Реализация паттерна Decorator

Павлов Даниил

4п3

<https://github.com/danpaf4g/DecoratorPatternExample>

Данный проект демонстрирует использование структурного паттерна проектирования Decorator . Этот паттерн позволяет динамически добавлять объектам новое поведение, оборачивая их в декораторы. В проекте реализована система для работы с данными, где базовый компонент записывает и читает данные из файла, а декораторы добавляют дополнительные функции, такие как шифрование и логирование.

**Основные компоненты:**

1. Интерфейс **IDataSource**  
   Описывает базовые методы для записи (**WriteData**) и чтения (**ReadData**) данных.
2. Класс **FileDataSource**  
   Реализует интерфейс **IDataSource** и обеспечивает запись/чтение данных в файл.
3. Абстрактный класс **DataSourceDecorator**  
   Базовый класс для всех декораторов. Содержит ссылку на объект-компонент и делегирует ему вызовы методов.
4. Конкретные декораторы:
   * **EncryptionDecorator** : Шифрует данные при записи и расшифровывает их при чтении.
   * **LoggingDecorator** : Логирует все операции записи и чтения.
5. Пример использования:  
   Создается цепочка декораторов, которая добавляет функциональность к базовому компоненту. В результате программа записывает данные в файл, предварительно зашифровав их и залогировав операцию. При чтении данных они расшифровываются, и операция также логируется.

