

Soluzione Esercizio 2

```

1 // Proposta.java
2
3 import java.io.*;
4
5 public class Proposta implements Serializable {
6
7     private String contenuto;
8     private int numeroSI = 0;
9     private int numeroNO = 0;
10    private int numeroNI = 0;
11    private String risultato = "ingiudicata";
12
13    public Proposta(String c) { contenuto = c; }
14
15    public String restituisciContenuto() { return contenuto; }
16
17    public void vota (String voto) {
18        switch(voto){
19            case "SI": numeroSI++; break;
20            case "NO": numeroNO++; break;
21            default: numeroNI++;
22        }
23    }
24
25    public void elabora() {
26        if (numeroNI < numeroSI + numeroNO)
27            if (numeroSI > numeroNO)
28                risultato = "accettata";
29            else if (numeroSI < numeroNO)
30                risultato = "rifiutata";
31    }
32
33    public String restituisciRisultato() { return risultato; }
34 }

```

```

1 // Votante.java
2
3 import java.net.*;
4 import java.io.*;
5
6 public class Votante {
7
8     public static void inviaProposta(int porta, Proposta pr) {
9         try (Socket sock = new Socket("localhost", porta)) {
10             ObjectOutputStream oos =
11                 new ObjectOutputStream(sock.getOutputStream());
12             oos.writeObject(pr);
13         } catch (IOException e) { e.printStackTrace();}
14     }
15
16     public static Proposta riceviProposta(int porta) {
17         Proposta pr = null;
18         try (
19             ServerSocket servsock = new ServerSocket(porta);
20             Socket sock = servsock.accept()
21         ) {
22             ObjectInputStream ois =
23                 new ObjectInputStream(sock.getInputStream());
24             pr = (Proposta)ois.readObject();
25         } catch (IOException | ClassNotFoundException e) { e.printStackTrace();}
26         return pr;
27     }
28
29     public static void main(String[] args) {
30         Proposta pr;
31         if (args.length > 3) {
32             pr = new Proposta(args[3]);
33             System.out.println(pr.restituisceContenuto());
34             pr.vota(args[2]);
35             inviaProposta(Integer.parseInt(args[1]), pr);
36             pr = riceviProposta(Integer.parseInt(args[0]));
37             pr.elabora();
38             System.out.println(pr.restituisceRisultato());
39             inviaProposta(Integer.parseInt(args[1]), pr);
40             riceviProposta(Integer.parseInt(args[0]));
41         } else {
42             pr = riceviProposta(Integer.parseInt(args[0]));
43             System.out.println(pr.restituisceContenuto());
44             pr.vota(args[2]);
45             inviaProposta(Integer.parseInt(args[1]), pr);
46             pr = riceviProposta(Integer.parseInt(args[0]));
47             System.out.println(pr.restituisceRisultato());
48             inviaProposta(Integer.parseInt(args[1]), pr);
49         }
50     }
51 }

```