

Soluzione Esercizio 2

```

1 // Dati.java
2
3 import java.io.*;
4 import java.util.*;
5
6 public class Dati implements Serializable {
7
8     private final double[] dati;
9
10    public Dati(int dim) {
11        dati = new double[dim];
12    }
13
14    public void inizializza(double valore) {
15        Arrays.fill(dati, valore);
16    }
17
18    public boolean equals(Object d) { //(1)
19        return Arrays.equals(dati, ((Dati)d).dati); //(2)
20    }
21 }
22
23 /*
24 (1) Si usa il parametro generico Object per ridefinire equals ereditato da Object.
25 (2) All'interno del corpo di un metodo di una classe si puo' accedere ai campi
26     privati di parametri di quella classe.
27 */

```



```

1 // BancaDati.java
2
3 import java.net.*;
4 import java.io.*;
5
6 public class BancaDati {
7     public static void main(String[] args) {
8         try {
9             ServerSocket servsock = new ServerSocket(8081);
10             Socket sock = servsock.accept();
11             ObjectInputStream ois =
12                 new ObjectInputStream(sock.getInputStream());
13             Dati dati = (Dati)ois.readObject();
14             servsock.close(); sock.close();
15
16             ObjectOutputStream oos =
17                 new ObjectOutputStream(
18                     new FileOutputStream("data.bin"));
19             oos.writeObject(dati);
20             oos.close();
21
22             ois = new ObjectInputStream(
23                 new FileInputStream("data.bin"));

```

```

24     dati = (Dati)ois.readObject();
25     ois.close();
26
27     sock = new Socket("localhost", 8080); //(1)
28     oos = new ObjectOutputStream(sock.getOutputStream());
29     oos.writeObject(dati);
30     sock.close();
31
32 } catch (IOException | ClassNotFoundException e)
33 { e.printStackTrace(); }
34 }
35 }
36
37 /*
38 (1) L'invio avviene su una diversa connessione rispetto alla ricezione, per
39 consentire protocolli applicativi a più banche dati connesse ad anello.
40 */

1 // SportelloDati.java
2
3 import java.net.*;
4 import java.io.*;
5
6 public class SportelloDati {
7     public static void main(String args[]) {
8         Dati datiBackup, dati = new Dati(Integer.parseInt(args[0]));
9         dati.inizializza(Double.parseDouble(args[1]));
10        try {
11            Socket sock = new Socket("localhost", 8081);
12            ObjectOutputStream oos =
13                new ObjectOutputStream(sock.getOutputStream());
14            oos.writeObject(dati);
15            sock.close();
16
17            ServerSocket servsock = new ServerSocket(8080);
18            sock = servsock.accept(); // (1)
19            ObjectInputStream ois =
20                new ObjectInputStream(sock.getInputStream());
21            datiBackup = (Dati)ois.readObject();
22            servsock.close(); sock.close();
23
24            System.out.println(dati.equals(datiBackup));
25        } catch (IOException | ClassNotFoundException e)
26        { e.printStackTrace(); }
27    }
28 }
29
30 /*
31 (1) Si mette in ascolto su una diversa connessione rispetto alla trasmissione,
32 per consentire protocolli applicativi a più banche dati connesse ad anello.
33 */

```