекта, необходимо настроить запуск NUnit в свойствах проекта (вкладка Debug). Необходимо найти вкладку «Start external program»(запускать внешнюю программу) и задать путь до NUnit, например, C:\Program Files (x86)\NUnit 2.5.3\bin\net-2.0\nunit.exe.

Далее, во вкладке Command line arguments необходимо задать путь до файла настроек NUnit и категорию для автозапуска, например, файл настроек ..\..\NunitTests.nunit и категория /include:Debug /runselected (разделяются пробелом).

В папке проекта необходимо создать файл с названием NunitTests.nunit со следующим содержанием(название сборки необходимо заменить на свое):

```
<NUnitProject>
<Settings activeconfig="Debug" />
<Config name="Debug">
<assembly path="bin\Debug\[название сборки].dll" />
</Config>
<Config name="Release"></Config>
</NUnitProject>
```

В дальнейшем, тесты, которые необходимо запускать при отладке, отмечаются атрибутом [Test, Category(«Debug»)].

Разработка тестов

Сначала необходимо создать проект библиотечного класса(рисунок 1). Для этого, необходимо выбрать пункт меню «File -> New -> Project», в открывшемся окне выбрать Visual C # в установленных шаблонах и выбрать тип приложения «Class library». После этого необходимо задать название проекта и нажать «ОК». В результате созданный проект появится в Solution Explorer (в правом верхнем углу Visual Studio).

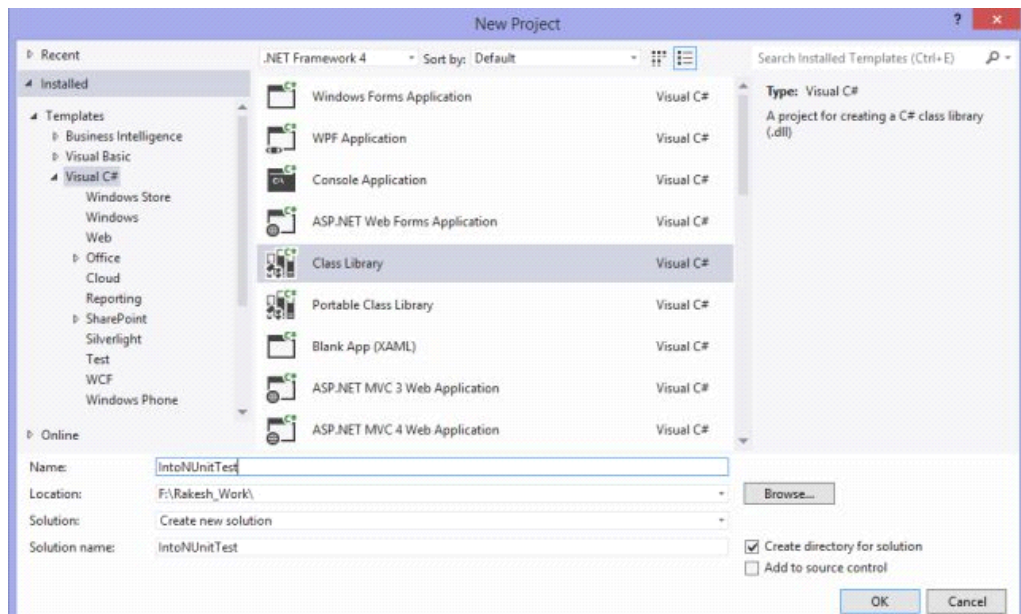


Рисунок 1 – Окно создания проекта

Далее необходимо добавить ссылку на NUnit в проект. Для этого необходимо кликнуть правой кнопкой мыши на ссылке, выбрать «Add reference -> Browse », затем nunit.framework.dll и «OK».

Далее необходимо создать тестовый класс(рисунок 2). Для этого необходимо кликнуть правой кнопкой мыши на проекте, выбрать Add -> Class, ввести имя и нажать кнопку "Add".

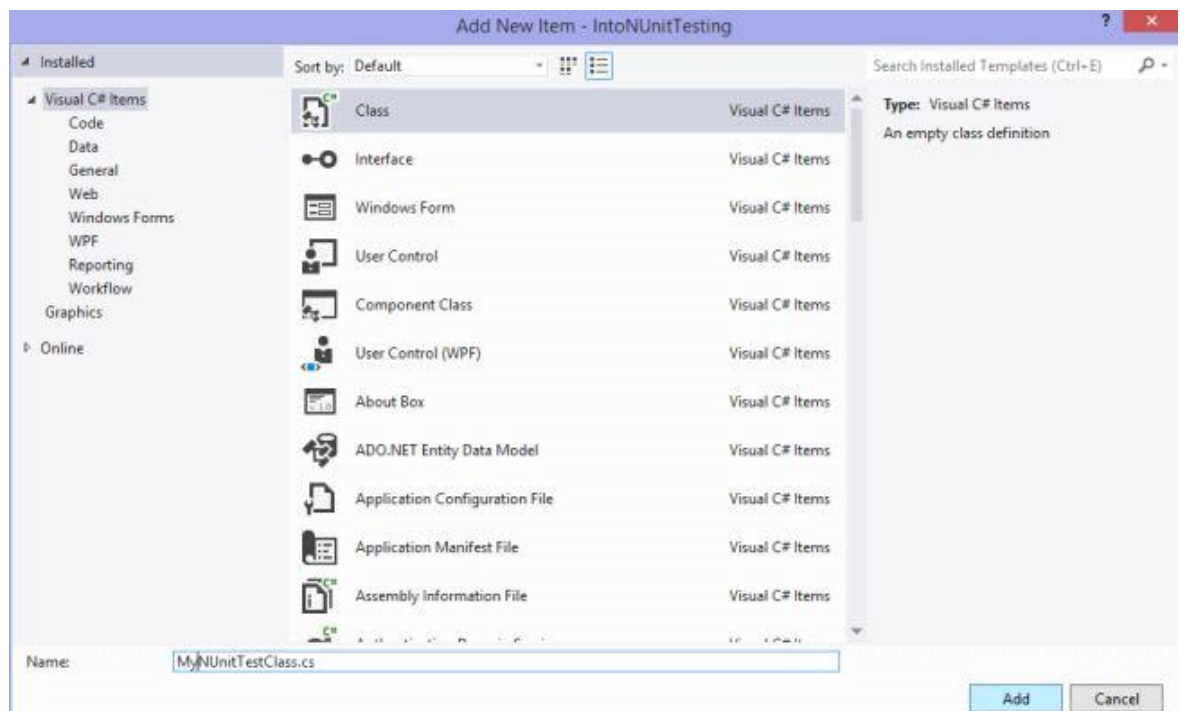


Рисунок 2 – Окно добавления класса

После этого отобразится созданный класс(рисунок 3).

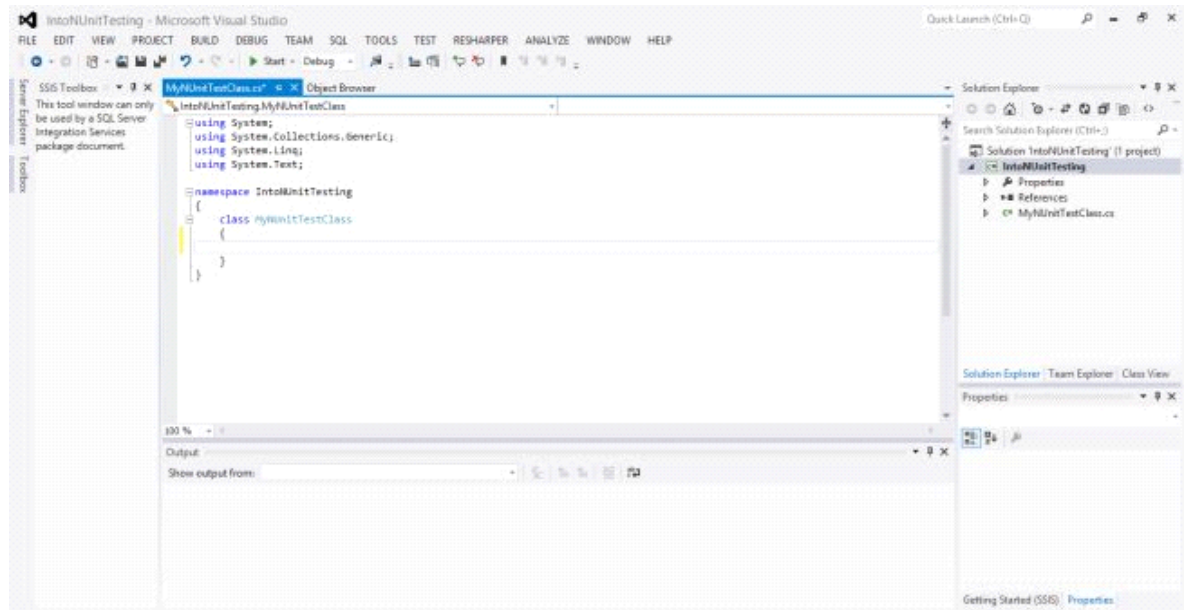


Рисунок 3 – Код только что созданного тестового класса

В код необходимо добавить строчку `using NUnit.Framework;`

При написании кода тестового класса(рисунок 4) необходимо соблюдать некоторые условия:

- каждый класс должен содержать атрибут `[TestFixture]` и должен быть общедоступен;
- в каждом методе должен присутствовать атрибут `[Test]`;
- оператор подтверждения об отсутствии ошибок: Булевские значения, описывающие, что должно быть ключевым словом, когда выполняется действие;
- ожидаемое исключение: один из типов исключения, который мы ожидаем во время выполнения тест-метода;
- установка: программа, которая запускается перед выполнением каждого тест-метода (например, регистрация в системе конкретного пользователя или инициализация одноэлементных классов).
- демонтаж: программа, запускается после окончания каждого тест-метода (например, удаление строк из таблицы, которые были вставлены во время теста).

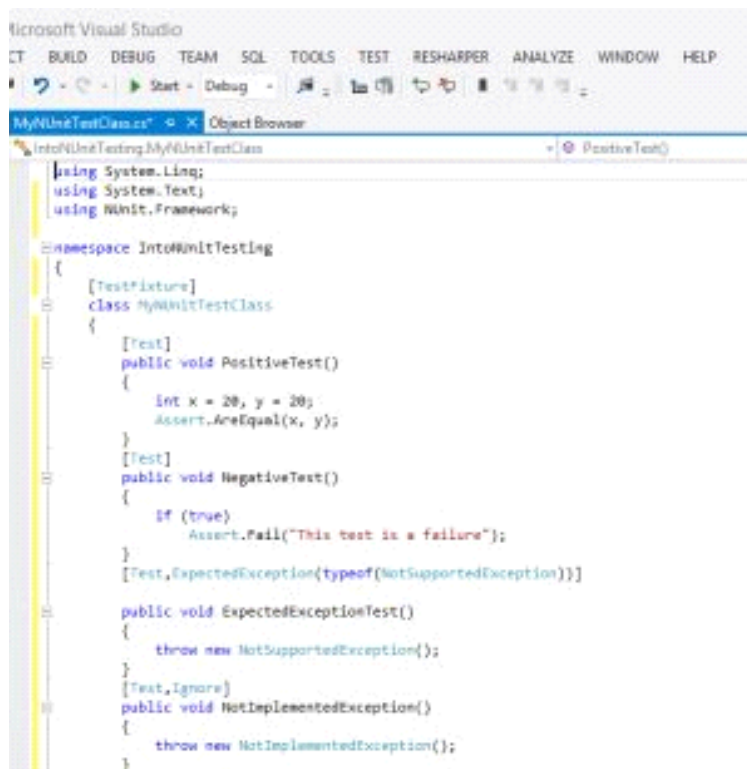


Рисунок 4 – Пример кода тестового класса

Запуск тестов

После написания всех тестов в тестовом классе, необходимо запустить тесты, чтобы проверить, проходят ли они успешно. Для запуска тестов нужно перейти в папку NUnit, выбрать NUnit Application (.exe) и дважды кликнуть по ней, затем выбрать команду File-> Open Project, выбрать проект, затем кликнуть Run(рисунок 5).

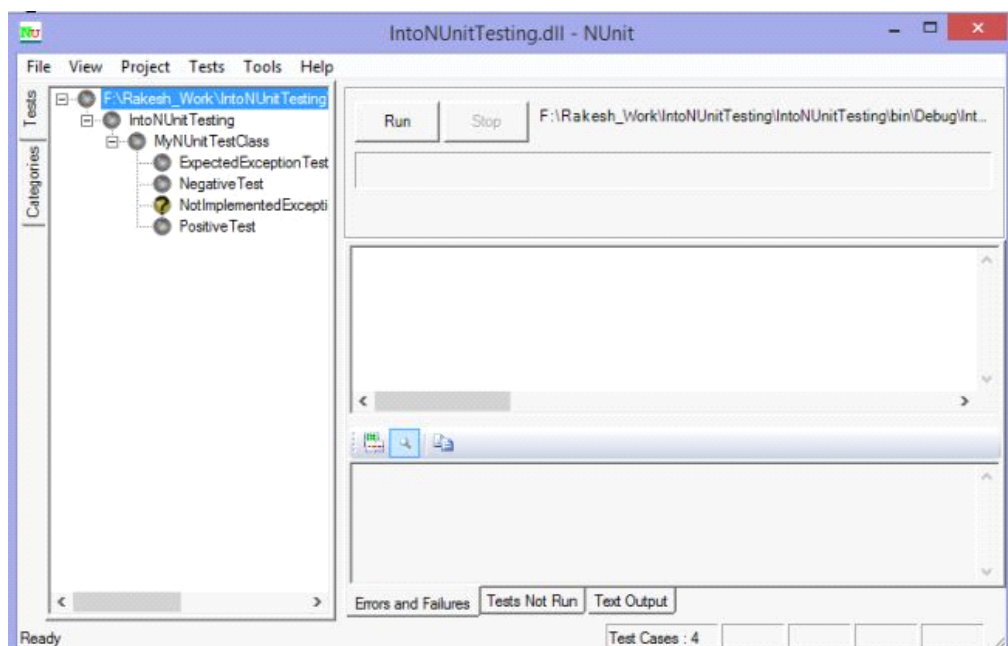


Рисунок 5 – Главное окно программы NUnit

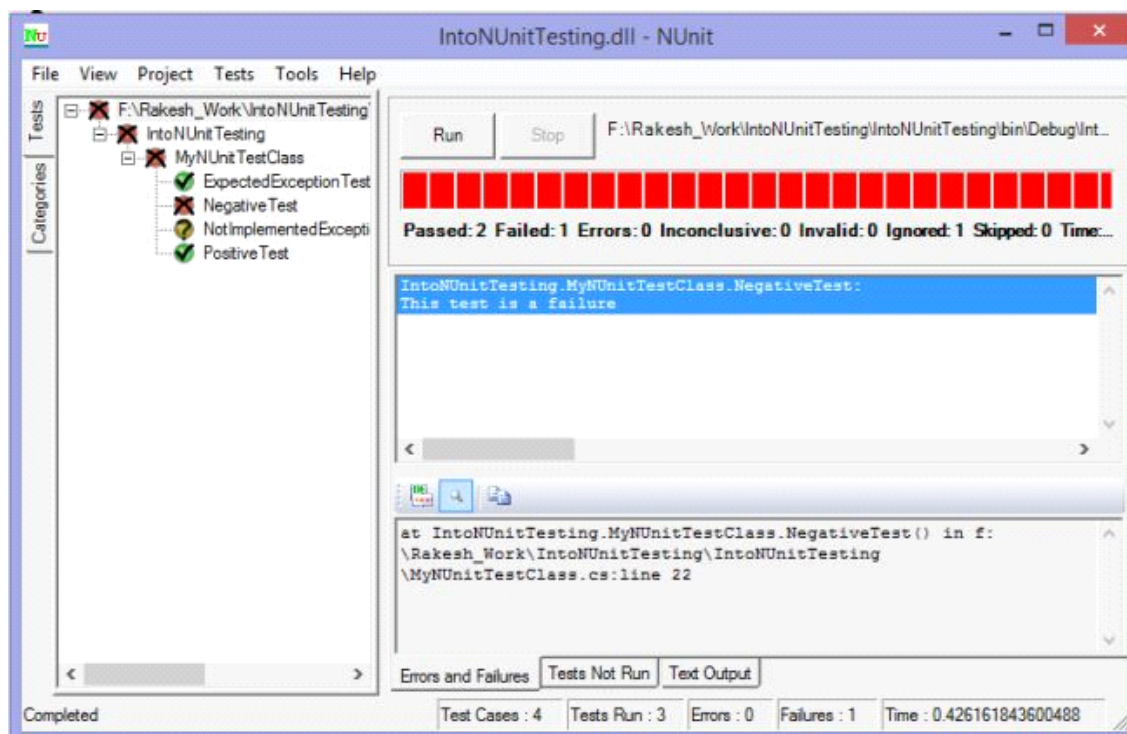


Рисунок 6 – Запуск тестов в NUnit.

После запуска тестов отобразится количество тестов, которые не сработали и количество невыполненных тестов.

Для запуска определенного теста, необходимо кликнуть по нему правой кнопкой мыши и выбрать Run test(s).

Задания для самостоятельного выполнения:

Задание 1.

Изучить средства тестирования, доступные в Visual Studio – Unit Testing Framework, NUnit. Допускается использование альтернативных средств (необходимо согласовать с преподавателем).

Разработать набор unit-тестов.

Обеспечить максимально возможное покрытие кода тестами.

В качестве справочных материалов предлагается набор статей:
<http://msdn.microsoft.com/ru-ru/library/dd264975.aspx>

Шкала оценивания результатов деятельности

Задания	Количество баллов
1	1-10

Контрольные вопросы:

1. Модульное тестирование – это...?

2. Что называется юнитом?
3. Для каких целей предназначен драйвер?
4. Для каких целей предназначены заглушки?
5. При каком условии покрытия кода тестами модульное тестирование считается успешно выполненным?