****

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 №9  
РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ:

CLIENT-SERVER, PEER-TO-PEER,

SERVICE-ORIENTED ARCHITECTURE

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 і їх взаємодій для

досягнення конкретних функціональних можливостей.

3. Реалізувати взаємодію програми в одній з архітектур відповідно до обраної теми.

**Хід роботи**

Клієнт-серверні додатки являють собою найпростіший варіант розподілених додатків, де виділяється два види додатків: клієнти (представляють додаток користувачеві) і сервери (використовується для зберігання і обробки даних). Розрізняють тонкі клієнти і товсті клієнти.

Технології розроблення ПЗ Тонкий клієнт - клієнт, який повністю всі операції (або більшість, пов'язаних з логікою роботи програми) передає для обробки на сервер, а сам зберігає лише візуальне уявлення одержуваних від сервера відповідей. Грубо кажучи, тонкий клієнт - набір форм відображення і канал зв'язку з сервером.



public class CommunicationServer {

public void start() {

// Логіка запуску сервера

System.out.println("Server started");

}

public void stop() {

// Логіка зупинки сервера

System.out.println("Server stopped");

}

public void processRequest(ClientRequest request) {

// Логіка обробки запитів від клієнтів

System.out.println("Processing request from client: " + request.getContent());

}

}

// Клас ClientRequest для передачі запитів від клієнтів

public class ClientRequest {

private String content;

public ClientRequest(String content) {

this.content = content;

}

public String getContent() {

return content;

}

}

// Клас Client

public class CommunicationClient {

private CommunicationServer server;

public CommunicationClient(CommunicationServer server) {

this.server = server;

}

public void sendRequest(String content) {

// Логіка відправки запиту на сервер

server.processRequest(new ClientRequest(content));

}

}

public class Main {

public static void main(String[] args) {

CommunicationServer server = new CommunicationServer();

server.start();

CommunicationClient client1 = new CommunicationClient(server);

CommunicationClient client2 = new CommunicationClient(server);

client1.sendRequest("Hello from client 1");

client2.sendRequest("Hi from client 2");

server.stop();

}

}