Gli eventi in Java Swing

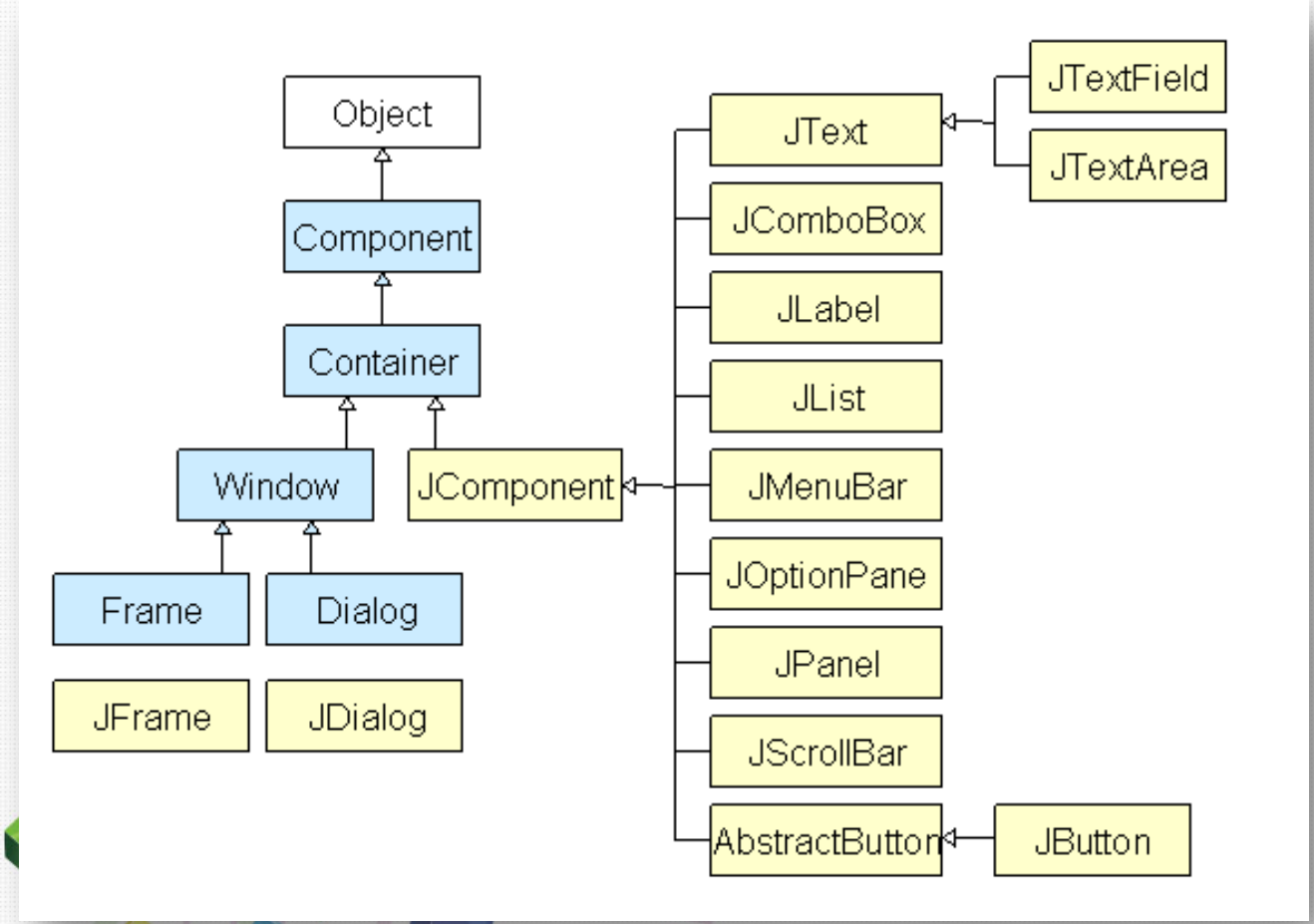
Ricordiamo: Interfaccia grafica!

Swing è un framework per Java per lo sviluppo di interfacce grafiche.

Creano GUI indipendenti dal Sistema Operativo.

I componenti Swing sono pure-Java quindi personalizzabili da parte del programmatore!

Gerarchia swing

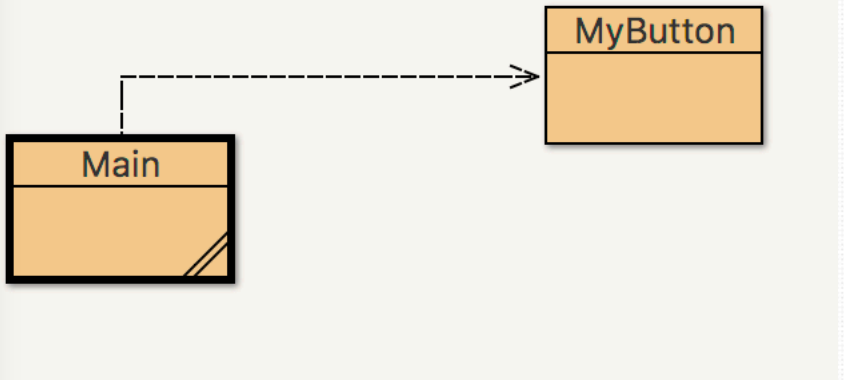


https://courses.cs.washington.edu/courses/cse331/10sp/sections/section7-src/swing.gif

Main che chiama MyButton

```java
import javax.swing.*;

public class MyButton extends JFrame
{
    private JPanel p = null;
    private JButton b = null;

    MyButton(){
        init();
    }

    private void init(){
        setSize(100,100);
        p = new JPanel();
        b = new JButton("Clicca qui!");
        p.add(b);
        add(p);
    }

}
```

```java
public class Main
{
    // instance variables - replace th
    static public void main()
    {
        MyButton myb = new MyButton()
        myb.setVisible(true);
    }
}
```

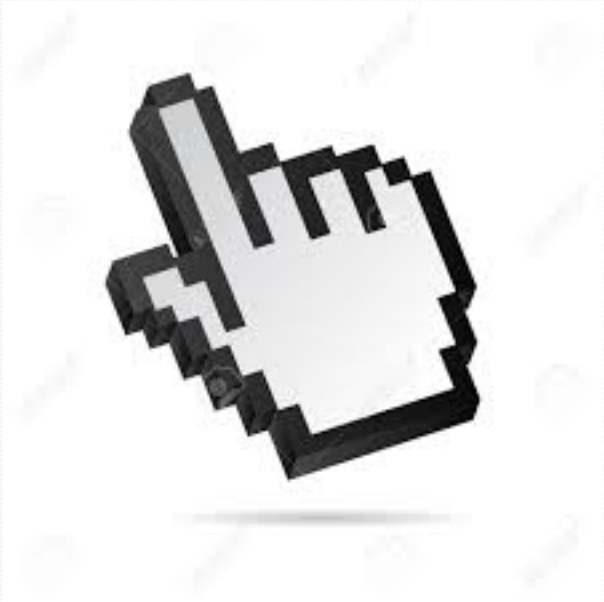


Eventi

- Avendo una interfaccia grafica, ci dovremo preoccupare di "intercettare" vari EVENTI generati dall'utente:
  - **Movimenti del mouse**
  - **Azioni, tipo click del mouse**
  - **Input da tastiera o cambiamento di focus**

Eventi e chi aspetta gli eventi...

- Abbiamo bisogno di oggetti che siano in "ascolto" degli eventi che possano capitare sulla interfaccia grafica
- Ci sono oggetti di tipo **Event** e oggetti di tipo **Listner**



ActionListner

- La classe **ActionListner** è in ascolto sui click di un Jbutton
- Creare una piccola classe che "implementa" ActionListner, si comporta come un listner… Non è un listner
- Il concetto di "interfaccia" in Java serve per costringere le classi che la implementano a fare l'override dei suoi metodi,

```
import javax.swing.*;
import java.awt.event.*;

public class MyButton extends JFrame
{
    private JPanel p = null;
    private JButton b = null;

    MyButton(){
        init();
    }

    private void init(){
        setSize(100,100);
        p = new JPanel();
        b = new JButton("Clicca qui!");
        b.addActionListener(new MyButtonListner());
        p.add(b);
        add(p);
    }
}

class MyButtonListner implements  ActionListener {
    public void actionPerformed(ActionEvent event)
    {
        JOptionPane.showMessageDialog(null,
            "Funziona!",
            "primo messaggio",
            JOptionPane.WARNING_MESSAGE);
    }
}
```

Aggiungere import java.awt.event.*;

Aggiungere MyButtonListner sul nostro bottone

Classe MyButtonListner che implementa un ActionListner

Nuova Classe per visualizzare un piccolo dialogo

Esercitazione

Con un **GridLayout** creare una interfaccia per una calcolatrice, tutti **Jbutton** a parte il testo dove scrivere e visualizzare i risultati: