import java.io.\*;

import java.net.\*;

public class InetClient{

public static void main (String args[]) {

String serverName;

if (args.length < 1) serverName = "localhost";

else serverName = args[0];

System.out.println("Jess Bender's Inet Client, 1.8.\n");

System.out.println("Using server: " + serverName + ", Port: 1581");

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

try {

String name;

do {

System.out.print

("Enter a hostname or an IP address, (quit) to end: ");

System.out.flush ();

name = in.readLine ();

if (name.indexOf("quit") < 0)

getRemoteAddress(name, serverName);

} while (name.indexOf("quit") < 0);

System.out.println ("Cancelled by user request.");

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

}

static String toText (byte ip[]) {

StringBuffer result = new StringBuffer ();

for (int i = 0; i < ip.length; ++ i) {

if (i > 0) result.append (".");

result.append (0xff & ip[i]);

}

return result.toString ();

}

static void getRemoteAddress (String name, String serverName){

Socket sock;

BufferedReader fromServer;

PrintStream toServer;

String textFromServer;

try{

sock = new Socket(serverName, 1581);

fromServer =

new BufferedReader(new InputStreamReader(sock.getInputStream()));

toServer = new PrintStream(sock.getOutputStream());

toServer.println(name); toServer.flush();

for (int i = 1; i <=3; i++){

textFromServer = fromServer.readLine();

if (textFromServer != null) System.out.println(textFromServer);

}

sock.close();

} catch (IOException x) {

System.out.println ("Socket error.");

x.printStackTrace ();

}

}

}

import java.io.\*;

import java.net.\*;

class Worker extends Thread {

Socket sock;

Worker (Socket s) {sock = s;}

public void run(){

PrintStream out = null;

BufferedReader in = null;

try {

in = new BufferedReader

(new InputStreamReader(sock.getInputStream()));

out = new PrintStream(sock.getOutputStream());

try {

String name;

name = in.readLine ();

System.out.println("Looking up " + name);

printRemoteAddress(name, out);

} catch (IOException x) {

System.out.println("Server read error"); x.printStackTrace ();

}

sock.close();

} catch (IOException ioe) {System.out.println(ioe);}

}

static void printRemoteAddress (String name, PrintStream out) {

try {

out.println("Looking up " + name + "...");

InetAddress machine = InetAddress.getByName (name);

out.println("Host name : " + machine.getHostName ());

out.println("Host IP : " + toText (machine.getAddress ()));

} catch(UnknownHostException ex) {

out.println ("Failed in atempt to look up " + name);

}

}

static String toText (byte ip[]) {

StringBuffer result = new StringBuffer ();

for (int i = 0; i < ip.length; ++ i) {

if (i > 0) result.append (".");

result.append (0xff & ip[i]);

}

return result.toString ();

}

}

public class InetServer {

public static void main(String a[]) throws IOException {

int q\_len = 6;

int port = 1581;

Socket sock;

ServerSocket servsock = new ServerSocket(port, q\_len);

System.out.println

("Jess Bender's Inet server 1.8 starting up, listening at port 1565.\n");

while (true) {

sock = servsock.accept();

new Worker(sock).start();

}

}

}