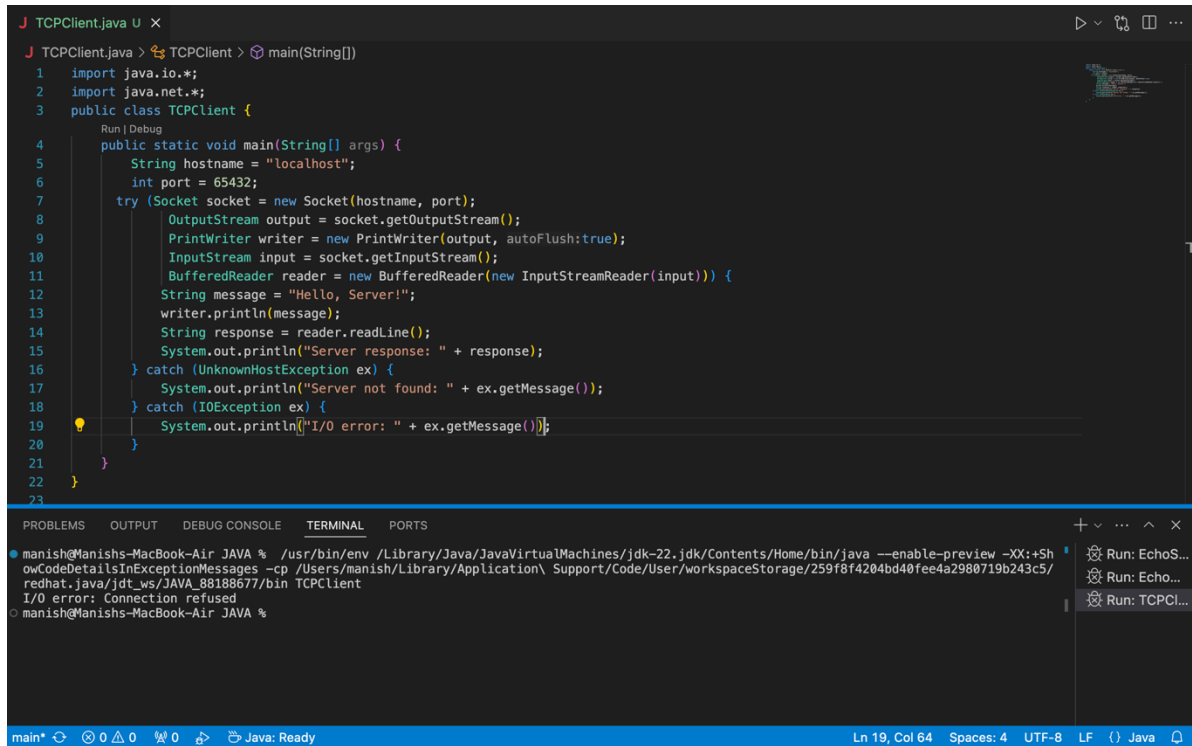


TCP Client :

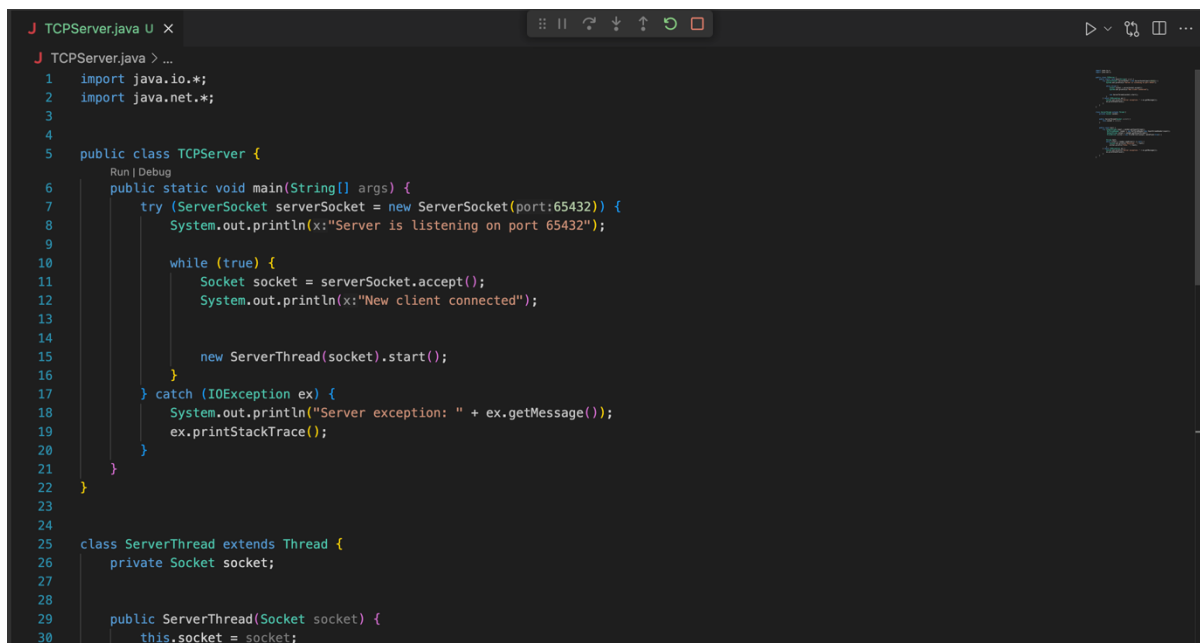


The screenshot shows an IDE with a Java file named `TCPClient.java`. The code defines a `TCPClient` class with a `main` method that connects to `localhost` on port `65432`. It sends the message "Hello, Server!" and prints the server's response. The terminal output shows the command `java TCPClient` being executed, and the output is "Server response: Hello, Server!".

```
J TCPClient.java U X
J TCPClient.java > TCPClient > main(String[])
1  import java.io.*;
2  import java.net.*;
3  public class TCPClient {
4      public static void main(String[] args) {
5          String hostname = "localhost";
6          int port = 65432;
7          try (Socket socket = new Socket(hostname, port);
8              OutputStream output = socket.getOutputStream();
9              PrintWriter writer = new PrintWriter(output, autoFlush:true);
10             InputStream input = socket.getInputStream();
11             BufferedReader reader = new BufferedReader(new InputStreamReader(input))) {
12              String message = "Hello, Server!";
13              writer.println(message);
14              String response = reader.readLine();
15              System.out.println("Server response: " + response);
16          } catch (UnknownHostException ex) {
17              System.out.println("Server not found: " + ex.getMessage());
18          } catch (IOException ex) {
19              System.out.println("I/O error: " + ex.getMessage());
20          }
21      }
22  }
23

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
manish@Manishs-MacBook-Air JAVA % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/manish/Library/Application\ Support/Code/User/workspaceStorage/259f8f4204bd40fee4a2980719b243c5/redhat.java/jdt_ws/JAVA_88188677/bin TCPClient
I/O error: Connection refused
manish@Manishs-MacBook-Air JAVA %
```

TCP Server :



The screenshot shows an IDE with a Java file named `TCPServer.java`. The code defines a `TCPServer` class with a `main` method that listens on port `65432`. It accepts connections and prints "New client connected". It also defines a `ServerThread` class that extends `Thread` and implements the `run` method to handle the connection.

```
J TCPServer.java U X
J TCPServer.java > ...
1  import java.io.*;
2  import java.net.*;
3
4
5  public class TCPServer {
6      public static void main(String[] args) {
7          try (ServerSocket serverSocket = new ServerSocket(port:65432)) {
8              System.out.println(x:"Server is listening on port 65432");
9
10             while (true) {
11                 Socket socket = serverSocket.accept();
12                 System.out.println(x:"New client connected");
13
14                 new ServerThread(socket).start();
15             }
16         } catch (IOException ex) {
17             System.out.println("Server exception: " + ex.getMessage());
18             ex.printStackTrace();
19         }
20     }
21 }
22
23
24
25 class ServerThread extends Thread {
26     private Socket socket;
27
28     public ServerThread(Socket socket) {
29         this.socket = socket;
30     }
31 }
```

```
J TCPServer.java U
J TCPServer.java > ServerThread > socket
25 class ServerThread extends Thread {
26     public ServerThread(Socket socket) {
27     }
28 }
29
30 public void run() {
31     try (InputStream input = socket.getInputStream();
32         BufferedReader reader = new BufferedReader(new InputStreamReader(input));
33         OutputStream output = socket.getOutputStream();
34         PrintWriter writer = new PrintWriter(output, autoFlush:true)) {
35
36
37         String text;
38         while ((text = reader.readLine()) != null) {
39             System.out.println("Received: " + text);
40             writer.println("Echo: " + text);
41         }
42     } catch (IOException ex) {
43         System.out.println("Server exception: " + ex.getMessage());
44         ex.printStackTrace();
45     }
46 }
47
48 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: TCPServer + - - - ^ x
manish@Manishs-MacBook-Air JAVA % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/manish/Library/Application\ Support/Code/User/workspaceStorage/259f8f4204bd40fee4a2980719b243c5/redhat.java/jdt_ws/JAVA_88188677 /bin TCPServer
Server is listening on port 65432
```

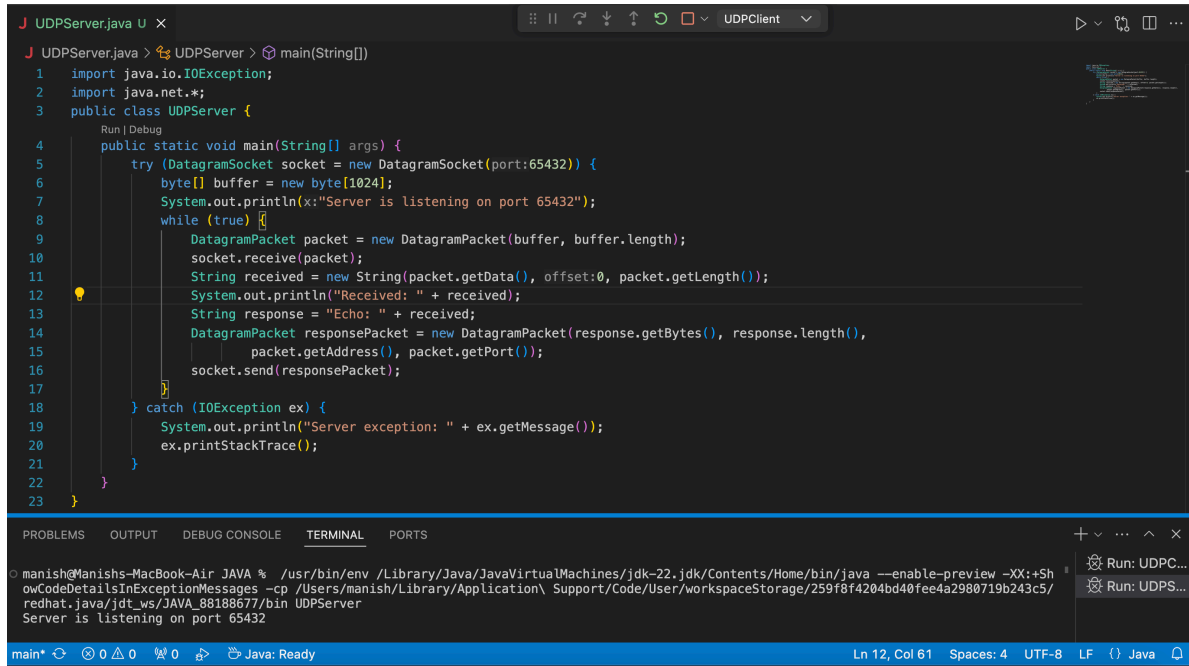
UDP Client :

```
J UDPClient.java U x
J UDPClient.java > UDPClient > main(String[])
1 import java.io.IOException;
2 import java.net.*;
3 public class UDPClient {
4     public static void main(String[] args) {
5         String hostname = "localhost";
6         int port = 65432;
7         try (DatagramSocket socket = new DatagramSocket()) {
8             byte[] buffer = new byte[1024];
9             String message = "Hello, Server! im ready";
10            InetAddress address = InetAddress.getByName(hostname);
11            DatagramPacket packet = new DatagramPacket(message.getBytes(), message.length(), address, port);
12            socket.send(packet);
13            packet = new DatagramPacket(buffer, buffer.length);
14            socket.receive(packet);
15            String response = new String(packet.getData(), offset:0, packet.getLength());
16            System.out.println("Server response: " + response);
17        } catch (IOException ex) {
18            System.out.println("Client error: " + ex.getMessage());
19            ex.printStackTrace();
20        }
21    }
22 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: UDPClient + - - - ^ x
manish@Manishs-MacBook-Air JAVA % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/manish/Library/Application\ Support/Code/User/workspaceStorage/259f8f4204bd40fee4a2980719b243c5/redhat.java/jdt_ws/JAVA_88188677 /bin UDPClient

main* 0 0 0 0 Java: Ready Ln 14, Col 36 Spaces: 4 UTF-8 LF () Java
```

UDP Server :



The screenshot shows an IDE with a Java file named `UDPServer.java`. The code implements a simple UDP server that listens on port 65432 and echoes back any received message. The terminal output shows the server starting and listening on the specified port.

```
J UDPServer.java U x
J UDPServer.java > UDPServer > main(String[])
1  import java.io.IOException;
2  import java.net.*;
3  public class UDPServer {
4      public static void main(String[] args) {
5          try (DatagramSocket socket = new DatagramSocket(port:65432)) {
6              byte[] buffer = new byte[1024];
7              System.out.println("Server is listening on port 65432");
8              while (true) {
9                  DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
10                 socket.receive(packet);
11                 String received = new String(packet.getData(), offset:0, packet.getLength());
12                 System.out.println("Received: " + received);
13                 String response = "Echo: " + received;
14                 DatagramPacket responsePacket = new DatagramPacket(response.getBytes(), response.length(),
15                     packet.getAddress(), packet.getPort());
16                 socket.send(responsePacket);
17             }
18         } catch (IOException ex) {
19             System.out.println("Server exception: " + ex.getMessage());
20             ex.printStackTrace();
21         }
22     }
23 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
manish@Manishs-MacBook-Air JAVA % /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-22.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/manish/Library/Application\ Support/Code/User/workspaceStorage/259f8f4204bd40fee4a2980719b243c5/redhat.java/jdt_ws/JAVA_88188677/bin UDPServer
Server is listening on port 65432
```

main* 0 0 0 0 Java: Ready Ln 12, Col 61 Spaces: 4 UTF-8 LF {} Java