

```
1 import java.net.*;
2 import java.io.*;
3 import java.util.Scanner;
4
5 public class RemoteVariableClientUDP{
6     static InetAddress aHost;
7     static int serverPort;
8     static DatagramSocket aSocket;
9     static String replyString = "";
10
11
12     public static void main(String args[]){
13         // args give message contents and server
hostname
14         System.out.println("The client is running."
);
15         try {
16             // set to localhost to host on local
machine and set port
17             aHost = InetAddress.getByName("
localhost");
18             serverPort = 6789;
19
20             // input server port to use
21             System.out.print("Input a server side
port number: ");
22             Scanner readline = new Scanner(System.
in);
23             serverPort = readline.nextInt();
24
25
26             String ID = "";
27             String operation = "";
28             String value = "";
29
30             String packet = "";
31
32             while (true) {
33                 System.out.println("\n1. Add a
value to your sum.\n" +
34                 "2. Subtract a value from
```

```
34 your sum.\n" +
35         "3. Get your sum.\n" +
36         "4. Exit client");
37
38         int in = readline.nextInt();
39
40         if(in == 1){
41             operation = "1 ";
42             System.out.println("Enter value
43         to add: ");
44             value = String.valueOf(readline
45         .nextInt());
46             System.out.println("Enter your
47         ID: ");
48             ID = String.valueOf(readline.
49         nextInt()) + " ";
50             packet += operation + ID +
51         value;
52         }
53         else if(in == 2){
54             operation = "2 ";
55             System.out.println("Enter value
56         to subtract: ");
57             value = String.valueOf(readline
58         .nextInt());
59             System.out.println("Enter your
60         ID: ");
61             ID = String.valueOf(readline.
62         nextInt()) + " ";
63             packet += operation + ID +
64         value;
65         }
66         else if(in == 3){
67             operation = "3 ";
68             System.out.println("Enter your
69         ID: ");
70             ID = String.valueOf(readline.
```

```

63 nextInt()) + " ";
64         value = "";
65
66         packet += operation + ID +
        value;
67     }
68     else if(in == 4){
69         System.out.println("Client
        side quitting. The remote variable server is still
        running."); // send reply if not halt
70         break;
71     }
72
73     System.out.println("The result is
        : " + communicate(packet));
74
75     packet = "";
76
77     }
78
79     // catch potential exceptions
80     }catch (SocketException e) {System.out.
        println("Socket: " + e.getMessage());
81     }catch (IOException e){System.out.println(
        "IO: " + e.getMessage());
82     }finally {if(aSocket != null) aSocket.
        close();}
83     }
84     public static String communicate(String in)
        throws IOException {
85
86         aSocket = new DatagramSocket();
87
88         // operation code
89         byte [] m = in.getBytes(); // convert
        console in into byte packets
90         DatagramPacket operationRequest = new
        DatagramPacket(m, m.length, aHost, serverPort);
91         aSocket.send(operationRequest); // send
        packet
92

```

```
93         byte[] buffer = new byte[1000]; // create
           a buffer to receiver server packet
94         DatagramPacket reply = new DatagramPacket(
           buffer, buffer.length); // format for receiving
           packet
95         aSocket.receive(reply); // receive from
           socket
96         replyString = new String(reply.getData()).
           substring(0,reply.getLength()); // get proper
           length of string
97
98         return replyString;
99     }
100 }
```

```
1 import java.net.*;
2 import java.io.*;
3 import java.util.Scanner;
4 import java.util.*;
5
6 public class RemoteVariableServerUDP{
7     private static int sum = 0;
8     private static TreeMap<Integer, Integer>
tree_map = new TreeMap<Integer, Integer>();
9
10    static String instructions = "1. Add a value to
your sum.\n" +
11        "2. Subtract a value from your sum.\n"
+
12        "3. Get your sum.\n" +
13        "4. Exit client";
14
15    public static void main(String args[]){
16        System.out.println("The server is running."
); // lab instructions
17        DatagramSocket aSocket = null;
18        byte[] buffer = new byte[1000]; // set up
packet buffer for client message
19        try{
20            // set up ports
21            System.out.print("Input a server port
number to listen on: ");
22            Scanner readline = new Scanner(System.
in);
23            int serverPort = readline.nextInt();
// convert to int
24
25            // set up sockets to receive client
packets
26            aSocket = new DatagramSocket(serverPort
);
27            aSocket.setReuseAddress(true);
28
29            DatagramPacket request = new
DatagramPacket(buffer, buffer.length); // syntax
for buffer
```

```
30
31         int id;
32         int value;
33
34         while(true){ // loop to continue until
35             'halt!' is sent
36
37             // operation request and reply
38             aSocket.receive(request); //
39             receive request, and format a reply
40             String requestString = new String(
41                 request.getData()).substring(0,request.getLength
42                 ()).strip(); // proper length
43
44             String[] arrRequestString =
45                 requestString.split(" ");
46
47             id = Integer.valueOf(
48                 arrRequestString[1]);
49
50             if(!tree_map.containsKey(id)){
51                 if(arrRequestString[0].equals("
52                 1")){
53                     value = Integer.valueOf(
54                         arrRequestString[2]);
55                     tree_map.put(id, value);
56                 }
57                 else if(arrRequestString[0].
58                     equals("2")){
59                     value = -1*Integer.valueOf(
60                         arrRequestString[2]);
61                     tree_map.put(id, value);
62                 }
63             }
64             else{
65                 if(arrRequestString[0].equals("
66                 1")){
67                     value = Integer.valueOf(
68                         arrRequestString[2]);
69                     tree_map.replace(id,
70                         tree_map.get(id)+value);
```

```

58         }
59         else if(arrRequestString[0].
    equals("2")){
60             value = Integer.valueOf(
    arrRequestString[2]);
61             tree_map.replace(id,
    tree_map.get(id)-value);
62         }
63     }
64
65     System.out.println("\nVisitorID: "
    + arrRequestString[1]);
66     System.out.println("Operation #: "
    + arrRequestString[0]);
67     System.out.println("Return Variable
    " + String.valueOf(tree_map.get(id)));
68
69
70     byte [] m = (String.valueOf(
    tree_map.get(id))).getBytes(); // convert console
    in into byte packets
71
72     DatagramPacket reply = new
    DatagramPacket(m, m.length, request.getAddress(),
    request.getPort());
73     aSocket.send(reply); // send back
    the reply
74
75
76     }
77     // catch potential exceptions
78     }catch (SocketException e){System.out.
    println("Socket: " + e.getMessage());
79     }catch (IOException e) {System.out.println(
    "IO: " + e.getMessage());
80     }finally {if(aSocket != null) aSocket.close
    ()};
81     }
82 }

```