

**Corso di Laurea in Ingegneria Informatica e Automatica**  
**A. A. 2017/2018**

# **Programmazione orientata agli oggetti**

**Paolo Liberatore, Massimo Mecella,  
Daniele Sora**

## **Esercitazione 6**

### **GUI in Java Swing**



Lo scopo di questa esercitazione è la realizzazione in Java di un'applicazione la cui interfaccia grafica è qui mostrata

The image shows a Java Swing window titled "Pannello Digitale". The window contains a digital panel interface. On the left, there is a numeric keypad with buttons for digits 7, 8, 9, 4, 5, 6, 1, 2, 3, 0, and a "Mostra" button. Above the keypad is a small rectangular display area. To the right of the keypad, the text "Codice digitato:" is displayed above a large rectangular input field.

7	8	9
4	5	6
1	2	3
C	0	Mostra

La finestra principale è organizzata in due pannelli (`JPanel`) disposti secondo il layout *FlowLayout*. Il primo pannello (a sinistra) è costituito da due componenti collocati utilizzando le politiche del *BorderLayout*. Il componente posizionato a nord è costituito da un `JPasswordField` che mostra il testo introdotto sostituendo ogni carattere con un asterisco. Il componente centrale è costituito da un pannello, strutturato secondo il layout *GridLayout* con 4 righe e 3 colonne, che contiene i 12 bottoni come in figura.

Il secondo pannello (a destra), i cui elementi sono disposti secondo *BorderLayout*, contiene due componenti: a nord una `JLabel` con il testo in figura, ed al centro una `JTextArea` (nota: Il bordo intorno all'area di testo (o altro componente/pannello) pu' essere impostato usando il metodo `setBorder`. Ad esempio:

```
setBorder(BorderFactory.createLineBorder(Color.black))
```

crea un bordo di colore nero intorno al componente/pannello oggetto d'invocazione.



L'utente interagisce con il programma premendo i pulsanti a sinistra. Ogni volta che un bottone corrispondente ad un numero viene premuto, il numero corrispondente viene inserito in coda nel campo password e mostrato come un asterisco. Non deve essere possibile modificare direttamente il testo nè nel `JPasswordField` nè nella `JTextArea` (nota: per fare ciò occorre impostare la proprietà *Editable* dei componenti al valore *false*).

Il bottone *Mostra* ha l'unico scopo di copiare la sequenza inserita, finora visualizzata solo come sequenza di asterischi, nell'area di testo a destra. Il bottone *C* serve a svuotare l'area di testo ed il campo password per inserire un nuovo codice segreto. Per evitare di cancellare erroneamente una codice inserito, dopo aver premuto il bottone *C* e prima dell'effettiva eliminazione, si può prevedere una finestra di conferma SI/NO. Se l'utente preme NO, l'operazione di cancellazione viene abortita.



La finestra di conferma può essere realizzata utilizzando il metodo **statico** `showConfirmDialog` della classe `JOptionPane` come segue:

- `JOptionPane.showConfirmDialog(null, "Cancellare il codice inserito?", "Pannello digitale", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE)`

L'invocazione di questo metodo restituirà:

- la costante `JOptionPane.YES_OPTION`, se viene premuto **SI**;
- la costante `JOptionPane.NO_OPTION` se viene premuto **NO**.

