

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

1. Патерн "Factory Method" (Фабричний метод)

На автовокзалі можна використати для створення різних типів квитків (наприклад, одноразовий квиток, місячний абонимент на певний маршрут) може використовуватися патерн "Factory Method". Для створення конкретних типів квитків можна створити фабричний метод, що визначає інтерфейс для створення об'єктів у суперкласі, але дозволяє підкласам змінювати тип створюваних об'єктів.

2. Патерн "Singleton" (Одинак)

Для забезпечення єдиного доступу до бази даних протягом усього часу роботи системи, реалізуємо як одинак, щоб гарантувати, що доступ до бази даних здійснюється через єдиний екземпляр цього класу. Це дозволяє уникнути конфліктів і забезпечити ефективну роботу з даними.

3. Патерн "Observer" (Спостерігач)

У нашій системі цей патерн може бути корисним для оновлення інформації в інтерфейсі користувача у реальному часі при зміні стану замовлення. У класі "Замовлення" може бути об'єктом, за змінами якого стежать інші об'єкти. У класі "Інтерфейс" можемо реалізувати спостерігачів, які будуть оновлювати відображення інформації при зміні стану замовлення.

4. Патерн "Strategy" (Стратегія)

Для реалізації різних методів оплати (кредитна картка, Apple Pay, Google Pay, банківський переказ тощо). Можемо використовувати різні стратегії для обробки платежів. Це дозволяє легко додавати нові способи оплати без зміни існуючого коду.

5. Патерн "Decorator" (Декоратор)

Для додавання додаткових властивостей або поведінки до квитків, наприклад страховка поїзди від компанії, яка надає послуги.