Чернівецький національний університет імені Юрія Федьковича

Факультет математики та інформатики

Кафедра математичного моделювання

**Лабораторна робота №4**

***«Застосування патернів програмування»***

Виконав: студент 3 курсу

301 групи

спеціальності “Комп’ютерні науки”

Козуб Назар

Викладач: Піддубна Л.А

**Чернівці – 2025**

### **1. Патерн "Одинак"**

**Застосування:**  
Клас **Автовокзал.**

**Обґрунтування:**  
Система оперує єдиним центральним об’єктом “Автовокзал”, який координує інші компоненти (каси, розклади, платформи). Для уникнення дублювання екземплярів та централізованого керування доцільно використати патерн Singleton.

### **2. Патерн "Абстрактна фабрика"**

**Застосування:**  
Класи **Квиток, Замовлення, ДодатковаПослуга.**

**Обґрунтування:**  
Існує кілька типів квитків (повний, зі знижкою, з додатковим місцем), які мають різну логіку створення. Для інкапсуляції цієї логіки — фабричний метод або окрема фабрика.

**Переваги:**

* Зменшення залежності від конкретних класів.
* Легка підтримка нових типів без зміни клієнтського коду.

### 3. **Патерн "Стратегія"**

**Застосування:**  
Клас **Знижка.**

**Обґрунтування:**  
Можливість реалізувати кілька незалежних стратегій обчислення знижки — студентська, пенсійна, акційна. Застосування патерну дозволяє динамічно обирати алгоритм без зміни класу користувача.

### 4. **Патерн "Декоратор"**

**Застосування:**  
Клас **Квиток +** додаткові послуги (додаткове місце, багаж, харчування).

**Обґрунтування:**  
Додаткові сервіси повинні мати змогу динамічно додаватися до основного квитка. Декоратор дозволяє створювати нову функціональність без зміни базового класу.

### 5. **Патерн "Спостерігач" (Observer)**

**Застосування:**  
Класи **Клієнт, СистемаСповіщення, Рейс.**

**Обґрунтування:**  
Зміни в розкладі рейсів повинні бути повідомлені всім зацікавленим користувачам. Це класична ситуація для реалізації Observer — клієнти підписуються на об’єкти і отримують оновлення.

### 6. **Патерн "Стан"**

**Застосування:**  
Клас **Квиток.**

**Обґрунтування:**  
Квиток може перебувати в кількох станах: Зарезервований, Оплачений, Скасований, Активований. Патерн дозволяє винести логіку поведінки залежно від стану в окремі класи.

**Посилання на репозиторій:**

[**https://github.com/nazarkozub/Software-systems-design**](https://github.com/nazarkozub/Software-systems-design)