****

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  
**ШАБЛОНИ «SINGLETON»,**

**«ITERATOR», «PROXY», «STATE»,**

**«STRATEGY»**

Виконав Перевірив:

студент групи ІА – 13: Мягкий М. Ю

Мельничук Дмитро

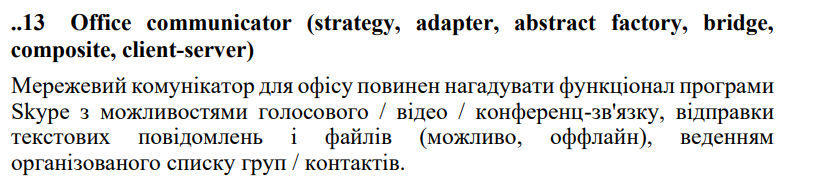
**Завдання:**

1. Ознайомитися з короткими теоретичними відомостями.

2. Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії

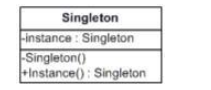
для досягнення конкретних функціональних можливостей.

3. Застосування одного з розглянутих шаблонів при реалізації програми



**Хід роботи**

Шаблон Singleton — це породжуючий шаблон проектування, який гарантує, що для певного класу буде створено тільки один екземпляр, і надає глобальну точку доступу до цього екземпляра. Давайте створимо простий шаблон Singleton для офісного комунікатора. Оскільки це буде елемент програмного забезпечення для мережевого комунікатора, ми можемо також розглянути певні аспекти клієнт-серверної архітектури.



public class OfficeCommunicator {

private static OfficeCommunicator instance;

// Приватний конструктор, щоб заборонити створення екземплярів поза класом

private OfficeCommunicator() {

// Ініціалізація

}

// Глобальна точка доступу

public static OfficeCommunicator getInstance() {

if (instance == null) {

instance = new OfficeCommunicator();

}

return instance;

}

// Додаткові методи та функціонал для комунікатора

public void sendMessage(String message, String recipient) {

// Логіка відправки повідомлення

}

public void initiateVoiceCall(String contact) {

// Логіка початку голосового дзвінка

}

}

public class Main {

public static void main(String[] args) {

// Отримання єдиного екземпляру комунікатора

OfficeCommunicator communicator = OfficeCommunicator.getInstance();

// Використання комунікатора

communicator.sendMessage("Привіт, як справи?", "user1");

communicator.initiateVoiceCall("user2");

}

}