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”

Мацалковський Владислав

Викладач: Піддубна Л.А

**Чернівці – 2025**

***Завдання***

*Аналізуючи діаграми класів (Лабораторна робота №3), обґрунтувати застосування підібраних для реалізації патернів.*

**№ 1**

🎯 **Назва патерна:** Singleton  
🔎 **Призначення:** Забезпечення єдиного екземпляра класу UserRole для керування ролями в системі.  
👥 **Класи-учасники:** UserRole  
⚙ **Механізм:** Клас UserRole може реалізовувати шаблон Singleton через приватний конструктор і статичний метод getInstance(), який повертає єдиний екземпляр. Методи addPermission() та removePermission() дають змогу змінювати набір дозволів.  
📌 **Причини вибору:** Ролі користувачів мають бути єдині в системі (наприклад, Admin, Pharmacist, Client), і централізоване управління цими ролями забезпечить консистентність та контроль доступу.

**№ 2**

🎯 **Назва патерна:** Strategy  
🔎 **Призначення:** Надання можливості змінювати логіку розрахунку ціни ліків на основі умов (наприклад, кількості, терміну придатності тощо).  
👥 **Класи-учасники:** drugs, методи getPrice(), updatePrice()  
⚙ **Механізм:** Метод getPrice() у класі drugs може реалізовуватись різними стратегіями (базова, зі знижкою, за категорією, з урахуванням строку придатності). Стратегії можуть інкапсулюватися у вигляді окремих класів, що передаються до об’єкта.  
📌 **Причини вибору:** Спрощує підтримку та розширення логіки ціноутворення без зміни коду основного класу drugs.

**№ 3**

🎯 **Назва патерна:** Command  
🔎 **Призначення:** Інкапсуляція дій користувача під час обробки замовлення.  
👥 **Класи-учасники:** Order, методи addItem(), removeItem(), calculateTotal(), processPayment()  
⚙ **Механізм:** Кожна дія з об’єктом Order (додати товар, видалити, розрахувати вартість, обробити оплату) може бути оформлена як окрема команда. Наприклад, AddItemCommand, RemoveItemCommand, які виконуються централізовано з OrderManager.  
📌 **Причини вибору:** Дозволяє легко реалізувати історію дій (undo/redo), а також централізовано керувати сценаріями замовлення.

**№ 4**

🎯 **Назва патерна:** Observer  
🔎 **Призначення:** Сповіщення користувача або ролі при зміні даних замовлення.  
👥 **Класи-учасники:** Order, User, метод updateProfile(), getOrderHistory()  
⚙ **Механізм:** Order виступає як Subject і зберігає список підписників (наприклад, користувачів). При зміні статусу замовлення викликаються методи підписаних об'єктів (наприклад, оновлення профілю користувача, історії замовлень).  
📌 **Причини вибору:** Автоматизація оновлень інформації про замовлення у користувача без прямого зв’язку між об'єктами.

***Завдання***

Використовуючи лабораторні роботи із дисциплін професійної підготовки, створити репозиторій, розмістити файли проектів. Надати доступ викладачам на github / gitlab ресурсі.

*Посилання: https://github.com/DeFrag73/Software-systems-design-Lab\_4*