Ministerul Educației și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

**Лабораторная работа**

по предмету **«Методы и механизмы проектирования программных продуктов»**

Тема: **Шаблоны проектирования – Структурные (Design Patterns – Structural)**

Студент: Никита Маршагин TI-212 FR

Преподаватель: Андрей Постару

Кишинёв 2025**Цель работы**

Изучить и реализовать три поведенческих шаблона проектирования:

Adapter (Адаптер)

Decorator (Декоратор)

Facade (Фасад)

**Теоретическая часть**

**Observer**

Позволяет объектам с несовместимыми интерфейсами взаимодействовать путём обёртывания одного интерфейса в другой. Используется, когда нужно подключить устаревший модуль к новой системе.

**Strategy**

Позволяет динамически добавлять объектам новое поведение, не изменяя их класс. Особенно полезен для добавления функциональности без создания подклассов.

**Command**

Предоставляет упрощённый интерфейс к сложной подсистеме. Позволяет изолировать клиентский код от внутренней структуры системы.

**Практическая часть**

**Adapter (Адаптер)**

**Описание:** Реализация адаптера, преобразующего метод log() старого интерфейса в print() нового.

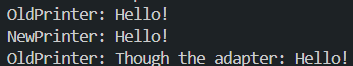
<https://raw.githubusercontent.com/Niki-Mar/TMPP_Labs/refs/heads/main/Lab2/Adapter.js>  
  


Рисунок 1 Вывод работы паттерна Adapter

**Decorator (Декоратор)**

**Описание:** Расширение функциональности объекта печати, добавляя текущую дату и время.

<https://raw.githubusercontent.com/Niki-Mar/TMPP_Labs/refs/heads/main/Lab2/Decorator.js>



Рисунок 2 Вывод работы паттерна Decorator

**Facade (Фасад)**

**Описание:** Упрощённый интерфейс объединяет три внутренних подсистемы и выполняет комплексную операцию одной командой.

<https://raw.githubusercontent.com/Niki-Mar/TMPP_Labs/refs/heads/main/Lab2/Facade.js>



Рисунок 3 Вывод работы паттерна Facade

**Выводы**

Были реализованы три структурных шаблона проектирования: Adapter, Decorator и Facade.  
Показано, как можно адаптировать устаревшие интерфейсы, расширять поведение объектов и скрывать сложность системы от клиента.  
Работа выполнена в среде JavaScript, все шаблоны протестированы и продемонстрированы на практике.