



Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

### **Лабораторна робота №8**

З дисципліни “Технології розроблення програмного забезпечення”

Тема: “ ШАБЛони «COMPOSITE», «FLYWEIGHT», «INTERPRETER»,  
«VISITOR»”

Виконав

студент групи ІА–22:

Прохоров О.Д.

Перевірів

Мягкий М. Ю.

Київ 2024

## Зміст

Короткі теоретичні відомості: .....	3
Крок 1. Код шаблону Composite. ....	3
Крок 2. Основний клас системи. ....	5
Крок 3. Результати програми.....	5
Висновки. ....	5

**Тема:** ШАБЛОНИ «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR»

**Мета:** Навчитися використанню одного з зазначених шаблонів проектування.

### **Хід роботи:**

#### **Короткі теоретичні відомості:**

Шаблон Composite використовується для складання об'єктів в деревоподібну структуру для подання ієрархій типу «частина цілого». Даний шаблон дозволяє уніфіковано обробляти як поодинокі об'єкти, так і об'єкти з вкладеністю.

Простим прикладом може служити складання компонентів всередині звичайної форми. Форма може містити дочірні елементи (поля для введення тексту, цифр, написи, малюнки тощо); дочірні елементи можуть в свою чергу містити інші елементи.

Даний шаблон зручно використовувати при необхідності подання та обробки ієрархій об'єктів.

#### **Крок 1. Код шаблону Composite.**

Для виконання цієї лабораторної роботи були написані нові та змінені наступні класи:

ShoppingComponent.java, Goods.java, ShoppingList.java та змінено код головного файлу проекту. На наступних рисунках(1.1, 1.2, 1.3, 2.1) відображено код цих класів.

```
11 usages 2 implementations
public interface ShoppingComponent {
    1 usage 2 implementations
    double getTotalPrice();
    3 usages 2 implementations
    void display();
}
```

*Рисунок 1.1 ShoppingComponent.java*

```

35 usages
public class Goods implements ShoppingComponent {
    4 usages
    private Long Id;
    5 usages
    private String name;
    4 usages
    private String description;
    4 usages
    private double estPrice;
    6 usages
    private double actPrice;
    4 usages
    private Receipt receipt;

    1 usage
    @Override
    public double getTotalPrice() { return actPrice; }

    3 usages
    @Override
    public void display() { System.out.println("Goods: " + name + ", Price: " + actPrice); }

```

### *Рисунок 1.2 Goods.java*

```

9 usages
public class ShoppingList implements ShoppingComponent {
    2 usages
    private Long Id;
    3 usages
    private String name;
    5 usages
    private List<ShoppingComponent> components = new ArrayList<>();

    3 usages
    public ShoppingList(Long Id, String name) {
        this.Id = Id;
        this.name = name;
    }

    1 usage
    @Override
    public double getTotalPrice() { return components.stream().mapToDouble(ShoppingComponent::getTotalPrice).sum(); }

    3 usages
    @Override
    public void display() {
        System.out.println("List: " + name);
        for (ShoppingComponent component : components) {
            component.display();
        }
    }
}

```

### *Рисунок 1.3 ShoppingList.java*

## Крок 2. Основний клас системи.

```
@SpringBootApplication
public class CpsProjectConsoleApplication {
    public static void main(String[] args) {
        SpringApplication.run(CpsProjectConsoleApplication.class, args);
        SystemFacade systemFacade = new SystemFacade();

        systemFacade.attachGoodToList( shoppingListId: 1L, goodId: 1L);
        systemFacade.attachGoodToList( shoppingListId: 2L, goodId: 3L);
        systemFacade.attachGoodToList( shoppingListId: 2L, goodId: 5L);

        ShoppingList allShopping = new ShoppingList( Id: 3L, name: "All Shopping");
        systemFacade.getLists().forEach(allShopping::addComponent);

        System.out.println("---- All Shopping List ----");
        allShopping.display();
    }
}
```

*Рисунок 2.1 CpsProjectConsoleApplication.java*

## Крок 3. Результати програми

На рисунку 3.1 можна побачити, що програма успішно скомпільовала код та видала необхідний результат, яким є:

- вивід додавання товарів у список
- вивід помилки, якщо товар не знайдено
- виведення всіх списків товарів

```
Good 'Laptop' added to Shopping List with ID 1
Good 'Apple' added to Shopping List with ID 2
Good with ID 5 not found in the system.
---- All Shopping List ----
List: All Shopping
List: Electronics
List: Groceries
```

*Рисунок 3.1 Results*

## Висновки.

В результаті цієї лабораторної роботи було виконано дії з вивчення зазначених патернів проектування, в результаті якого я визначив, що для нашої системи найбільше підходить саме патерн Composite, створення коду, що відповідає шаблону проектування Composite та базового коду проекту. Результати програми було виведено через консоль для демонстрації роботи програми.