**Prepare UDP Client-Server application that receive and send message.**

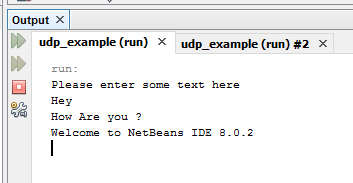
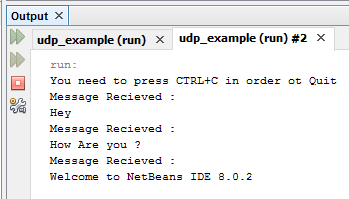
**UDPServer.java**

1. package udp\_example;
2. import java.io.IOException;
3. import java.net.\*;
4. public class UDPServer
5. {
6. public static DatagramSocket mysocket;
7. public static byte myBuffer[] = new byte[2000];
8. public static void serverMethod() throws Exception
9. {
10. int position=0;
11. while(true)
12. {
13. int charData=System.in.read();
14. switch(charData)
15. {
16. case -1: System.out.println("The execution of the server has been terminated");
17. return;
18. case '\r':break;
19. case '\n': mysocket.send(new DatagramPacket(myBuffer,position,InetAddress.getLocalHost(),777));
20. position=0;
21. break;
22. default: myBuffer[position++]=(byte) charData;
23. }
24. }
25. }
26. public static void main(String[] args) throws IOException, Exception {
27. System.out.println("Please enter some text here ");
28. mysocket = new DatagramSocket(888);
29. serverMethod();
30. }
31. }

**UDPClient.java**

1. package udp\_example;
2. import java.net.\*;
3. public class UDPClient
4. {
5. public static DatagramSocket mySocket;
6. public static byte myBuffer[] = new byte[2000];
8. public static void clientMethod() throws Exception
9. {
10. while(true)
11. {
12. DatagramPacket dataPacket = new DatagramPacket (myBuffer,myBuffer.length);
13. mySocket.receive(dataPacket);
14. System.out.println("Message Recieved : ");
15. System.out.println(new String(dataPacket.getData(),0,dataPacket.getLength()));
16. }
17. }
18. public static void main(String[] args) throws SocketException, Exception
19. {
20. System.out.println("You need to press CTRL+C in order ot Quit");
21. mySocket = new DatagramSocket(777);
22. clientMethod();
23. }
24. }

**OUTPUT**

** **

**Server Client**