

## Лабораторна робота №4

### Застосування патернів програмування

#### Завдання

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

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

**Патерн "Сутності"** використовується для моделювання основних об'єктів системи, таких як "Пацієнт", "Адміністратор", "Лікар", "Запис", "Відгук", "Медична карта", "Рахунок", "Спеціальність" та "Статистика". Кожна сутність має свої атрибути та методи, які описують її характеристики та поведінку.

**Патерн "Спадщина"** використовується для моделювання ієрархії між сутностями. Наприклад, сутність "Лікар" успадковує атрибути та методи від сутності "Користувач". Це дозволяє уникнути дублювання коду, коли різні типи лікарів (терапевт, хірург, педіатр) можуть мати специфічні атрибути та методи.

**Патерн "Фабрика"** використовується для створення екземплярів сутностей. Наприклад, фабрика "Запис" може створювати записи різних типів (огляд, консультація, обстеження). Це дозволяє динамічно створювати об'єкти без необхідності знати їх конкретні типи.

**Патерн "Спостерігач"** використовується для реалізації системи повідомлень між сутностями. Наприклад, коли пацієнт змінює свою інформацію, система може надіслати повідомлення лікарям.

**Патерн "Синглтон"** використовується для створення єдиного екземпляра сутності. Наприклад, клас "Система" може бути синглтоном, щоб

гарантувати, що в системі є лише один екземпляр даних та логіки. Це корисно для сутностей, які мають бути глобально доступними.

**Патерн "Стратегія"** використовується для реалізації різних алгоритмів або поведінки для однієї й тієї ж операції. Наприклад, система може використовувати різні стратегії оплати (готівка, кредитна картка, переказ). Це дозволяє динамічно змінювати поведінку системи без необхідності змінювати її код.

**Патерн "Команда"** використовується для реалізації групи об'єктів, які працюють разом для досягнення спільної мети. Наприклад, команда "Обробка запису" може складатися з об'єктів "Створення запису", "Перевірка запису" та "Підтвердження запису". Це дозволяє розбити складні завдання на дрібніші кроки та розподілити їх між різними об'єктами.

**Патерн "Декоратор"** використовується для динамічного додавання нових функціональних можливостей до об'єкта. Наприклад, декоратор може додавати до запису функцію надсилення SMS-повідомлення пацієнту про нагадування про прийом.

**Патерн "Стан"** використовується для моделювання поведінки об'єкта, яка залежить від його поточного стану. Наприклад, запис може мати такі стани: "Заплановано", "Скасовано", "Завершено". Це дозволяє чітко описувати стан об'єкта та його поведінку в кожному стані.

**Патерн "Відвідувач"** використовується для реалізації операцій, які виконуються над множиною об'єктів. Наприклад, візитор "Звіт про прийоми" може використовуватися для генерування звіту про кількість прийомів для кожного лікаря. Це дозволяє розділити логіку обробки даних від самих даних.