

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Звіт

з лабораторної роботи №7

з дисципліни «Ефективність та якість архітектурних рішень  
інформаційних систем»

Виконав:

студент групи ІКМ-М225В

Суліма Нікіта Володимирович

Перевірив:

аспірант каф. КМПС Хорошун Андрій Сергійович

Харків 2025

Створено інтерфейс `DeliveryStrategy`. Цей інтерфейс визначає спільні методи для всіх стратегій:

```
virtual double calculateCost(double distance, double weight) = 0;  
virtual string getName() = 0;
```

Він описує структуру для розрахунку вартості доставки, яку реалізують конкретні стратегії.

Реалізовано три конкретні стратегії:

- `SelfPickupStrategy` — стратегія самовивозу, доставка безкоштовна.
- `ExternalDeliveryStrategy` — стратегія доставки зовнішньою службою (вартість залежить від відстані та ваги).
- `InternalDeliveryStrategy` — стратегія доставки власною службою.

Контекст `DeliveryContext`

Клас, який зберігає поточну стратегію і викликає її метод:

```
void setStrategy(DeliveryStrategy* s);  
void calculate(double distance, double weight);
```

Контекст дозволяє динамічно змінювати стратегію під час виконання програми.

Клієнтський код

У функції `main()` створюються три стратегії доставки, які по черзі передаються у контекст:

```
context.setStrategy(&selfPickup);  
context.calculate(10, 5);
```

Це демонструє зміну алгоритму без зміни структури самого класу `DeliveryContext`.

## Висновок

У ході лабораторної роботи реалізовано патерн проєктування «Стратегія», який дозволяє змінювати поведінку програми (алгоритм розрахунку вартості доставки) без зміни її структури.