---------------------------------------------------------------------------

**Name : Atharva Paliwal**

**Roll No : 40 [5B]**

---------------------------------------------------------------------------

**\*\*\* EXPERIMENT NO: 04 \*\*\***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**AIM : i)To study networking in Java &amp; study InetAddress , socket class**

**ii)Write a program for creating chat server.**

**iii)Design a client server model which will communicate by using Image.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODES:**

1. **INETADDRESS**

import java.io.IOException;

import java.net.InetAddress;

/\*The java.net.InetAddress class provides

methods to get the IP of any host name for example www.javatpoint.com,

www.google.com, www.facebook.com, etc.\*/

public class inetAddress

{

public static void main(String[] args) throws IOException, InterruptedException, Exception

{

InetAddress ip1 = InetAddress.getByName("www.amazon.com"); //remote website

System.out.println("Host Name: "+ ip1.getHostName());

System.out.println("IP Address: "+ ip1.getHostAddress());

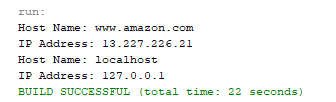
InetAddress ip2 = InetAddress.getByName("localhost"); //local pc

System.out.println("Host Name: "+ ip2.getHostName());

System.out.println("IP Address: "+ ip2.getHostAddress());

}

}



1. **CLIENT TEXT**

import java.io.BufferedReader;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class ClientText

{

public static void main(String[] args) throws IOException, Exception

{

new Client().client();

}

}

class Client

{

public void client() throws Exception

{

// Create client socket

Socket s = new Socket("localhost", 2546);

System.out.println("Connecting to Server");

// to send data to the server

DataOutputStream dos = new DataOutputStream(s.getOutputStream());

// to read data coming from the server

InputStreamReader ir1 = new InputStreamReader(s.getInputStream());

BufferedReader br1 = new BufferedReader(ir1);

// write data that to be send or "exit" to end connection

InputStreamReader ir2 = new InputStreamReader(System.in);

BufferedReader br2 = new BufferedReader(ir2);

String str1, str2;

System.out.print("Client: ");

// repeat as long as exit

// is not typed at client

while (!(str1 = br2.readLine()).equals("exit")) {

// write data that to be send or "exit" to end connection

dos.writeBytes(str1 + "\n");

// receive from the server

str2 = br1.readLine();

System.out.println("Server: "+str2);

System.out.print("Client: ");

}

// close all connection.

dos.close();

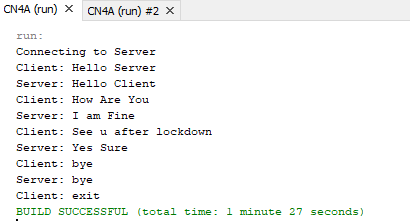
br1.close();

br2.close();

s.close();

}

}



1. **SERVER TEXT**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.ServerSocket;

import java.net.Socket;

public class ServerText

{

public static void main(String[] args) throws IOException, Exception {

new Server().server();

}

}

class Server

{

public void server() throws Exception

{

// Create server Socket

ServerSocket ss = new ServerSocket(2546);

// connect it to client socket

Socket s = ss.accept();

System.out.println("Connected to client Successfully");

// to send data to the client

PrintStream ps = new PrintStream(s.getOutputStream());

// to read data coming from the client

InputStreamReader ir1 = new InputStreamReader( s.getInputStream());

BufferedReader br1 = new BufferedReader( ir1);

// enter data for sending to client

InputStreamReader ir2 = new InputStreamReader(System.in);

BufferedReader br2 = new BufferedReader( ir2);

// server executes continuously

while (true) {

String str1, str2;

// read from client

while ((str1 = br1.readLine()) != null) {

System.out.println("Client: " + str1);

System.out.print("Server: ");

str2 = br2.readLine();

// send to client

ps.println(str2);

}

// close all connection

ps.close();

br1.close();

br2.close();

ss.close();

s.close();

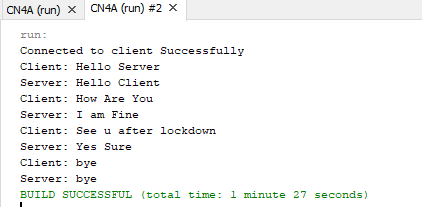
// terminate application

System.exit(0);

} // end of while

}

}



1. **CLIENT IMAGE**

import java.io.BufferedReader;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class ClientText

{

public static void main(String[] args) throws IOException, Exception

{

new Client().client();

}

}

class Client

{

public void client() throws Exception

{

// Create client socket

Socket s = new Socket("localhost", 2546);

System.out.println("Connecting to Server");

// to send data to the server

DataOutputStream dos = new DataOutputStream(s.getOutputStream());

// to read data coming from the server

InputStreamReader ir1 = new InputStreamReader(s.getInputStream());

BufferedReader br1 = new BufferedReader(ir1);

// write data that to be send or "exit" to end connection

InputStreamReader ir2 = new InputStreamReader(System.in);

BufferedReader br2 = new BufferedReader(ir2);

String str1, str2;

System.out.print("Client: ");

// repeat as long as exit

// is not typed at client

while (!(str1 = br2.readLine()).equals("exit")) {

// write data that to be send or "exit" to end connection

dos.writeBytes(str1 + "\n");

// receive from the server

str2 = br1.readLine();

System.out.println("Server: "+str2);

System.out.print("Client: ");

}

// close all connection.

dos.close();

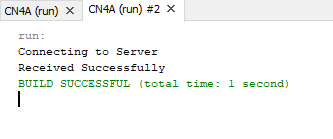
br1.close();

br2.close();

s.close();

}

}



1. **SERVER IMAGE**

import java.io.File;

import java.io.IOException;

import java.io.OutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.nio.file.Files;

public class ServerImage

{

public static void main(String[] args) throws IOException, Exception

{

new Server1().server();

}

}

class Server1

{

public void server() throws Exception

{

// Create server Socket

ServerSocket sc = new ServerSocket(2546);

// connect it to client socket

Socket s = sc.accept();

System.out.println("Connected to client Successfully");

// save image in file

File f = new File("C:\\Users\\user\\Pictures\\Screenshots\\out2a.jpg");

// convert image into byte array

byte[] b = Files.readAllBytes(f.toPath());

//send image to client

OutputStream os = s.getOutputStream();

os.write(b,0,b.length);

System.out.println("Sending image...\nCompleted");

// closing all connections

s.close();

}

}

