**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

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



**ЗВІТ**

до виконаної лабораторної роботи №4

з дисципліни *«Клієнт-серверне програмування»*

на тему

**« Програмування сокетів наJava »**

Виконав:

студент гр. КН-214

**Дубницький Ю.І.**

Прийняв:

**Скобилко І.Я.**

**Львів–2019**

**Лабораторна робота №4**

**Тема:***Програмування сокетів на Java*

**Мета:***Навчитися програмувати сокети на Java*

**Теоретичні відомості**

У цій лабораторній роботі описується базова одностороння настройка клієнта і сервера, при якій клієнт підключається, відправляє повідомлення на сервер, а сервер показує їх за допомогою сокетного з'єднання. Є багато речей низького рівня, які повинні відбутися, щоб ці речі працювали, але мережевий пакет Java API (java.net) подбає про все це, роблячи програмування мережі дуже простим для програмістів.

**Код**

**Клієнт:**

importjava.net.\*;

importjava.io.\*;

publicclassClient

{

    privateSocketsocket            = null;

    privateDataInputStream  input   = null;

    privateDataOutputStreamout     = null;

    publicClient(Stringaddress, intport)

    {

        try

        {

            socket = newSocket(address, port);

            System.out.println("Connected");

            input  =newDataInputStream(System.in);

            out    = newDataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException u)

        {

            System.out.println(u);

        }

        catch(IOException i)

        {

            System.out.println(i);

        }

        Stringline = "";

        while(!line.equals("Over"))

        {

            try

            {

                line = input.readLine();

                out.writeUTF(line);

            }

            catch(IOException i)

            {

                System.out.println(i);

            }

        }

         try

        {

            input.close();

            out.close();

            socket.close();

        }

        catch(IOException i)

        {

            System.out.println(i);

        }

    }

    publicstaticvoidmain(Stringargs[])

    {

        Clientclient = newClient("127.0.0.1", 5000);

    }

}

**Сервер:**

importjava.net.\*;

importjava.io.\*;

publicclassServer

{

    privateSocket          socket   = null;

    privateServerSocket    server   = null;

    privateDataInputStreamin       =  null;

     publicServer(intport)

    {

        try

        {

            server = newServerSocket(port);

            System.out.println("Serverstarted");

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

            socket = server.accept();

            System.out.println("Clientaccepted");

            in = newDataInputStream(

                newBufferedInputStream(socket.getInputStream()));

            Stringline = "";

            while(!line.equals("Over"))

            {

                try

                {

                    line = in.readUTF();

                    System.out.println(line);

                }

                catch(IOException i)

                {

                    System.out.println(i);

                }

            }

            System.out.println("Closingconnection");

            socket.close();

            in.close();

        }

        catch(IOException i)

        {

            System.out.println(i);

        }

    }

    publicstaticvoidmain(Stringargs[])

    {

        Serverserver = newServer(5000);

    }

}

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