****

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 №4  
**Технологія розроблення програмного забезпечення**

ШАБЛОНИ «SINGLETON», «ITERATOR», «PROXY», «STATE», «STRATEGY»

Варіант 19

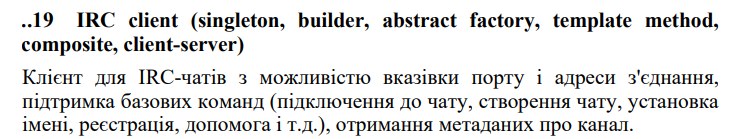
Виконав Перевірив:

студент групи ІА-13 Драган Михайло

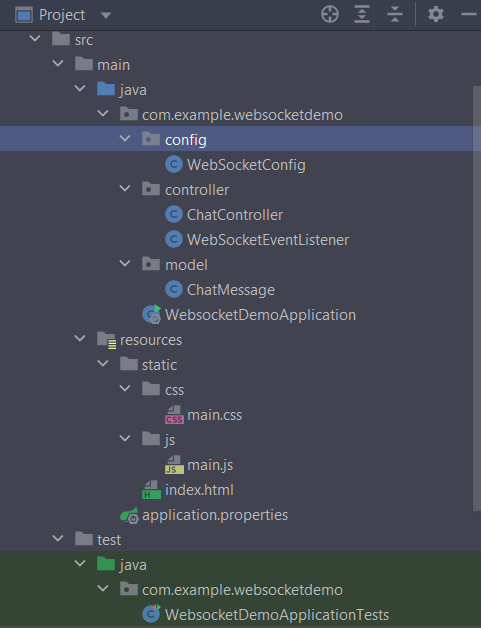
Павлюк Оскар Ігорович Сергійович

Київ 2023р.

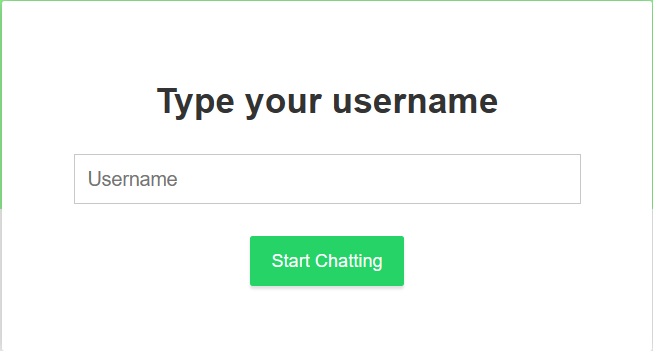
**Мета:** Ознайомитися з короткими теоретичними відомостями. 2. Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей. 3. Застосування одного з розглянутих шаблонів при реалізації програми.

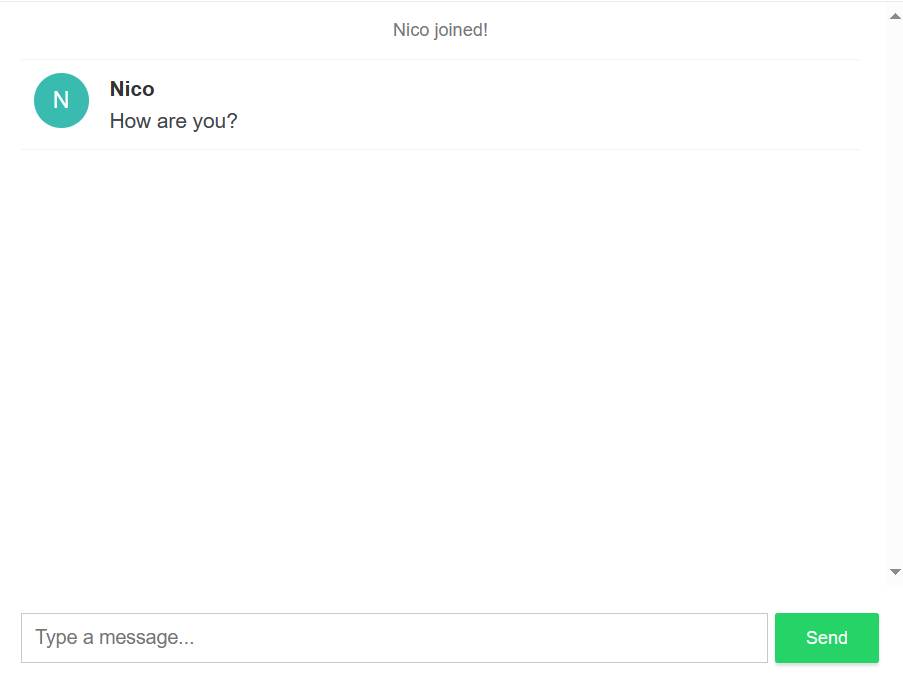
**Хід роботи**

**Структура проекту:**



**Вигляд роботи:**





**Опис класів:**

1. Клас **WebSocketConfig** є конфігураційним класом для налаштування веб-сокетів у застосунку, який використовує Spring Framework.
2. Клас **ChatController** відповідає за обробку повідомлень, які приходять від клієнтів через веб-сокети.
3. Клас **WebSocketEventListener** відповідає за обробку подій, пов'язаних із з'єднанням та від'єднанням веб-сокетів.
4. Клас **ChatMessage** служить для представлення повідомлень в чаті.

**Реалізація шаблону “singleton”**

Клас WebSocketConfig використовується для конфігурації WebSocket. Коли ми додаємо анотацію @Scope("singleton") до цього класу, ми вказуєте Spring IoC контейнеру створювати та управляти лише одним екземпляром цього класу протягом всього життєвого циклу застосунку.

**Основні вигоди використання Singleton в цьому контексті:**

1. Оптимізація ресурсів: Оскільки WebSocketConfig відповідає за налаштування WebSocket, і його стан зазвичай не змінюється під час виконання програми, має сенс мати тільки один екземпляр цього класу.
2. Консистентність налаштувань: Якщо би було створено багато екземплярів WebSocketConfig, то кожен з них міг би мати свої власні налаштування, що призвело б до непередбачуваної поведінки вашого WebSocket. Використання Singleton забезпечує консистентність налаштувань для всього застосунку.
3. Ефективність роботи з пам'яттю: Оскільки тільки один екземпляр класу існує в пам'яті, це може допомогти економії пам'яті та зменшенню витрат ресурсів.

package com.example.websocketdemo.config;  
  
import org.springframework.context.annotation.*Configuration*;  
import org.springframework.context.annotation.*Scope*;  
import org.springframework.messaging.simp.config.MessageBrokerRegistry;  
import org.springframework.web.socket.config.annotation.*EnableWebSocketMessageBroker*;  
import org.springframework.web.socket.config.annotation.StompEndpointRegistry;  
import org.springframework.web.socket.config.annotation.WebSocketMessageBrokerConfigurer;  
  
*@Configuration  
@EnableWebSocketMessageBroker  
@Scope*("singleton")  
public class WebSocketConfig implements WebSocketMessageBrokerConfigurer {  
  
 public WebSocketConfig() {  
 }  
  
 *@Override* public void registerStompEndpoints(StompEndpointRegistry *registry*) {  
 *registry*.addEndpoint("/ws").withSockJS();  
 }  
  
 *@Override* public void configureMessageBroker(MessageBrokerRegistry *registry*) {  
 *registry*.setApplicationDestinationPrefixes("/app");  
 *registry*.enableSimpleBroker("/topic");  
 }  
}

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