Отчёт  
по практической работе №10  
по дисциплине «МДК 02.02 Инструментальные средства разработки программного обеспечения»  
Тема: «Разработка тестовых модулей проекта для тестирования отдельных модулей»

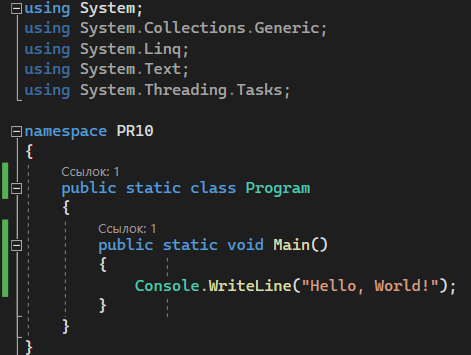
|  |  |  |
| --- | --- | --- |
| Студент гр. 22 ИТ |  | Липинский К.С. |
| Преподаватель |  | Быковский З.С. |

### Цель работы

Научиться выполнять модульное тестирование программного продукта средствами Visual Studio.

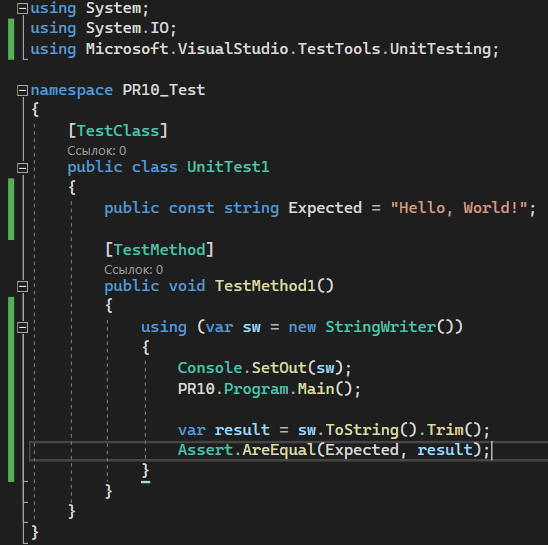
### Выполнение работы

Создадим проект консольного приложения со следующим содержанием (рис. 1)



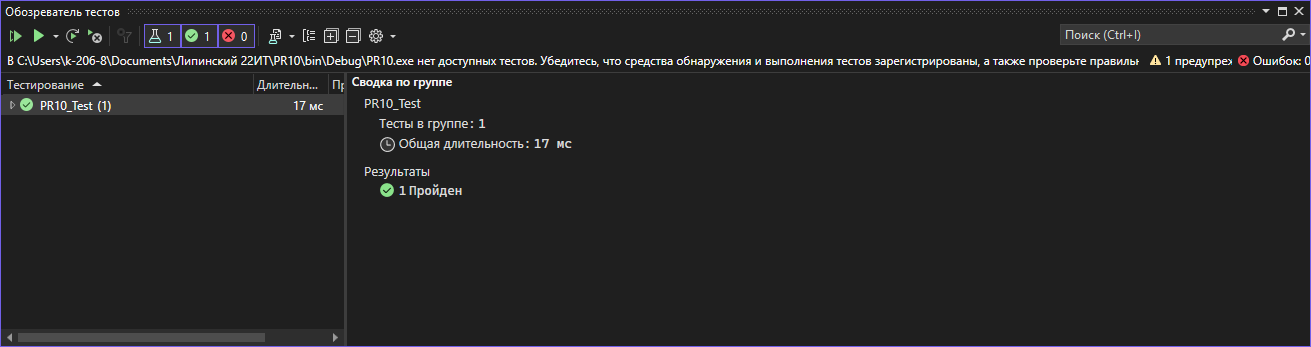
Рисунок

Создадим проект тестов со следующим содержанием (рис. 2):



Рисунок

Запустим его. Результат (рис. 3)



Рисунок

### Контрольные вопросы.

1. Цель модульного тестирования заключается в проверке отдельных модулей (компонентов) программного обеспечения на предмет их корректности и соответствия требованиям.

*Основные задачи:*

* Выявление дефектов на ранних стадиях разработки.
* Проверка функциональности модуля в изоляции от других компонентов.
* Обеспечение высокого уровня покрытия кода тестами.
* Упрощение процесса отладки и исправления ошибок.

1. Две основные проблемы при модульном тестировании:

* Изоляция модуля: Трудности в изоляции модуля от других компонентов системы, что может привести к необходимости использования моков (mock objects) или стабов (stubs).
* Полнота тестирования: Обеспечение достаточного покрытия всех возможных сценариев использования модуля, что может быть сложным из-за большого количества входных данных и состояний.

1. Модуль — это самостоятельная часть программы, выполняющая определенную функцию. Границы модуля определяются его интерфейсами, то есть набором входных и выходных данных, а также методами, которые он предоставляет для взаимодействия с другими компонентами.
2. Тестирование классов включает проверку их методов и свойств на корректность работы. Это может включать:

* Проверку корректности возвращаемых значений методов.
* Проверку изменения состояния объекта после вызова методов.
* Проверку обработки исключительных ситуаций.

1. Степень полноты тестирования класса определяется покрытием кода тестами. Это может включать:

* Покрытие всех методов класса.
* Покрытие всех ветвей кода (например, всех условий в if-else конструкциях).
* Покрытие всех возможных состояний объекта.

1. Протоколирование состояний объектов включает запись их текущих значений и изменений, происходящих в процессе выполнения тестов. Это может быть полезно для:

* Отладки и анализа причин сбоев.
* Проверки корректности изменения состояния объекта после вызова методов.

1. Тестирование изменений включает:

* Регрессионное тестирование: проверка, что новые изменения не нарушили существующую функциональность.
* Проверка новых или измененных функций на соответствие требованиям.

1. Проектирование тестового окружения включает:

* Создание изолированной среды для выполнения тестов.
* Использование моков и стабов для симуляции взаимодействия с другими компонентами.
* Автоматизация запуска тестов и сбора результатов.

### Вывод.

В данной практической работе я освоил выполнение модульного тестирования программного продукта с использованием средств Visual Studio. Я научился создавать и запускать юнит-тесты для отдельных модулей, а также анализировать результаты тестирования для выявления и исправления ошибок на ранних стадиях разработки. Это позволило мне повысить качество кода и уверенность в его корректности.