Лабораторная работа No**2.**

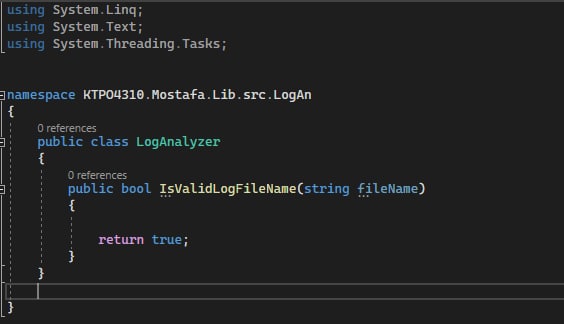
Использование заглушек для разрыва зависимостей

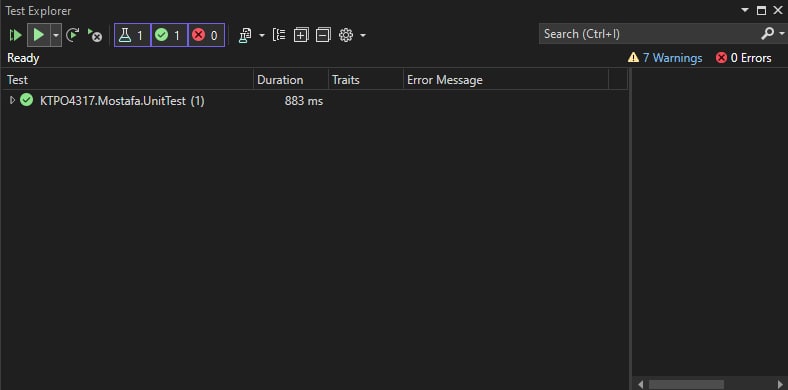
Цель работы

Приобретение практических навыков использования заглушен для разрыва зависимостей при автономном тестировании модулей, практика использования тестового каркаса NUnit.

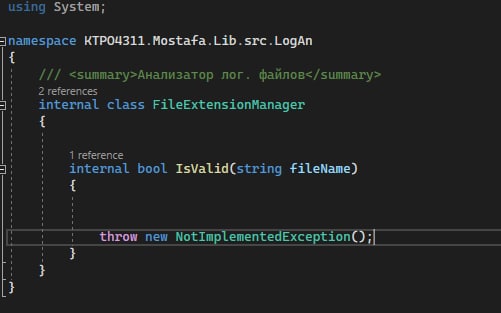
Порядок выполнения работы

**1.** Подготовка проекта



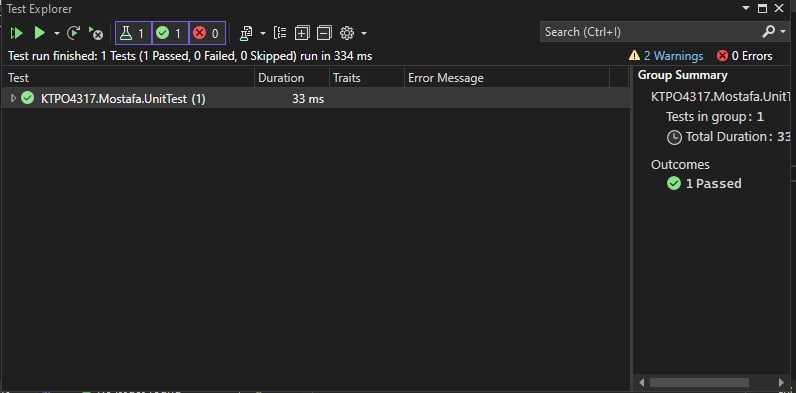


**2.** Рефакторинг проекта для повышения тестопригодности**:** выделение в отдельный класс операций с файловой системой



Text

Description automatically generated



**3.** Рефакторинг проекта для повышения тестопригодности**:** выделение интерфейса

1. Замените у класса FileExtensionManager и его метода на IsValid модификатор доступа internal на public.

Text

Description automatically generated

1. Выделите интерфейс FileExtensionManager, создать его в отдельном файле: Быстрые действия и рефакторинг->Извлечение интерфейса

Graphical user interface, text, application, email

Description automatically generated

3. Объявите переменную диспетчера расширений в тестируемом методе IsValidLogFileName как интерфейс.

Text

Description automatically generated

Text

Description automatically generated

**4.** Внедрение зависимости**.** Внедрение подделки через конструктор**.**

1. Объявите в классе LogAnalyzer поле типа IExtensionManager, конструктор с параметром тоже типа, инициализирующий это поле. В методе IsValidLogFileName замените создание нового класса на

обращение к полю.

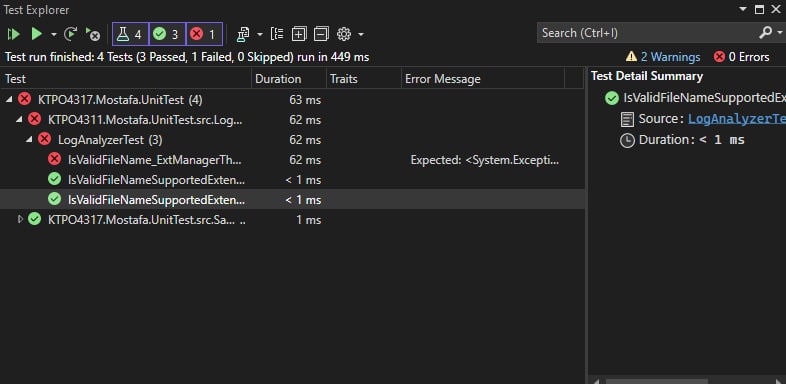
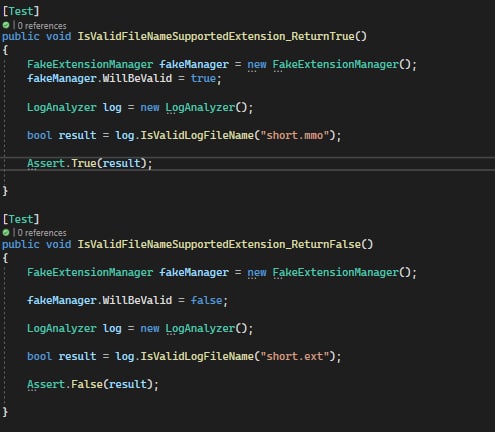
1. Создайте заглушку FakeExtensionManager, реализующую интерфейс

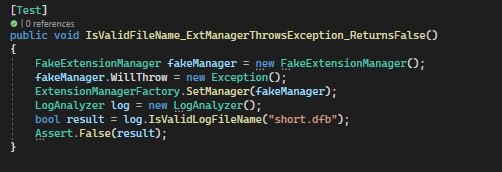
IExtensionManager. Разместите исходный код поддельного класса в том же файле, что и тестовый класс.

Text

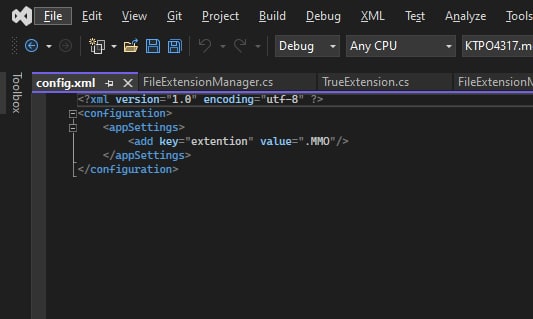
Description automatically generated

3. Создайте тестовый метод IsValidFileName\_NameSupportedExtension\_ReturnsTrue для тестового сценария «Для поддерживаемых расширение метод возвращает true». Создайте и настройте заглушку для этого сценария. Используйте ее при создании экземпляра LogAnalyzer. Добавьте вызов метода IsValidLogFileName и проверку результата.

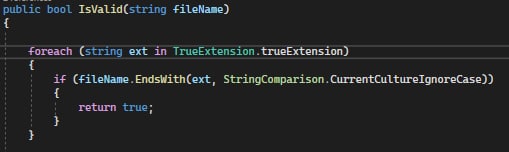


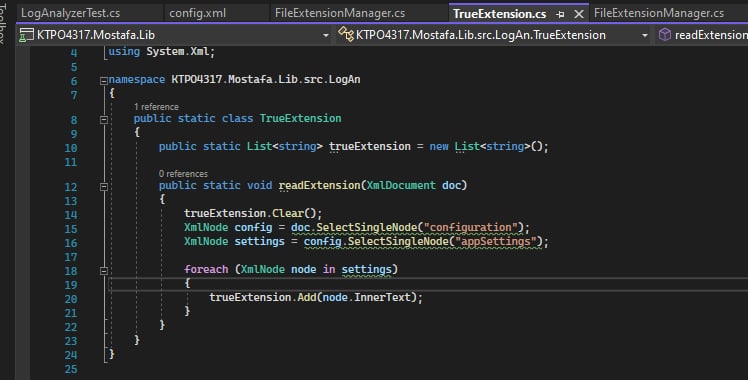


1. Добавьте в проект файл настроек. Добавьте в него информацию о правильно расширении.



1. Реализуйте логику класса FileExtensionManager, так, чтобы допустимое расширение читалось из файла настроек.





5. Добавьте в KTPO4317.SabitovBI.Service вызов класса LogAnalyzer для

разных имен файлов, с правильными и не правильными расширениями.

Результат вызова выведете на консоль.

6. Протестируйте и отладьте приложение.

7. Запустите тесты.

