Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №8

Технологія розроблення програмного забезпечення

ШАБЛОНИ «COMPOSITE», «FLYWEIGHT», «INTERPRETER», «VISITOR»

Варіант 19

Виконав
студент групи ІА-13
Павлюк Оскар Ігорович

Перевірив:
Мягкий М. Ю.

Київ 2023р.

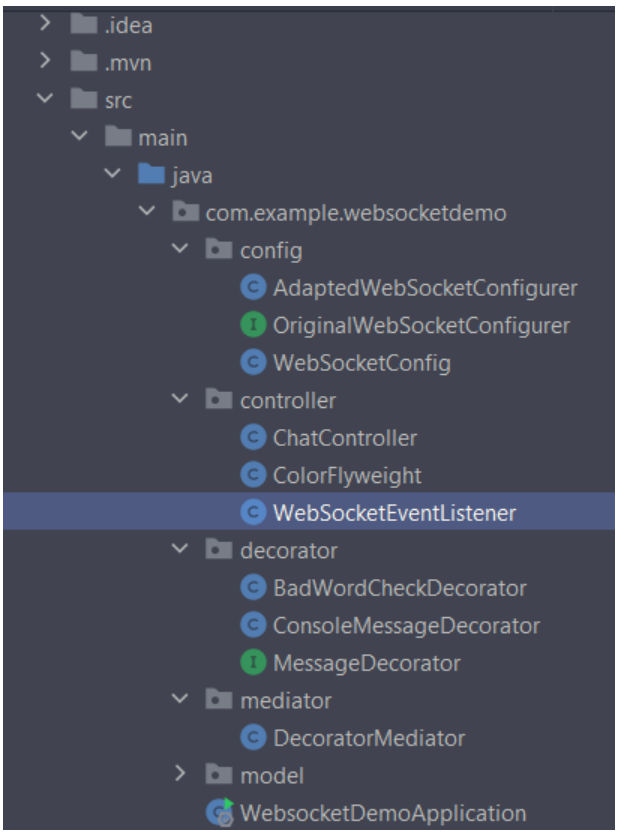
Мета: 1. Ознайомитися з короткими теоретичними відомостями 2. Реалізувати частину функціонала робочої програми у вигляді класів і їх взаємодій для досягнення конкретних функціональних можливостей. 3. Застосування одного з даних шаблонів при реалізації програми

Хід роботи

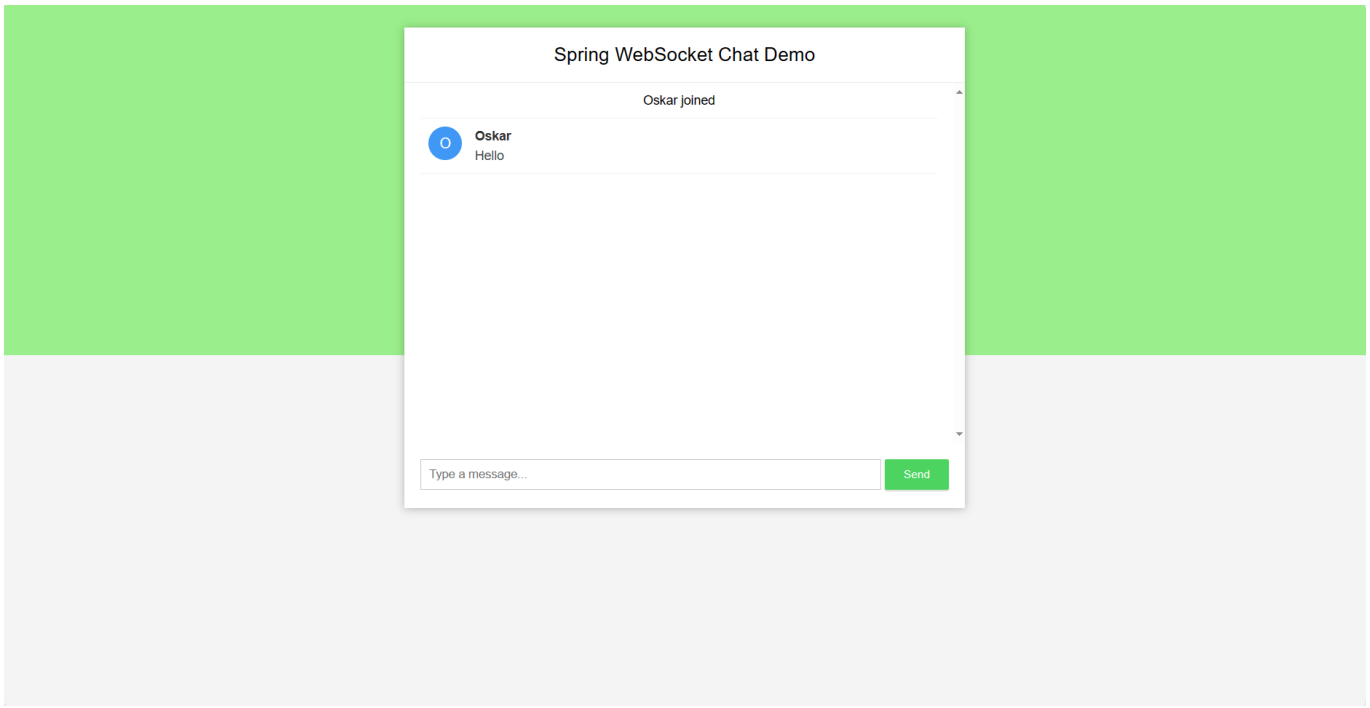
..19 IRC client (singleton, builder, abstract factory, template method, composite, client-server)

Клієнт для IRC-чатів з можливістю вказівки порту і адреси з'єднання, підтримка базових команд (підключення до чату, створення чату, установка імені, реєстрація, допомога і т.д.), отримання метаданих про канал.

Структура проекту:



Вигляд роботи:



Реалізація шаблону “Flyweight”

Реалізація патерну "Flyweight" загалом використовується для зменшення об'єму пам'яті, який займається об'єктами, що повторюються. У моєму випадку це відноситься до кольорів, які використовуються для аватарів користувачів.

Клас ColorFlyweight представляє кольори і дозволяє економно використовувати пам'ять, зберігаючи лише унікальні кольори в пулі.

```
package com.example.websocketdemo.controller;

public class ColorFlyweight {
    public String color;

    public ColorFlyweight(String color) {
        this.color = color;
    }

    public String getColor() {
        return color;
    }
}
```

Використання пула для кольорів:

При формуванні аватара для користувача ми використовуємо метод getAvatarColor, який використовує пул кольорів для забезпечення використання одного і того ж кольору для кількох користувачів з однаковими характеристиками.

Зменшення обсягу пам'яті:

Оскільки кожен об'єкт ColorFlyweight використовується повторно, замість того, щоб створювати новий об'єкт для кожного користувача, це дозволяє зменшити обсяг пам'яті, яку використовує програма.

```
private String getAvatarColor(String messageSender) {
        int hash = 0;

        if (messageSender != null) {
            for (int i = 0; i < messageSender.length(); i++) {
                hash = 31 * hash + messageSender.charAt(i);
            }
        }

        String[] colors = new String[0];
        int index = Math.abs(hash % colors.length);

        // Перевірте, чи колір вже є в пулі
        if (!colorPool.containsKey(index)) {
            colorPool.put(index, new ColorFlyweight(colors[index]));
        }

        return colorPool.get(index).getColor();
    }
}
```

Паттерн "Flyweight" особливо ефективний, коли у нас є велика кількість об'єктів, які мають спільні характеристики.

Висновок: Під час цієї лабораторної роботи було розроблено частково реалізацію IRC client і також патерн Flyweight.