

Soluzione Esercizio 2

```
1 // RichiestaDiCompetenze.java
2
3 import java.io.*;
4
5 public class RichiestaDiCompetenze implements Serializable {
6
7     private final String descrizione;
8     private final int portaRichiedente;
9     private int portaProponente = 0;
10
11     public RichiestaDiCompetenze(String d, int pr) {
12         descrizione = d;
13         portaRichiedente = pr;
14     }
15
16     public int restituisciPortaRichiedente() { return portaRichiedente; }
17
18     public void inserisciPortaProponente(int pp) {
19         if (!esistePortaProponente())
20             portaProponente = pp;
21     }
22
23     public boolean esistePortaProponente() { return (portaProponente != 0); }
24
25     public String toString() {
26         String r = "descrizione: " + descrizione +
27                 "\nporta richiedente: " + portaRichiedente +
28                 "\nportaProponente: " + portaProponente;
29         return r;
30     }
31 }

```



```
1 // AgenziaDelLavoro.java
2
3 import java.net.*;
4 import java.io.*;
5
6 public class AgenziaDelLavoro {
7
8     public static void invia(int porta, RichiestaDiCompetenze r) {
9         try (Socket sock = new Socket("localhost", porta)) {
10             ObjectOutputStream oos =
11                 new ObjectOutputStream(sock.getOutputStream());
12             oos.writeObject(r);
13         } catch (IOException e) { e.printStackTrace(); }
14     }
15 }

```

```

16 public static RichiestaDiCompetenze ricevi(int porta) {
17     RichiestaDiCompetenze r = null;
18     try (
19         ServerSocket servsock = new ServerSocket(porta);
20         Socket sock = servsock.accept();
21     ) {
22         ObjectInputStream ois =
23             new ObjectInputStream(sock.getInputStream());
24         r = (RichiestaDiCompetenze)ois.readObject();
25     } catch (IOException | ClassNotFoundException e) { e.printStackTrace();}
26     return r;
27 }
28
29 public static void main(String[] args) {
30     System.out.println(java.time.LocalDateTime.now() + ") Sono " + args[0] + ":");
31     RichiestaDiCompetenze r;
32     if (args.length==3 && args[2].endsWith("?")){ // richiedente
33         r = new RichiestaDiCompetenze(args[2], Integer.parseInt(args[0])); //1
34         System.out.println(java.time.LocalDateTime.now() + ")\n" + r); //2
35         invia(Integer.parseInt(args[1]), r); //3
36         r = ricevi(Integer.parseInt(args[0])); //7
37         System.out.println(java.time.LocalDateTime.now() + ")\n" + r); //8
38         invia(Integer.parseInt(args[1]), r); //9
39         r = ricevi(Integer.parseInt(args[0])); //13
40     } else { // altri
41         r = ricevi(Integer.parseInt(args[0])); //4 oppure 10
42         System.out.println(java.time.LocalDateTime.now() + ")\n" + r); //5 oppure 11
43         if (args.length==2) { // 6.b oppure //10
44             invia(Integer.parseInt(args[1]), r); //6.b.1 oppure 12
45             if (!r.esistePortaProponente()) { // deve ancora fare 10
46                 r = ricevi(Integer.parseInt(args[0])); //10
47                 System.out.println(java.time.LocalDateTime.now() + ")\n" + r); //11
48                 invia(Integer.parseInt(args[1]), r); //12
49             } // se 10 lo aveva fatto allora non deve fare altro
50         }
51         else if (args[2].endsWith("!")) { // 6.a ha competenze
52             r.inserisciPortaProponente(Integer.parseInt(args[0])); //6.a.1
53             invia(r.restituiscePortaRichiedente(), r); //6.a.2
54             r = ricevi(Integer.parseInt(args[0])); //10
55             System.out.println(java.time.LocalDateTime.now() + ")\n" + r); //11
56             invia(Integer.parseInt(args[1]), r); //12
57         }
58     }
59 }
60 }

```