

```
// Esempio per il design pattern Mediator
/** Elemento e' un Colleague */

public interface Elemento {
    public void put(final String s);

    public void put(final int v);

    public String doWork();

    public void reset();
}

/** Numeri e' un ConcreteColleague */
public class Numeri implements Elemento {
    private final Central m;
    private final int[] seq = { 8, 13, 21 };
    private final int[] num = new int[3];
    private int i = 0;

    public Numeri(final Central c) {
        m = c;
        System.out.println("Indovina i successivi tre numeri della sequenza 2, 3, 4");
    }

    @Override
    public void put(final String msg) {
        System.out.print(msg + " ");
        if (i == 3)
            System.out.println("Bravo, sequenza corretta");
        else
            System.out.println("Sequenza non corretta");
    }

    @Override
    public String doWork() {
        System.out.println("Hai inserito: " + num[0] + " " + num[1] + " " + num[2]);
        if (seq[0] == num[0] && seq[1] == num[1] && seq[2] == num[2]) {
            System.out.println("Hai indovinato la sequenza");
            m.stop();
        } else
            System.out.println("Non hai indovinato la sequenza, riprova");
        i = 0;
        return null;
    }

    @Override
    public void put(final int v) {
        if (i < 3) {
            num[i] = v;
            i++;
        }
    }
}
```

```
        i++;
    }
    if (i == 3)
        doWork();
}

@Override
public void reset() {
    i = 0;
}
}

/** Sommatore e' un ConcreteColleague */

public class Sommatore implements Elemento {
    private final Central m;
    private int somma = 0;
    private int count = 0;
    private int media = 0;

    public Sommatore(final Central c) {
        m = c;
    }

    @Override
    public void put(final String msg) {
        System.out.println(msg);
        doWork();
        System.out.println("La somma e': " + somma);
        System.out.println("La media e': " + media);
    }

    @Override
    public String doWork() {
        if (count > 0)
            media = somma / count;
        return null;
    }

    @Override
    public void put(final int v) {
        count++;
        somma += v;
        if (somma >= 100)
            m.stop();
    }

    @Override
    public void reset() {
        somma = 0;
        count = 0;
    }
}
```

```
    }  
}  
  
/** Nomi e' un ConcreteColleague */  
public class Nomi implements Elemento {  
    private final Central m;  
    private final String[] seq = { "Monica", "Erica", "Rita", "Tina", "Sandra",  
    private final String[] nom = new String[3];  
    private final int iniz = 2;  
    private int i = 0;  
  
    public Nomi(final Central c) {  
        m = c;  
        System.out.println("Indovina i successivi tre nomi: " + seq[0] + " " + s  
    }  
  
    @Override  
    public void put(final String msg) {  
        if (i < 3) {  
            nom[i] = msg;  
            i++;  
        }  
        if (i == 3)  
            doWork();  
    }  
  
    @Override  
    public String doWork() {  
        System.out.println("Hai inserito: " + nom[0] + " " + nom[1] + " " + r  
        if (seq[iniz].equals(nom[0]) && seq[iniz + 1].equals(nom[1]) && seq[iniz  
            System.out.println("Benissimo, i nomi sono corretti");  
            m.stop();  
        } else  
            System.out.println("Non hai indovinato i nomi, riprova");  
        i = 0;  
        return null;  
    }  
  
    @Override  
    public void put(final int v) {  
        if (i == 3)  
            System.out.println("Bravo! Hai indovinato la sequenza di nomi");  
        else  
            System.out.println("Non hai indovinato la sequenza di nomi");  
    }  
  
    @Override  
    public void reset() {  
    }  
}
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

/** KeyListen e' un ConcreteColleague */
public class KeyListener implements Elemento {
    private final Central m;
    private BufferedReader inp;

    private int count = 0;

    public KeyListener(final Central c) {
        m = c;
        System.out.println("Scrivi un numero o una lettera e premi invio");
        System.out.println("Per uscire scrivi q");
    }

    @Override
    public String doWork() {
        try {
            inp = new BufferedReader(new InputStreamReader(System.in));
            String str = inp.readLine();
            count++;
            if (count == 10)
                m.stop();
            return str;
        } catch (final IOException ioe) {
            System.err.println(ioe);
        }
        return null;
    }

    @Override
    public void put(final String msg) {
        System.out.println("Hai inserito " + count + " input");
    }

    @Override
    public void put(final int v) {

    }

    @Override
    public void reset() {
    }
}

/** Central e' un Mediator */
public class Central {
    private Elemento s = new Sommatore(this);
}
```

```
private Elemento k = new KeyListener(this);
private Elemento n = new Numeri(this);
private Elemento nm = new Nomi(this);

private boolean active = true;

public void init() {
    String s;
    do {
        s = k.doWork();
        if (s.equals("q"))
            break;
        alert(s);
    } while (active);
}

public void alert(String str) {
    try {
        int v = Integer.parseInt(str);
        s.put(v);
        n.put(v);
    } catch (NumberFormatException nfe) {
        n.reset();
        s.reset();
        nm.put(str);
    }
}

public void stop() {
    active = false;
    n.put("Numeri:");
    s.put("Somma:");
    nm.put(0);
}
}

public class MainDialog {
    public static void main(final String[] args) {
        final Central c = new Central();
        c.init();
    }
}
```