**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Н-Н ІНСТИТУТ ПІДПРИЄМНИЦТВА ТА ПЕРСПЕКТИВНИХ ТЕХНОЛОГІЙ**



**ЗВІТ**

до виконаної лабораторної роботи №3

з дисципліни *«Клієнт-серверне програмування»*

на тему

**« Клієнт-серверне програмування в Java »**

Виконав:

студент гр. КН-214

**Дубницький Ю.І.**

Прийняв:

**Скобилко І.Я.**

**Львів–2019**

**Лабораторна робота №3**

**Тема:***Клієнт-серверне програмування в Java*

**Мета:***Навчитися писати програми типу клієнт-сервер на Java*

**Теоретичні відомості**

На сьогоднішній день використання клієнтів служб миттєвого обміну повідомленнями (instantmessanger) стало незамінним засобом для всіх користувачів Інтернету. Існує безліч різних протоколів і клієнтів (MSN, ICQ, Skype і т. Д.), Про які кожен чув і які ми всі щодня використовуємо. Але не всі знають, що покладено в основу їх роботи. Припустимо, ви встановили один з клієнтів служб миттєвого обміну повідомленнями на ваш комп'ютер. Після його запуску і введення імені і пароля вашого користувача, програма намагається підключитися до сервера. Що саме означає слово "підключитися"?

Кожен комп'ютер в мережі має IP-адресу. Ця електронна адреса була схожа на вашу домашню адресу - він однозначно ідентифікує ваш комп'ютер і дозволяє іншим спілкуватися з вашим комп'ютером. Не будемо вдаватися в подробиці IP-адреси, хочу тільки зазначити що IP-адреса це набір номерів між якими ставлять крапку (наприклад, 64.104.137.158). Хоча існує й інший спосіб ідентифікації комп'ютерів в мережі - доменне ім'я, яке більш зручне і наочніше ідентифікує комп'ютер, ніж простий набір чисел (наприклад, www.quizful.net). В Інтернеті існують спеціальні комп'ютери, які здійснюють перетворення доменного імені в IP-адресу і навпаки.

**Код**

**Клієнт:**

import java.net.\*;

import java.io.\*;

publicclassClient {

publicstaticvoidmain(String[] ar) {

intserverPort = 6666;

Stringaddress = "127.0.0.1";

try {

InetAddressipAddress = InetAddress.getByName(address);

System.out.println("Anyofyouheardof a socketwith IP address " + address + " andport " + serverPort + "?");

Socketsocket = newSocket(ipAddress, serverPort

System.out.println("Yes! I justgotholdoftheprogram.");

InputStreamsin = socket.getInputStream();

OutputStreamsout = socket.getOutputStream();

DataInputStreamin = newDataInputStream(sin);

DataOutputStreamout = newDataOutputStream(sout);

BufferedReaderkeyboard = newBufferedReader(newInputStreamReader(System.in));

Stringline = null;

System.out.println("Typeinsomethingandpressenter. Willsendittotheserverandtellyawhatitthinks.");

System.out.println();

while (true) {

line = keyboard.readLine();

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

out.writeUTF(line);

out.flush();

line = in.readUTF();

System.out.println("Theserverwasverypolite. Itsentmethis : " + line);

System.out.println("Looksliketheserverispleasedwithus. Goaheadandentermorelines.");

System.out.println();

}

} catch (Exception x) {

x.printStackTrace();

}

}

}

**Сервер:**

import java.net.\*;

import java.io.\*;

publicclass Server {

publicstaticvoidmain(String[] ar) {

intport = 6666;

try {

ServerSocketss = newServerSocket(port);

System.out.println("Waitingfor a client...");

Socketsocket = ss.accept();

System.out.println("Got a client :) ... Finally, someonesawmethroughallthecover!");

System.out.println();

InputStreamsin = socket.getInputStream();

OutputStreamsout = socket.getOutputStream();

DataInputStreamin = newDataInputStream(sin);

DataOutputStreamout = newDataOutputStream(sout);

Stringline = null;

while(true) {

line = in.readUTF();

System.out.println("Thedumbclientjustsentmethisline : " + line);

System.out.println("I'msendingitback...");

out.writeUTF(line);

out.flush();

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

System.out.println();

}

} catch(Exception x) { x.printStackTrace(); }

}

}

**Висновок:***В цій лабораторній роботі я навчився писати програми типу клієнт-сервер на Java.*