****

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 №5  
**Технологія розроблення програмного забезпечення**

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

Варіант 19

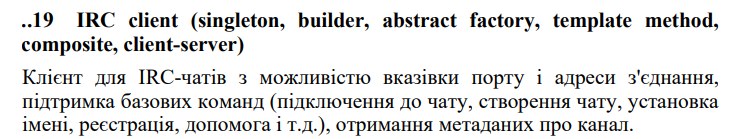
Виконав Перевірив:

студент групи ІА-13 Драган Михайло

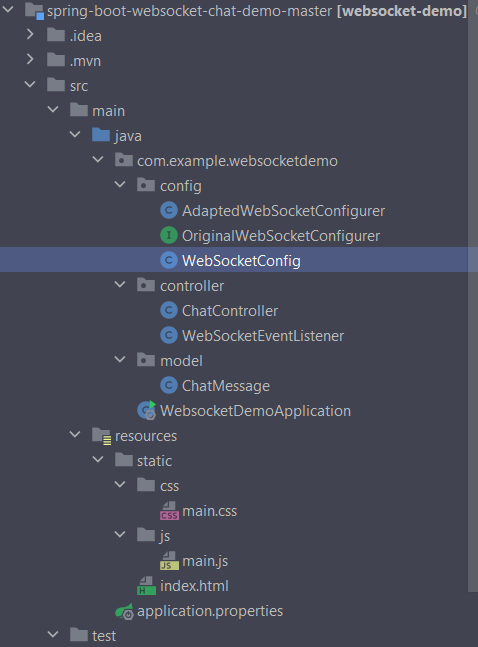
Павлюк Оскар Ігорович Сергійович

Київ 2023р.

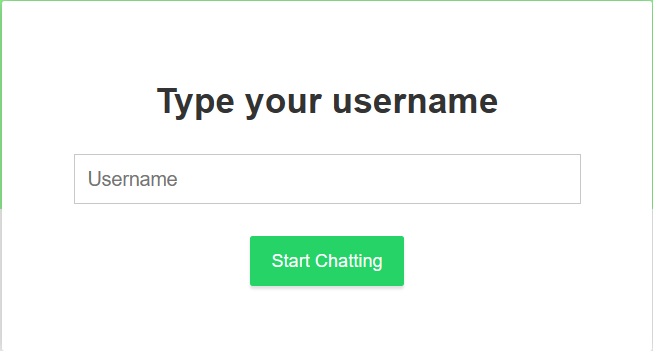
**Мета:** Ознайомитися з короткими теоретичними відомостями. 2. Реалізувати частину функціоналу робочої програми у вигляді класів та їх взаємодій для досягнення конкретних функціональних можливостей. 3. Застосування одного з розглянутих шаблонів при реалізації програми.

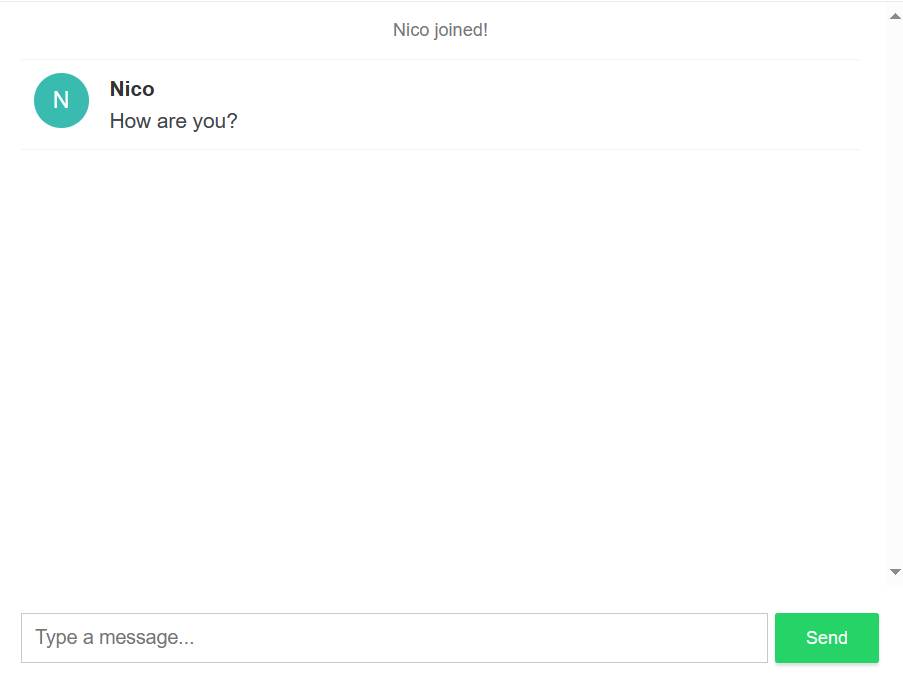
**Хід роботи**

**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 “adapter”**

Адаптер AdaptedWebSocketConfigurer реалізує інтерфейс OriginalWebSocketConfigurer та використовує WebSocketConfig для реалізації методу setupEndpointsAndBroker. Таким чином, ми можемо використовувати AdaptedWebSocketConfigurer там, де очікується OriginalWebSocketConfigurer, і він передасть виклики відповідним методам WebSocketConfig.

AdaptedWebSocketConfigurer слугує адаптером між інтерфейсом OriginalWebSocketConfigurer та класом WebSocketConfig.

package com.example.websocketdemo.config;  
  
import org.springframework.beans.factory.annotation.*Autowired*;  
import org.springframework.context.annotation.*Scope*;  
import org.springframework.messaging.simp.config.MessageBrokerRegistry;  
import org.springframework.stereotype.*Component*;  
import org.springframework.web.socket.config.annotation.StompEndpointRegistry;  
import org.springframework.web.socket.config.annotation.WebSocketMessageBrokerConfigurer;  
  
*@Component  
@Scope*("singleton")  
public class AdaptedWebSocketConfigurer implements WebSocketMessageBrokerConfigurer {  
  
 private final WebSocketConfig webSocketConfig;  
  
 *@Autowired* public AdaptedWebSocketConfigurer(WebSocketConfig *webSocketConfig*) {  
 this.webSocketConfig = *webSocketConfig*;  
 }  
  
 *@Override* public void registerStompEndpoints(StompEndpointRegistry *registry*) {  
 webSocketConfig.registerStompEndpoints(*registry*);  
 }  
  
 *@Override* public void configureMessageBroker(MessageBrokerRegistry *registry*) {  
 webSocketConfig.configureMessageBroker(*registry*);  
 }  
}

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