



# Grafica ed eventi

Più movimento alle classi



# ActionListener

- Listener: è una classe che implementa ActionListener e definisce actionPerformed(...)
- Per aggiungere un ActionListener ad un oggetto c
  - c.addActionListener(listener);
- Esempio:

```
 JButton b=new JButton("mio pulsante");  
 ActionListener l=new ...  
 b.addActionListener(l);
```



# JComponent e Repaint

- Come chiedere il ridisegno di un componente?
  - Ogni componente Swing ha il metodo `repaint()`:  
l'invocazione di questo metodo schedula il ridisegno del componente "quanto prima"
  - ESEMPIO:  

```
JPanel p=new JPanel();  
p.add(...)  
...  
p.repaint();
```
- Il `repaint` comporta la chiamata a metodo `paintComponent(Graphics g)`
  - Utile nei quando si usano componenti per le animazioni



# JFrame

- Come strutturare le classi che utilizzano le swing?
  - quando si ha bisogno di un JFrame, creare una classe che estende JFrame e impostare nel costruttore tutte le istruzioni di inizializzazione
  - Tutti i JComponent che devono essere accessibili vengono definiti come campi e inizializzati nel costruttore

## Esempio

```
public class Myframe extends JFrame{
    private JButton b;
    public Myframe(){
        b=new JButton("pulsante");
        ...
    }
    public void mioMetodo(){
        b.setForeground(Color.RED);
    }
}
```



# Listener e accesso alla classe

- Permettere al listener di accedere a campi e metodi della classe generale (in genere si estende JFrame ma per le animazioni a volte si usa JComponent)
  - Definire i listener come classi interne della classe

## ESEMPIO:

```
public class Myframe extends JFrame{
    private JButton b;
    public Myframe(){
        b=new JButton("pulsante");
        b.addActionListener(new MioListener());
    }
    private class MioListener implements ActionListener{
        public void actionPerformed(ActionEvent e){
            b.setBackground(Color.GREEN);
        }
    }
}
```



# Listener e Adapter

## ■ Adapter

- Di ogni tipo di listener esiste un Adapter, cioè una classe che ha una implementazione vuota di tutti i metodi del listener.

## ESEMPIO:

### □ **MouseAdapter**

- MouseListener
- MouseMotionListener
- MouseWheelListener
- EventListener

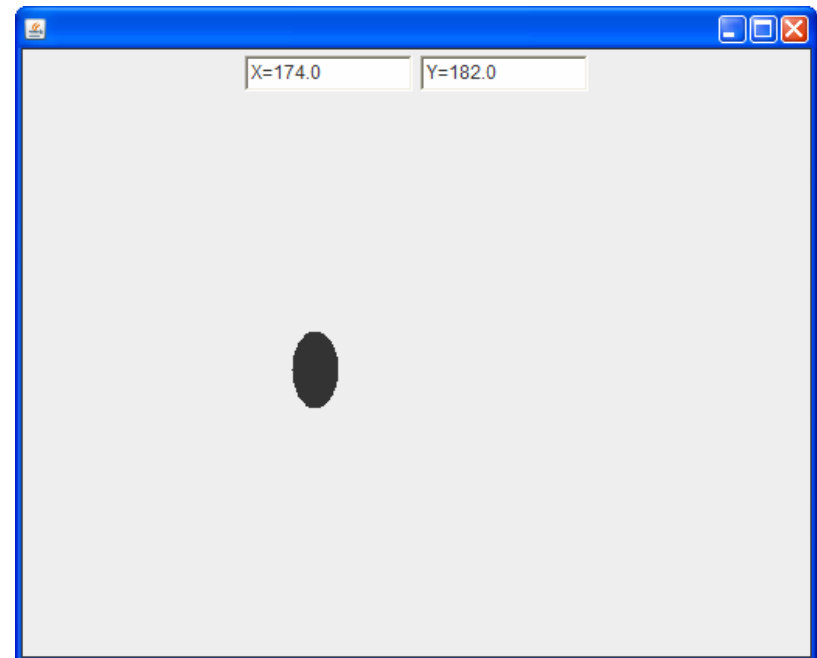


# Analisi degli eventi passo passo

- Definire quale è il componente principale della applicazione (JFrame, JComponent etc) e creare una classe che estende l'opportuna classe
  - Definire quali componenti interni occorrono (JButton etc)- se necessario definiti come variabili di istanza per accedere a questi in altri metodi
  - “Agganciare” i listener definiti ai componenti
- Stabilire quali sono gli eventi che interessano e quindi i tipi di listener utilizzare
- Definire in quali aree vengono generati gli eventi cioè a quali componenti “agganciare” i listener
- Definire le classi dei Listener necessari come classi interne che possono accedere alla classe generale

# Esercizio n.1

- Realizzare una applicazione che permette il trascinamento di una ellissi all'interno di un frame
  - Click all'interno della ellissi per iniziare il trascinamento
  - Movimento del mouse con pulsante per cliccato per spostare l'ellissi
  - Visualizzazione della posizione x,y







# Esercizio n.1

## ■ Osservazioni:

- Occorre differenziare il click iniziale dal trascinamento
  - Capire quali eventi occorrono
- La pallina si deve muovere quando il click iniziale è stato fatto all'interno della stessa
  - Come fare a capire se  $x,y$  appartengono ad un cerchio?
- Quando disegno la pallina in movimento devo fare attenzione al delta tra l'angolo a sinistra e il punto in cui è trascinata dal mouse



## Esercizio n.2

- Aggiungere al pannello precedente 2 pulsanti con le etichette “Rosso” e “Verde”
  - La pressione del primo imposta il colore **Rosso** alla pallina, mentre il secondo usa il **Verde**
- Aggiungere una JSpinner che imposta l'altezza della pallina in base al valore specificato

## Esercizio n.3 per casa

- A partire dall'esercizio originale, aggiungere una seconda pallina rossa che possa essere spostata in maniera indipendente dalla prima

