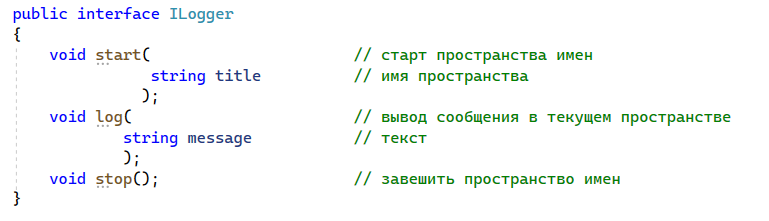
**Паттерны проектирования**

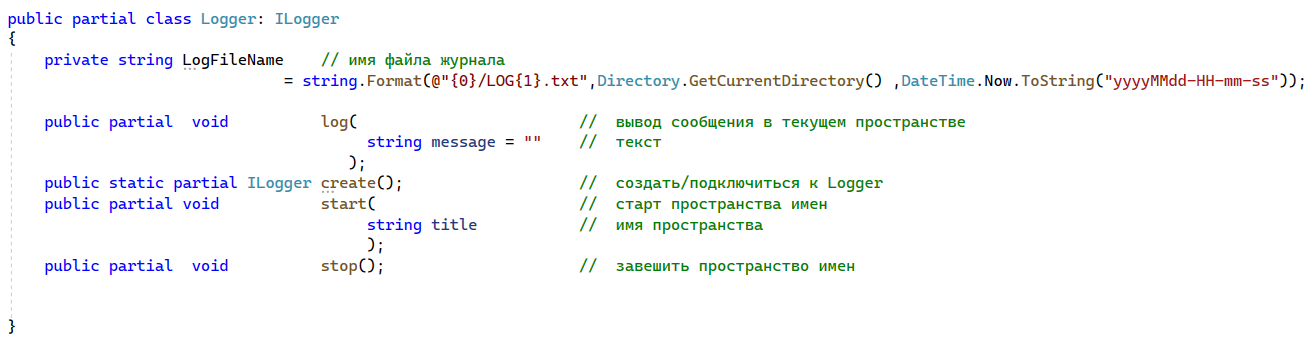
**ИСиТ,ПОИБМС-2**

**Лабораторная №4: *Singleton***

**Задание 1**

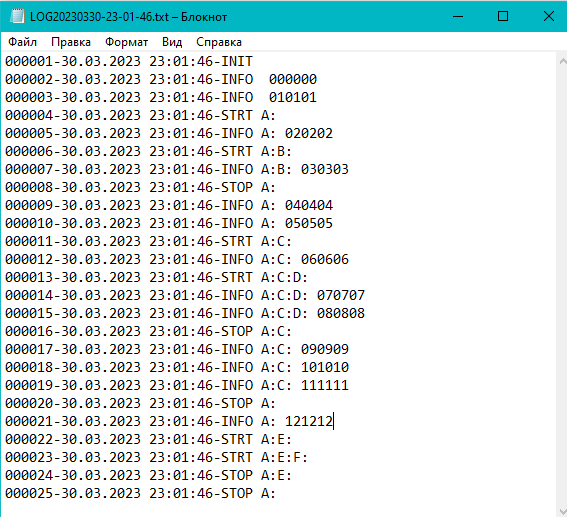
1. Разработайте библиотеку классов ***Lec04LibN***, реализующую паттерн ***Singleton***.
2. Библиотека позволяет выводить сообщения (осуществлять трассировку) в журнал.
3. Библиотека ***Lec04LibN*** предоставляет пользователю (программисту) доступ к интерфейсу ILogger и классу Logger.





1. Для проверки используйте тест представленный ниже.





**Задание 2.Ответье на вопросы**

1. Поясните суть паттерна ***Singleton,*** что дает его применение.
2. Поясните реализацию класса ***Logger:***

* статический метод create (почему статический);
* приватный конструктор (почему приватный);
* поясните жизненный цикл ***Singleton****-*класса на тестовом примере.

1. Приведите примеры приложений, в которых может быть применен паттерн ***Singleton****.*
2. ***Поясните суть паттерна Singleton, что дает его применение?***

*Паттерн Singleton используется для обеспечения того, что у класса есть только один экземпляр, и предоставляет глобальную точку доступа к этому экземпляру. Это достигается путем скрытия конструктора класса и предоставления статического метода для доступа к единственному экземпляру. Этот подход полезен, когда требуется, чтобы у класса был только один объект в системе, например, для доступа к общим ресурсам или для предотвращения конфликтов при работе с глобальными данными.*

1. ***Поясните реализацию класса Logger:***
   * ***Статический метод create (почему статический):*** *Статический метод* ***create*** *используется для получения экземпляра класса Logger. Он статический, чтобы можно было вызывать его без необходимости создавать объект класса Logger. Это соответствует идее Singleton, так как позволяет получить доступ к единственному экземпляру класса.*
   * ***Приватный конструктор (почему приватный):*** *Конструктор класса Logger объявлен как приватный, чтобы предотвратить создание экземпляров класса извне. Это гарантирует, что единственный экземпляр Logger может быть создан только внутри класса.*
   * ***Поясните жизненный цикл Singleton-класса на тестовом примере:*** *В данном случае, метод* ***create()*** *каждый раз возвращает один и тот же экземпляр Logger, если он уже был создан. Это обеспечивает, что Logger всегда работает с одним и тем же экземпляром в рамках всего приложения. Это видно из тестового примера в* ***Main****, где несколько объектов* ***ILogger*** *создаются с использованием метода* ***Logger.create()****, но все они указывают на один и тот же экземпляр Logger.*
2. ***Приведите примеры приложений, в которых может быть применен паттерн Singleton:***
   * ***Класс для работы с базой данных:*** *Например, класс, который управляет соединением с базой данных. Есть смысл иметь только одно активное соединение к базе данных, и Singleton гарантирует это.*
   * ***Менеджер настроек приложения:*** *Если ваше приложение использует глобальные настройки (например, конфигурационный файл), класс Singleton может быть использован для предоставления доступа к этим настройкам из любой точки программы.*
   * ***Логгер:*** *Как в примере, класс Logger, который записывает журналы или сообщения в файл или базу данных, может быть реализован как Singleton, чтобы гарантировать, что все сообщения записываются в один и тот же журнал.*