Лабораторна робота №4:

Використання патернів програмування

Завдання

1. На основі діаграм класів з Лабораторної роботи №3 обґрунтувати застосування обраних патернів для реалізації системи.

2. Використовуючи матеріали з професійних дисциплін, створити репозиторій з файлами проектів і надати доступ викладачам на ресурсі GitHub або GitLab.

Патерни

Патерн "Сутності"

Цей патерн застосовується для моделювання основних об'єктів системи: "Пацієнт", "Адміністратор", "Ветеринарний лікар", "Запис", "Відгук", "Медична карта", "Рахунок", "Спеціальність" і "Статистика". Кожна сутність має свої атрибути та методи, що описують її властивості та поведінку.

Патерн "Спадщина"

Для моделювання ієрархії між сутностями використовується патерн "Спадщина". Наприклад, "Ветеринарний лікар" успадковує атрибути та методи від "Користувача". Це дозволяє уникнути дублювання коду і забезпечує гнучкість при додаванні нових типів лікарів (терапевт, хірург, педіатр).

Патерн "Фабрика"

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

Патерн "Спостерігач"

Патерн "Спостерігач" використовується для реалізації системи повідомлень між сутностями. Наприклад, при зміні інформації пацієнта система може повідомити про це ветеринарному лікарю.

Патерн "Синглтон"

Патерн "Синглтон" забезпечує створення єдиного екземпляра сутності. Наприклад, клас "Система" може бути реалізований як синглтон, щоб гарантувати наявність тільки одного екземпляра даних та логіки, доступного глобально.

Патерн "Стратегія"

Патерн "Стратегія" дозволяє реалізовувати різні алгоритми або поведінку для однієї й тієї ж операції. Наприклад, система може використовувати різні стратегії оплати (готівка, кредитна картка, переказ), що дозволяє динамічно змінювати поведінку системи без зміни її коду.

Патерн "Команда"

Патерн "Команда" використовується для організації 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 менші кроки та розподілити їх між різними об'єктами.

Патерн "Декоратор"

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

Патерн "Стан"

Патерн "Стан" застосовується для моделювання поведінки об'єкта в залежності від його поточного стану. Наприклад, запис може мати стани: "Заплановано", "Скасовано", "Завершено". Це дозволяє чітко описувати стан об'єкта та його поведінку в кожному з них.

Патерн "Відвідувач"

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 відокремити логіку обробки даних від самих даних.