

Soluzione Esercizio 2

```

1 // Progetto.java
2 import java.io.*;
3
4
5 public class Progetto implements Serializable {
6     private final String obiettivo;
7     private final int numeroMinimoSponsor;
8     private int numeroCorrenteSponsor = 0;
9     private final double finanziamentoMinimo;
10    private double finanziamentoCorrente = 0.0;
11    private int portaRicezioneProponente;
12
13
14    public Progetto(String ob, int nms, double fm, int prp) {
15        obiettivo = ob;
16        numeroMinimoSponsor = nms;
17        finanziamentoMinimo = fm;
18        portaRicezioneProponente = prp;
19    }
20
21    public boolean verificaAttivazione() {
22        if (( numeroCorrenteSponsor >= numeroMinimoSponsor) &&
23            ( finanziamentoCorrente >= finanziamentoMinimo))
24            return true;
25        return false;
26    }
27
28    public String toString() {
29        String r = "obiettivo: " + obiettivo +
30            "; sponsor: " + numeroCorrenteSponsor + "/" + numeroMinimoSponsor +
31            "; finanziamento: " + finanziamentoCorrente + "/" + finanziamentoMinimo;
32        return r;
33    }
34
35    public void sponsorizza(double importo) {
36        if (importo > 0) {
37            finanziamentoCorrente += importo;
38            numeroCorrenteSponsor++;
39        }
40
41    public int restituisciPortaRicezioneProponente() { return portaRicezioneProponente; }
42    public double restituisciFinanziamentoCorrente() { return finanziamentoCorrente; }
43 }

```



```

1 // Sponsor.java
2 import java.net.*;
3 import java.io.*;
4
5
6 public class Sponsor {
7     public static void inviaProgetto(int porta, Progetto p) {
8         try (Socket sock = new Socket("localhost", porta)) {
9             ObjectOutputStream oos =

```

```

11     new ObjectOutputStream(sock.getOutputStream());
12     oos.writeObject(p);
13 } catch (IOException e) { e.printStackTrace();}
14 }
15
16 public static Progetto riceviProgetto(int porta) {
17     Progetto p = null;
18     try (
19         ServerSocket servsock = new ServerSocket(porta);
20         Socket sock = servsock.accept();
21     ) {
22         ObjectInputStream ois =
23             new ObjectInputStream(sock.getInputStream());
24         p = (Progetto)ois.readObject();
25     } catch (IOException | ClassNotFoundException e) { e.printStackTrace();}
26     return p;
27 }
28
29 public static void main(String[] args) {
30     Progetto p;
31     if (args.length > 3) {
32         p = new Progetto(args[3], Integer.parseInt(args[4]),
33             Double.parseDouble(args[5]), Integer.parseInt(args[0])); //1
34         p.sponsorizza(Double.parseDouble(args[2])); //2
35         System.out.println("Sono " + Integer.parseInt(args[0]) +
36             " ed offro " + Double.parseDouble(args[2]));
37         System.out.println(p); //3
38         inviaProgetto(Integer.parseInt(args[1]), p); //4
39         p = riceviProgetto(Integer.parseInt(args[0])); //9
40         if (p.verificaAttivazione()) { //10.a
41             System.out.println("Quota: " +
42                 Double.parseDouble(args[2])/p.restituisceFinanziamentoCorrente()); //10.a.1
43             inviaProgetto(Integer.parseInt(args[1]), p); //10.a.2
44         }
45     } else {
46         System.out.println("Sono " + Integer.parseInt(args[0]) +
47             " ed offro " + Double.parseDouble(args[2]));
48         p = riceviProgetto(Integer.parseInt(args[0])); //5
49         p.sponsorizza(Double.parseDouble(args[2])); //6
50         System.out.println(p); //7
51         if (p.verificaAttivazione()) { //8.a
52             System.out.println("Quota: " +
53                 Double.parseDouble(args[2])/p.restituisceFinanziamentoCorrente()); //8.a.1
54             inviaProgetto(p.restituiscePortaRicezioneProponente(), p); //8.a.2
55             p = riceviProgetto(Integer.parseInt(args[0])); //8.a.3
56         } else { //8.b
57             inviaProgetto(Integer.parseInt(args[1]), p); //8.b.1
58             p = riceviProgetto(Integer.parseInt(args[0])); //8.b.2
59             System.out.println("Quota: " +
60                 Double.parseDouble(args[2])/p.restituisceFinanziamentoCorrente()); //8.b.3
61             inviaProgetto(Integer.parseInt(args[1]), p); //8.b.4
62         }
63     }
64 }
65 }
66 }
67 }

```