

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Навчально-науковий інститут комп'ютерного моделювання,
прикладної фізики та математики

Звіт

З лабораторної роботи №7

З дисципліни:“ Ефективність та якість архітектурних рішень
інформаційних систем ”

На тему:

«Патерн проєктування Стратегія»

Виконав:

студент групи ІКМ-М225В

Загорулько Віталій Олегович

Харків 2025

Мета роботи

Метою даної лабораторної роботи є ґрунтовне вивчення та практичне засвоєння поведінкового патерна проєктування Стратегія (Strategy), а також набуття навичок його застосування для визначення сімейства алгоритмів (стратегій), інкапсулювання кожного з них та забезпечення їхньої взаємозамінності.

Завдання

У межах виконання лабораторної роботи необхідно виконати такі завдання:

1. Ознайомитися з теоретичними зasadами та основними ідеями патерна проєктування Стратегія.
2. Створити структуру класів, яка демонструє реалізацію патерна Стратегія на прикладі розрахунку вартості доставки в додатку доставки їжі.
3. Кожен спосіб доставки має бути представлений як окрема стратегія:
 - Самовивіз (SelfPickupStrategy).
 - Доставка зовнішньою службою (ExternalDeliveryStrategy).
 - Доставка власною службою (OwnDeliveryStrategy).
4. Створити структуру класів та методів, що вирішує описане завдання, та приклад клієнтського коду, що демонструє вибір стратегії.

Теоретичні відомості та Опис Проєкту

Патерн Стратегія (Strategy) — це поведінковий патерн проєктування, який визначає сімейство взаємозамінних алгоритмів, інкапсулює кожен із них і дозволяє обирати потрібний алгоритм безпосередньо під час виконання програми . Патерн дозволяє клієнту змінювати поведінку об'єкта (Контексту) шляхом заміни його внутрішнього об'єкта-стратегії. Основна перевага цього патерна полягає в тому, що він дозволяє уникнути умовних операторів (if/else або switch) у класі, де має виконуватися дія, та забезпечує принцип відкритості/закритості (OCP) — система відкрита для розширення новими стратегіями, але закрита для модифікації.

У межах даної лабораторної роботи було реалізовано систему розрахунку вартості доставки для умовного додатка доставки їжі. Для цього було створено три ключові компоненти патерна Стратегія:

- Strategy (Інтерфейс Стратегії): Інтерфейс DeliveryCostCalculator визначає єдиний метод calculateCost(double orderTotal, double distanceKm), який повинні реалізовувати всі стратегії розрахунку.
- Concrete Strategies (Конкретні Стратегії): Було створено три класи: SelfPickupStrategy (вартість завжди 0), ExternalDeliveryStrategy (розрахунок на основі фіксованої плати та комісії) та OwnDeliveryStrategy (розрахунок на основі відстані та мінімальної плати). Кожна стратегія інкапсулює свій унікальний алгоритм.
- Context (Контекст): Клас DeliveryApp містить посилання на об'єкт DeliveryCostCalculator і делегує йому виконання методу getDeliveryCost(). Контекст не знає, яка саме стратегія використовується, і може бути налаштований клієнтом динамічно під час виконання програми.

Клієнтський код демонструє, як додаток спочатку ініціалізується однією стратегією (наприклад, OwnDeliveryStrategy), а потім легко перемикається на іншу (ExternalDeliveryStrategy або SelfPickupStrategy) за допомогою методу setDeliveryStrategy(), забезпечуючи гнучкий та чистий код.

Програмний код реалізації на Java розміщено у Додатку А.

Висновок

Під час виконання лабораторної роботи було успішно вивчено та реалізовано поведінковий патерн Стратегія. Завдання розрахунку вартості доставки для різних методів було вирішено шляхом інкапсуляції кожного методу в окремий клас-стратегію. Клас DeliveryApp (Контекст) отримав можливість динамічно змінювати алгоритм розрахунку вартості доставки, просто замінюючи об'єкт стратегії, на який він посилається. Це дозволило уникнути умовних операторів для вибору алгоритму, підвищило гнучкість системи та забезпечило високу ступінь взаємозамінності алгоритмів. Отримані знання підтверджують ефективність патерна Стратегія для проєктування систем, де поведінка об'єкта повинна бути змінною та легко адаптованою до нових вимог.

Додаток А