**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

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



**ЗВІТ**

до виконаної лабораторної роботи №2

з дисципліни *«Клієнт-серверне програмування»*

на тему

**« Програмування сокетів »**

Виконав:

студент гр. КН-214

**Дубницький Ю.І.**

Прийняв:

**Скобилко І.Я.**

**Львів–2019**

**Лабораторна робота №2**

**Тема:***Програмування сокетів*

**Мета:***Навчитися програмувати сокети*

**Теоретичні відомості**

Сокети (англ. Socket - роз'єм) - назва програмного інтерфейсу для забезпечення обміну даними між процесами. Процеси при такому обміні можуть виконуватися як на одній ЕОМ, так і на різних ЕОМ, пов'язаних між собою мережею. Сокет - абстрактний об'єкт, який представляє кінцеву точку з'єднання.

Кожен процес може створити слухаючийсокет (серверний сокет) і прив'язати його до якого-небудь порту операційної системи (в UNIX непривілейовані процеси не можуть використовувати порти менше 1024). Слухаючий процес зазвичай знаходиться в циклі очікування, тобто прокидається при появі нового з'єднання. При цьому зберігається можливість перевірити наявність з'єднань на даний момент, встановити тайм-аут для операції і т.д.

Кожен сокет має свою адресу. ОС сімейства UNIX можуть підтримувати багато типів адрес, але обов'язковими є INET-адреса і UNIX-адреса. Якщо прив'язати сокет до UNIX-адреси, то буде створений спеціальний файл (файл сокета) по заданому шляху, через який зможуть повідомлятися будь-які локальні процеси шляхом читання / запису з нього (див. Доменний сокетUnix). Сокети типу INET доступні з мережі і вимагають виділення номера порту.

Зазвичай клієнт явно під'єднується до слухача, після чого будь-читання або запис через його дескриптор передаватимуть дані між ним і сервером.

**Код**

**Клієнт:**

packageclient;

import java.io.\*;

import java.net.\*;

publicclassClient{

publicstaticvoidmain(String[] args) throwsIOException {

Socketsocket;

PrintWriterout;

BufferedReader b=newBufferedReader(newInputStreamReader(System.in));

socket = newSocket("127.0.0.1", 8090);

System.out.println(socket.getInetAddress());

Stringstr=b.readLine();

out = newPrintWriter(socket.getOutputStream());

out.println(str);

}

}

**Сервер:**

packageserver;

importjava.io.BufferedReader;

importjava.io.IOException;

importjava.io.InputStreamReader;

import java.net.\*;

publicclass Server {

publicstaticvoidmain(String[] args) throwsIOException {

ServerSocket s;

Socketsocket;

BufferedReaderin;

s = newServerSocket(8090);

socket = s.accept();

System.out.println(socket.getInetAddress());

in = newBufferedReader(newInputStreamReader(socket.getInputStream()));

System.out.print(in.readLine());

}

}

**Висновок:***В цій лабораторній роботі я навчився програмувати сокети.*