Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання практичної роботи № 2**

**з дисципліни «Програмне забезпечення мережевих технологій»**

**на тему «створення клієнт/серверних програм за допомогою бібліотек Java»**

**Лектор:**

проф. кафедри ПЗ

Мельник Р.А.

**Виконав:**

студ. групи ПІ-31

Павленчик М. М.

**Прийняв:**

проф. кафедри ПЗ

Мельник Р.А.

«\_\_\_» \_\_\_\_\_\_ 2017 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2017

**ТЕМА РОБОТИ**: Cтворення клієнт/серверних програм за допомогою бібліотек Java.

**Завдання роботи**

Ознайомитись із концепцією клієнт/серверних застосувань. Створити клієнт/серверний застосунок та передати дані з клієнта на сервер.

**Результати виконання**

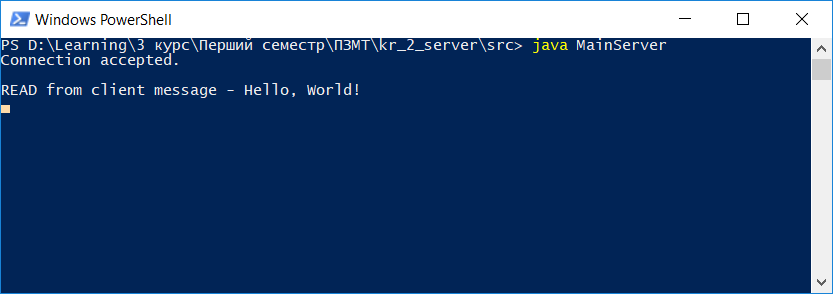


Рис. 1. Робота сервера

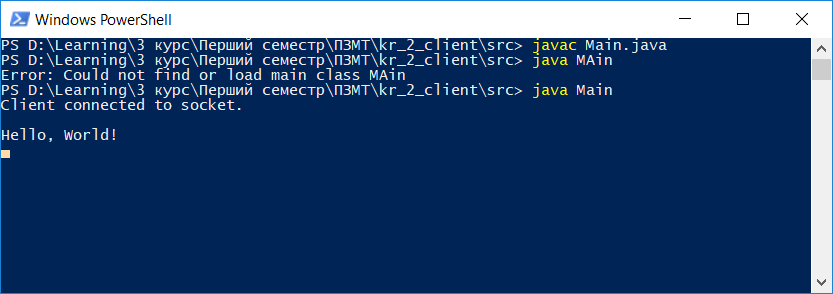


Рис. 2. Робота клієнта

Вихідний код

//Код серверу

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

public class MainServer {

static void main(String[] args) throws InterruptedException {

try (ServerSocket server= new ServerSocket(3345)){

Socket client = server.accept();

System.out.println("Connection accepted.");

System.out.println();

DataOutputStream out = new DataOutputStream(client.getOutputStream());

DataInputStream in = new DataInputStream(client.getInputStream());

out.writeUTF("Hello, client! Write \"Exit\" to close connection.");

out.flush();

while(!client.isClosed()){

String entry = in.readUTF();

System.out.println("READ from client message - " + entry);

if(entry == "Exit"){

System.out.println("Client initialize connections quit.");

out.writeUTF("Ok. Bye!");

out.flush();

break;

}

out.writeUTF("Message received!");

out.flush();

}

System.out.println("Client disconnected");

System.out.println("Closing connections & channels.");

in.close();

out.close();

client.close();

server.close();

System.out.println("Closing connections & channels - DONE.");

} catch (IOException e) {

e.printStackTrace();

}

}

}

//Код клієнта

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

import java.net.UnknownHostException;

public class Main {

public static void main(String[] args) throws InterruptedException {

try(Socket socket = new Socket("localhost", 3345)) {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

DataOutputStream oos = new DataOutputStream(socket.getOutputStream());

DataInputStream ois = new DataInputStream(socket.getInputStream());

System.out.println("Client connected to socket.");

System.out.println();

while(!socket.isOutputShutdown()){

if(br.ready()){

String clientCommand = br.readLine();

oos.writeUTF(clientCommand);

oos.flush();

if(clientCommand == "Exit") {

Thread.sleep(200);

if(ois.read() > -1) {

System.out.println("reading...");

String in = ois.readUTF();

System.out.println(in);

}

break;

}

Thread.sleep(200);

if(ois.read() > -1) {

String in = ois.readUTF();

System.out.println(in);

}

}

}

br.close();

ois.close();

oos.close();

System.out.println("Closing connections & channels on clientSide - DONE.");

} catch (UnknownHostException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}