

## Soluzione Esercizio 2

```
1 // Parametro.java
2 import com.thoughtworks.xstream.*;
3
4 public class Parametro {
5     private boolean stabilitaCampioni;
6     private double sommaCampioni;
7     private int numeroCampioni;
8
9     public Parametro(double c) {
10         sommaCampioni = c; numeroCampioni = 1; stabilitaCampioni = false;
11     }
12
13     public Parametro(String xml) {
14         Parametro p = (Parametro)(new XStream()).fromXML(xml);
15         numeroCampioni = p.numeroCampioni;
16         stabilitaCampioni = p.stabilitaCampioni;
17         sommaCampioni = p.sommaCampioni;
18     }
19
20     public void aggiungiCampione (double c) {
21         if ((Math.round(sommaCampioni/numeroCampioni) !=
22             Math.round((sommaCampioni+c)/(numeroCampioni+1)))
23         ) { sommaCampioni+=c; numeroCampioni++; }
24         else
25             stabilitaCampioni = true;
26     }
27
28     public boolean stabilitaCampioni() {
29         return stabilitaCampioni;
30     }
31
32     public double calcolaParametro() {
33         return Math.round(sommaCampioni/numeroCampioni);
34     }
35
36     public String toString() {
37         return (new XStream()).toXML(this);
38     }
39 }
40
41
42 1 // Terminale.java
43 2 import java.net.*;
44 3 import java.io.*;
45 4 import java.sql.*;
46 5
47 6 public class Terminale {
48 7     private static int portaRicezione, portaInvio;
49 8
50 9     public static void invia(int porta, Parametro p) {
51 10         try (Socket sock = new Socket("localhost", porta);
52 11             DataOutputStream dos =
53 12                 new DataOutputStream(sock.getOutputStream());
54 13             ) { dos.writeUTF(p.toString());
```

```

14     } catch (IOException e) { e.printStackTrace();}
15     visualizzaSuLog("invia", p.toString()); //2
16 }
17
18 public static Parametro ricevi() {
19     Parametro p = null;
20     try (
21         ServerSocket servsock = new ServerSocket(portaRicezione);
22         Socket sock = servsock.accept();
23         DataInputStream dis =
24             new DataInputStream(sock.getInputStream());
25     ) {
26         p = new Parametro(dis.readUTF());
27     } catch (IOException e) { e.printStackTrace();}
28     visualizzaSuLog("ricevi", p.toString()); //2
29     return p;
30 }
31
32 public static void visualizzaSuLog(String operazione, String risultato) {
33     try (
34         Connection co = DriverManager.getConnection("jdbc:mysql://localhost:3306/terminale",
35 "root", "");
36         PreparedStatement ps = co.prepareStatement("INSERT INTO log VALUES (?, ?, ?)");
37     ) {
38         ps.setInt(1, portaRicezione); ps.setString(2, operazione); ps.setString(3, risultato);
39         ps.executeUpdate();
40     } catch (SQLException e) { System.err.println(e.getMessage());}
41 }
42
43 public static void main(String[] args) {
44     portaRicezione = Integer.parseInt(args[1]);
45     portaInvio = Integer.parseInt(args[2]);
46     Parametro p;
47     if (args[1].equals("8080")) { // è la radice
48         visualizzaSuLog("inizio", "*****");
49         p = new Parametro(Double.parseDouble(args[0])); //1
50         visualizzaSuLog("nuovoParametro", p.toString()); //2
51         invia(portaInvio, p); //3
52         p = ricevi(); //6
53         visualizzaSuLog("calcolaParametro", Double.toString(p.calcolaParametro())); //7
54         visualizzaSuLog("fine", "*****");
55     }
56     else { //non è la radice
57         p = ricevi(); //4
58         if (p.stabilitaCampioni()) { //5.a
59             invia(portaInvio, p); //5.a.1
60         } else { //5.b
61             p.aggiungiCampione(Double.parseDouble(args[0])); //5.b.1
62             visualizzaSuLog("aggiungiCampione", p.toString()); //5.b.2
63             if (args.length < 4) { //5.b.3.c foglia
64                 invia(portaInvio, p); //5.b.3.c.1
65             } else { // 5.b.3.d nodo
66                 if (!p.stabilitaCampioni()) { //5.b.3.d.e nodo e campioni instabili
67                     invia(Integer.parseInt(args[3]), p); //5.b.3.d.e.1
68                     p = ricevi(); //5.b.3.d.e.2
69                 }
70                 invia(portaInvio, p); // //5.b.3.d.1
71             }
72         }
73     }
74 }

```

```
73  }
74  }
75
```

Log (formato contratto)

idprocesso operazione risultato (stab, somma, num)

```
8080  inizio *****
8080  nuovoParametro      false 10.2  1
8080  invia false 10.2  1
8081  ricevi false 10.2  1
8081  aggiungiCampione    true  10.2  1
8081  invia true 10.2  1
8082  ricevi true 10.2  1
8082  invia true 10.2  1
8080  ricevi true 10.2  1
8080  calcolaParametro    10.0
8080  fine *****
8080  inizio *****
8080  nuovoParametro      false 7.5   1
8080  invia false 7.5   1
8081  ricevi false 7.5   1
8081  aggiungiCampione    false 26.8  2
8081  invia false 26.8  2
8082  ricevi false 26.8  2
8082  aggiungiCampione    false 44.6  3
8082  invia false 44.6  3
8083  ricevi false 44.6  3
8083  aggiungiCampione    false 55.6  4
8083  invia false 55.6  4
8084  ricevi false 55.6  4
8084  aggiungiCampione    true  55.6  4
8084  invia true 55.6  4
8082  ricevi true 55.6  4
8082  invia true 55.6  4
8080  ricevi true 55.6  4
8080  calcolaParametro    14.0
8080  fine *****
8080  inizio *****
8080  nuovoParametro      false 7.5   1
8080  invia false 7.5   1
8081  ricevi false 7.5   1
8081  aggiungiCampione    false 26.8  2
8081  invia false 26.8  2
8082  ricevi false 26.8  2
8082  aggiungiCampione    false 44.6  3
8082  invia false 44.6  3
8083  ricevi false 44.6  3
8083  aggiungiCampione    false 94.6  4
8083  invia false 94.6  4
8084  ricevi false 94.6  4
8084  aggiungiCampione    false 125.0 5
8084  invia false 125.0 5
8082  ricevi false 125.0 5
8082  invia false 125.0 5
8080  ricevi false 125.0 5
8080  calcolaParametro    25.0
8080  fine *****
```