



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 №5

З дисципліни “Технології розроблення програмного забезпечення”

Тема: “ ШАБЛони «ADAPTER», «BUILDER», «COMMAND», «CHAIN
OF RESPONSIBILITY», «PROTOTYPE»”

Виконав

студент групи ІА–22:

Прохоров О.Д.

Перевірів

Мягкий М. Ю.

Київ 2024

Зміст

Короткі теоретичні відомості:	3
Крок 1. Код шаблону Builder.....	4
Крок 2. Основний клас системи.	5
Крок 3. Результати програми.....	5
Висновки.	5

Тема: ШАБЛОНИ «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF RESPONSIBILITY», «PROTOTYPE»

Мета: Навчитися використанню одного з зазначених шаблонів проектування.

Хід роботи:

Короткі теоретичні відомості:

Будь-який патерн проектування, використовуваний при розробці інформаційних систем, являє собою формалізований опис, який часто зустрічається в завданнях проектування, вдале рішення даної задачі, а також рекомендації по застосуванню цього рішення в різних ситуаціях. Крім того, патерн проектування обов'язково має загальнозастосовуване найменування. Правильно сформульований патерн проектування дозволяє, відшукавши одного разу вдале рішення, користуватися ним знову і знову.

Застосування шаблонів проектування не гарантує, що розроблена архітектура буде кристально чистою і зручною з точки зору програмування. Однак в потрібних місцях застосування шаблонів дозволить досягти наступних вигод:

- Зменшення трудовитрат і часу на побудову архітектури;
- Надання проєктованій системі необхідних якостей (гнучкість, адаптованість, ін.);
- Зменшити накладні витрати на подальшу підтримку системи;
- Та інші.

Шаблон "builder" (будівельник) використовується для відділення процесу створення об'єкту від його представлення. Це доречно у випадках, коли об'єкт має складний процес створення (наприклад, Web- сторінка як елемент повної відповіді web- сервера) або коли об'єкт повинен мати декілька різних форм створення (наприклад, при конвертації тексту з формату у формат)

Крок 1. Код шаблону Builder.

```
5 usages
public static class Builder {
    2 usages
    private String groupName;
    2 usages
    private List<User> participants = new ArrayList<>();
    2 usages
    private List<Goods> goodsList = new ArrayList<>();
    2 usages
    private double estimatedTotalCost = 0.0;

    1 usage
    public Builder withGroupName(String groupName) {
        this.groupName = groupName;
        return this;
    }

    2 usages
    public Builder addParticipant(User user) {
        this.participants.add(user);
        return this;
    }

    2 usages
    public Builder addGoods(Goods goods) {
        this.goodsList.add(goods);
        this.estimatedTotalCost += goods.getEstPrice();
        return this;
    }

    1 usage
    public UserGroup build() { return new UserGroup( builder: this); }
}
```

UserGroup.java

Крок 2. Основний клас системи.

```
@SpringBootApplication
public class CpsProjectConsoleApplication {
    public static void main(String[] args) {
        SpringApplication.run(CpsProjectConsoleApplication.class, args);
        User user1 = new User( id: 1L, name: "Alice", email: "alice@example.com");
        User user2 = new User( id: 2L, name: "Bob", email: "bob@example.com");

        Goods cake = new Goods( id: 1L, name: "Cake", description: "Birthday cake", estPrice: 30.0, actPrice: 30.0);
        Goods candles = new Goods( id: 2L, name: "Candles", description: "Pack of candles", estPrice: 5.0, actPrice: 5.0);

        UserGroup birthdayGroup = new UserGroup.Builder()
            .withGroupName("Birthday Party")
            .addParticipant(user1)
            .addParticipant(user2)
            .addGoods(cake)
            .addGoods(candles)
            .build();

        System.out.println("Participants: " + birthdayGroup.getParticipants());
        System.out.println("Goods: " + birthdayGroup.getGoodsList());
    }
}
```

CpsProjectConsoleApplication.java

Крок 3. Результати програми

```
Participants: [User{id=1, name='Alice', email='alice@example.com', shoppingList=[]}, User{id=2, name='Bob', email='bob@example.com', shoppingList=[]}]
Goods: [Goods{id=1, name='Cake', desc='Birthday cake', Estimated price=30.0, Actual price=5.0}, Goods{id=2, name='Candles', desc='Pack of candles', Estimated price=5.0, Actual price=5.0}]
```

Results

Висновки.

В результаті цієї лабораторної роботи було виконано дії з створення коду, що відповідає шаблону проектування Builder та базового коду проекту.