

(A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti:

- Layout manager
- FlowLayout**
- BorderLayout**
- GridLayout**

(B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

B1) Conoscenza

- Cosa è un *LM*?
- Quali sono gli *LM* più comuni?
- Come funziona un *layout dei controlli*?

B2) Competenza

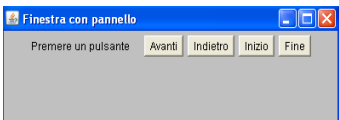
- Quali potrebbero essere i criteri che indirizzano ad usare un *LM* piuttosto che un altro?
- Quali sono le caratteristiche del *LM FlowLayout*?
- Quali sono le caratteristiche del *LM BorderLayout*?
- Quali sono le caratteristiche del *LM GridLayout*?

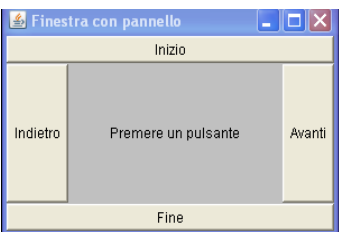
(C) ESERCIZI DI COMPRENSIONE


- Utilizzando l'help di Java, descrivere uso, attributi e metodi dei gestori dei layout **FlowLayout**, **BorderLayout** e **GridLayout**.
- La disposizione dei componenti in una interfaccia grafica si dice ed è consentita da strumenti Java detti, in inglese Normalmente, al momento della visualizzazione di un contenitore, Java dispone le componenti utilizzando un di default che può essere modificato.
- Java consente di impostare il attraverso classi predisposte, di cui le più comuni sono, e Un gestore di è un oggetto che viene creato usando il della classe relativa.
- Il che posiziona gli oggetti da sinistra a destra e dall'alto verso il basso si dice
- Associare a ciascun tipo di layout descritto, la classe associata, indicando nella casella la lettera corrispondente.

- | | | | |
|---|--------------------------|--------------------------------|-----------------------|
| 1 | <input type="checkbox"/> | Layout distinto in 5 regioni | A FlowLayout |
| 2 | <input type="checkbox"/> | Layout a forma di celle | B BorderLayout |
| 3 | <input type="checkbox"/> | Layout di default per finestre | C GridLayout |

- Indicando con *p* il nome di un pannello, per ciascuno dei *layout* mostrati, scrivere l'istruzione per inserirlo in *p*.

a)	Impostazione del layout
	

b)	Impostazione del layout
	

c)	Impostazione del layout
	

7. Per ciascuno degli esercizi seguenti, completare le parti mancanti, correggere eventuali errori, determinare gli output prodotti e dare una breve descrizione dell'applicazione.

- a. **import java.awt.*;**
class TestBorderLayout3 extends Frame
{
 private static final int ORR = 200;
 private static final int VER = 100;
 public TestBorderLayout3()
 {
 this.setLayout (.....);
 setSize(ORR,VER);
 setLocation(.....,);
 setTitle("Border Layout");
 this.....(new Button("Primo"),BorderLayout.....);
 this.....(new Button("Secondo"),BorderLayout.....);
 this.....(new Button("Terzo"),BorderLayout.....);
 this.....(new Button("Quarto"),BorderLayout.....);
 this.....(new Button("Quinto"),BorderLayout.....);
 }

 public static void main(String[] args)
 {
 TestBorderLayout3 frame = **new** TestBorderLayout3();
 frame.**setVisible**(true);
 }
} // end class
- b. **import**;
public class TestFlowLayout1 extends Panel
{
 public()
 {
 add(**new Button**("1"));
 add(**new Button**("2"));
 add(**new Button**("3"));
 add(**new Button**("4"));
 add(**new Button**("5"));
 add(**new Button**("6"));
 add(**new Button**("7"));
 add(**new Button**("8"));
 }

 public static void main(String[] argv)
 {
 Frame frame = **new Frame**("Flow");
 frame.**setSize**(400, 100);
 frame.**setLocation**(200, 200);
 Panel p = **new** TestFlowLayout1();
 frame.**add**(.....);
 frame.**setVisible**(true);
 }
}

(D) ESERCIZI DI APPLICAZIONE

1. Creare un'applicazione che simuli il cruscotto di un'auto mediante il LM **GridLayout**. I componenti presenti devono riguardare:
 - a. indicatori direzionali
 - b. stato del motore (acceso/spento)
 - c. stato luci di posizione (acceso/spento)
 - d. commutazione abbaglianti/anabbaglianti
 - e. spia livello carburante
 Porre i componenti in un *frame* con titolo "Quadro comandi".
2. Creare un'applicazione che tracci un'interfaccia grafica sul tipo di quella illustrata in **Fig. 1**, che simuli una pagina di invio testo (ad esempio una chat). Il pulsante *Invia* invia il testo, la label *Ready* indica che il sistema è pronto ad accettare il testo.
3. Creare un'interfaccia grafica che rappresenti una tastiera con una ottava, dal **Do** al **Si**.
4. Creare un'applicazione che tracci un'interfaccia grafica sul tipo di quella illustrata in **Fig. 2**, che simuli un quadro comandi di un macchinario, con indicazione di **Spento** o **Acceso** e con una pulsantiera con pulsanti numerati da 1 a 6.
5. Creare una finestra e disporre in essa:
 - a. un pannello contenente due pulsanti "Ok" e "Annulla";
 - b. due caselle di testo, precedute da due etichette identificate con "Cognome" e "Nome";
 - c. secondo il **BorderLayout**, 5 pulsanti, uno per ogni zona, con il nome della zona stessa;
 - d. un pannello di colore verde, contenente 3 pulsanti identificati da "Calcola", "Azzera" ed "Esci".
6. Progettare un'interfaccia di un'applicazione *Editor.java*, che gestisca un editor di testo. Creare una finestra e disporre in essa un menu operativo e gli opportuni controlli.

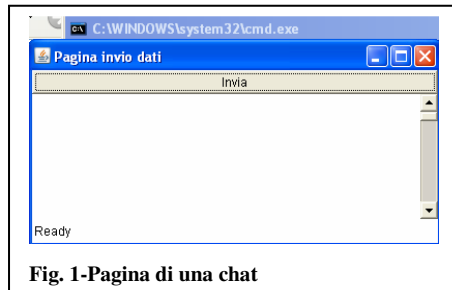


Fig. 1-Pagina di una chat

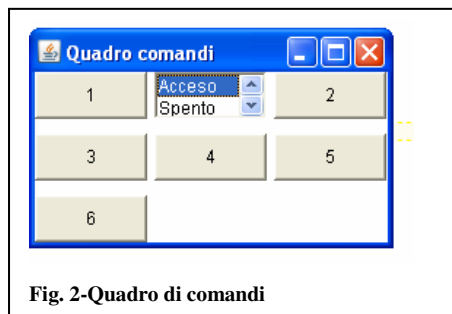


Fig. 2-Quadro di comandi

FINE ESERCITAZIONE

(E) ESERCITAZIONI PRATICHE
Esercitazione n. 1

Problema:

Obiettivi: utilizzare layout, controlli e frame.

1. Utilizzare come schema di lavoro il **Blocco note** del sistema operativo in uso e progettare una interfaccia di analogo tipo.
2. Realizzare il disegno su carta completo di dimensioni, menu e controlli
3. Usare JTextArea per realizzare l'area del testo