

Soluzione Esercizio 2

```
1 // Inserzione.java
2
3 import java.io.*;
4
5 public class Inserzione implements Serializable {
6
7     private final String descrizioneBene;
8     private double miglioreOfferta;
9     private final double incrementoAsta;
10    private final int portaProponente;
11    private int portaCompratore = 0;
12
13    public Inserzione(String db, double mo, double ia, int pp) {
14        descrizioneBene = db;
15        miglioreOfferta = mo;
16        incrementoAsta = ia;
17        portaProponente = pp;
18    }
19
20    public void effettuaOfferta(double prezzo) {
21        if ((( prezzo - miglioreOfferta ) >= incrementoAsta) &&
22            ((( prezzo - miglioreOfferta ) % incrementoAsta) == 0))
23            miglioreOfferta = prezzo;
24    }
25
26    public String toString() {
27        String r = "descrizione: " + descrizioneBene +
28            "\nmigliore offerta: " + miglioreOfferta +
29            "\nincremento d'asta: " + incrementoAsta +
30            "\ncompratore: " + portaCompratore;
31        return r;
32    }
33
34    public double restituisciMiglioreOfferta() { return miglioreOfferta; }
35    public void inserisciPortaCompratore(int pc) { portaCompratore = pc; }
36    public int restituisciPortaProponente() { return portaProponente; }
37 }
```

```

1 // Commerciante.java
2
3 import java.net.*;
4 import java.io.*;
5
6 public class Commerciante {
7
8     public static void inviaInserzione(int porta, Inserzione i) {
9         try (Socket sock = new Socket("localhost", porta)) {
10             ObjectOutputStream oos =
11                 new ObjectOutputStream(sock.getOutputStream());
12             oos.writeObject(i);
13         } catch (IOException e) { e.printStackTrace();}
14     }
15
16     public static Inserzione riceviInserzione(int porta) {
17         Inserzione i = null;
18         try (
19             ServerSocket servsock = new ServerSocket(porta);
20             Socket sock = servsock.accept();
21         ) {
22             ObjectInputStream ois =
23                 new ObjectInputStream(sock.getInputStream());
24             i = (Inserzione)ois.readObject();
25         } catch (IOException | ClassNotFoundException e) { e.printStackTrace();}
26         return i;
27     }
28
29     public static void main(String[] args) {
30         Inserzione i;
31         if (args.length > 3) {
32             i = new Inserzione( args[4], Double.parseDouble(args[2]),
33                                 Double.parseDouble(args[3]), Integer.parseInt(args[0])); //1
34             System.out.println(i); //2
35             inviaInserzione(Integer.parseInt(args[1]), i); //3
36             i = riceviInserzione(Integer.parseInt(args[0])); //12
37             System.out.println(i); //13
38             inviaInserzione(Integer.parseInt(args[1]), i); //14
39             i = riceviInserzione(Integer.parseInt(args[0])); //19
40             System.out.println(i); //20
41         } else {
42             i = riceviInserzione(Integer.parseInt(args[0])); //4
43             System.out.println(i); //5
44             i.effettuaOfferta(Double.parseDouble(args[2])); //6
45             System.out.println( "Sono " + Integer.parseInt(args[0]) +
46                                " ed offro " + Integer.parseInt(args[2]));
47             inviaInserzione(Integer.parseInt(args[1]), i); //7
48             i = riceviInserzione(Integer.parseInt(args[0])); //15
49             if (i.restituisceMiglioreOfferta() == Double.parseDouble(args[2])) { // 16
50                 i.inserisciPortaCompratore(Integer.parseInt(args[0])); //17
51                 inviaInserzione(i.restituiscePortaProponente(), i); //18
52             } else {
53                 inviaInserzione(Integer.parseInt(args[1]), i); //18 alternativo
54             }
55         }
56     }
57 }

```