



# Programmazione III

Prof.ssa Liliana Ardissono  
Dipartimento di Informatica  
Università di Torino

**Interfacce Utente Grafiche (GUI) – parte 2**  
**(basi con Java SWING)**

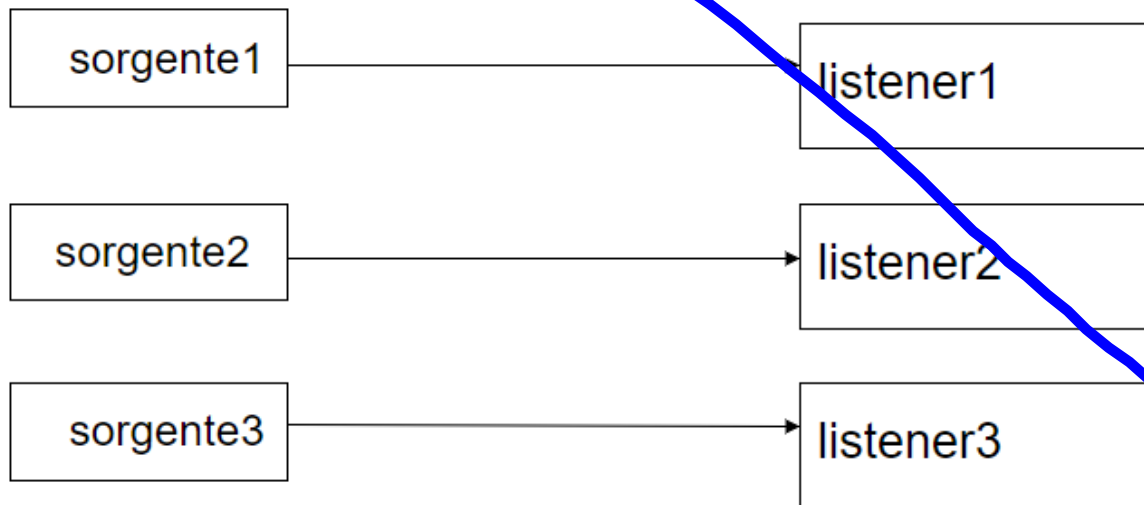


Questa presentazione è distribuita sotto licenza Creative Commons CC BY ND



# Schemi di uso dei Listener - I

In molti casi, ogni componente grafico (widget) ha il suo listener dedicato. Per esempio, in ButtonTest (dal libro «Core Java»), posso associare un listener specifico a ogni bottone per cambiare il colore dello sfondo del pannello:

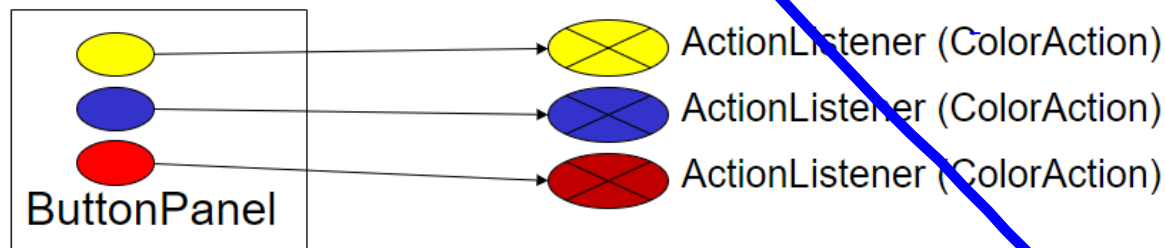


# Esempio (schema 1) – ButtonTest



```
class ButtonPanel extends JPanel {  
    public ButtonPanel() {  
        JButton yellowButton = new JButton("Yellow");  
        JButton blueButton = new JButton("Blue");  
        JButton redButton = new JButton("Red");  
  
        add(yellowButton); add(blueButton); add(redButton);  
        .....  
    }  
}
```

A parte il colore, tutti e tre gli ActionListener devono fare lo stesso tipo di operazione (cambiare il colore dello sfondo) → in una prima implementazione noi possiamo definire la classe ActionListener (ColorAction) e crearne tre istanze, una per bottone, a ciascuna delle quali passiamo il colore da usare.



# ButtonTest – I



```
class ColorAction implements ActionListener {  
    private Color backgroundColor;  
    private ButtonPanel bp;  
  
    public ColorAction(Color c, ButtonPanel p) {  
        backgroundColor = c;  
        bp = p;  
    }  
  
    public void actionPerformed(ActionEvent event) {  
        bp.setBackground(backgroundColor);  
        // il metodo setBackground() di JPanel cambia  
        // il colore dello sfondo  
    }  
}
```

# ButtonTest – II



```
public ButtonPanel() {  
    JButton yellowButton = new JButton("Yellow");  
    JButton blueButton = new JButton("Blue");  
    JButton redButton = new JButton("Red");  
    add(yellowButton); add(blueButton); add(redButton);  
  
    // il ButtonPanel deve passare se stesso come riferimento al  
    // listener ColorAction per permettere la modifica del proprio colore.  
    ColorAction yellowAction = new ColorAction(Color.YELLOW, this);  
    ColorAction blueAction = new ColorAction(Color.BLUE, this);  
    ColorAction redAction = new ColorAction(Color.RED, this);  
  
    yellowButton.addActionListener(yellowAction);  
    blueButton.addActionListener(blueAction);  
    redButton.addActionListener(redAction);  
}
```

# Miglioriamo il codice: Listener come classi interne degli oggetti grafici



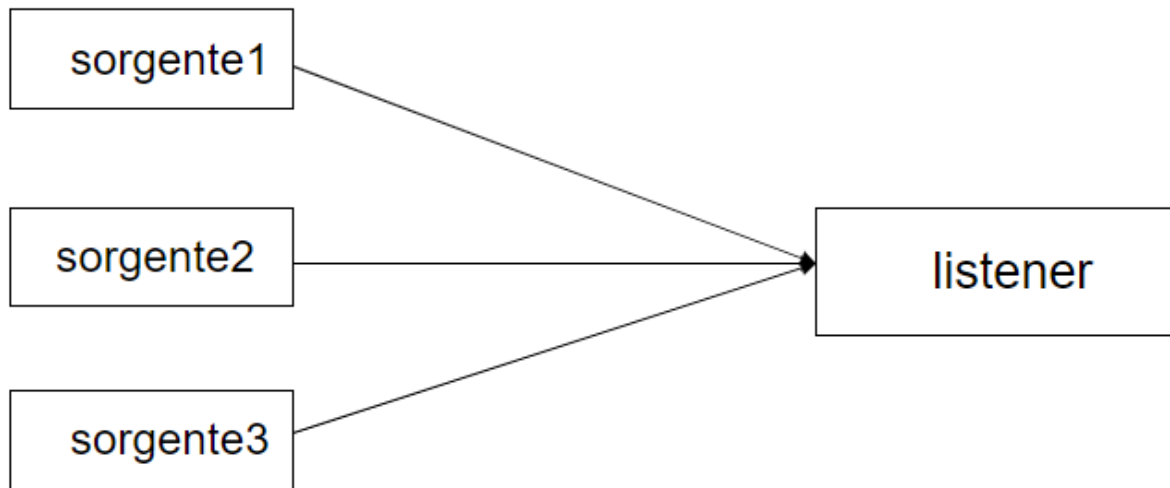
Se **ColorAction** è una **classe interna** a **ButtonPanel**, gli oggetti **ColorAction** possono accedere ai campi e metodi, anche privati, di **ButtonPanel**:

```
class ButtonPanel extends JPanel {  
    ....  
    private class ColorAction implements ActionListener {  
        private Color backgroundColor;  
        public ColorAction(Color c) { // non devo più passare il pannello  
            backgroundColor = c;  
        }  
        public void actionPerformed(ActionEvent event) {  
            setBackground(backgroundColor);  
        }  
    }  
    ...  
}
```

# Schemi di uso dei Listener - II



Può essere comodo definire un solo listener che gestisca gli eventi di più **sorgenti di eventi omogenee**. Per esempio, il pannello potrebbe implementare il listener degli eventi generati dai 3 bottoni, distinguendo la sorgente degli eventi attraverso il suo nome:





# Esempio (schema 2) – ButtonPanel

```
class ButtonPanel extends JPanel implements ActionListener {  
    public ButtonPanel() {  
        JButton yellowButton = new JButton("Yellow");  
        JButton blueButton = new JButton("Blue");  
        JButton redButton = new JButton("Red");  
        yellowButton.addActionListener(this);  
        blueButton.addActionListener(this);  
        redButton.addActionListener(this);  
    }  
  
    public void actionPerformed(ActionEvent event) {  
        String command = event.getActionCommand();  
        if (command.equals("Yellow")) setBackground(Color.YELLOW);  
        else if (command.equals("Blue")) setBackground(Color.BLUE);  
        else if (command.equals("Red")) setBackground(Color.RED);  
    }  
}
```

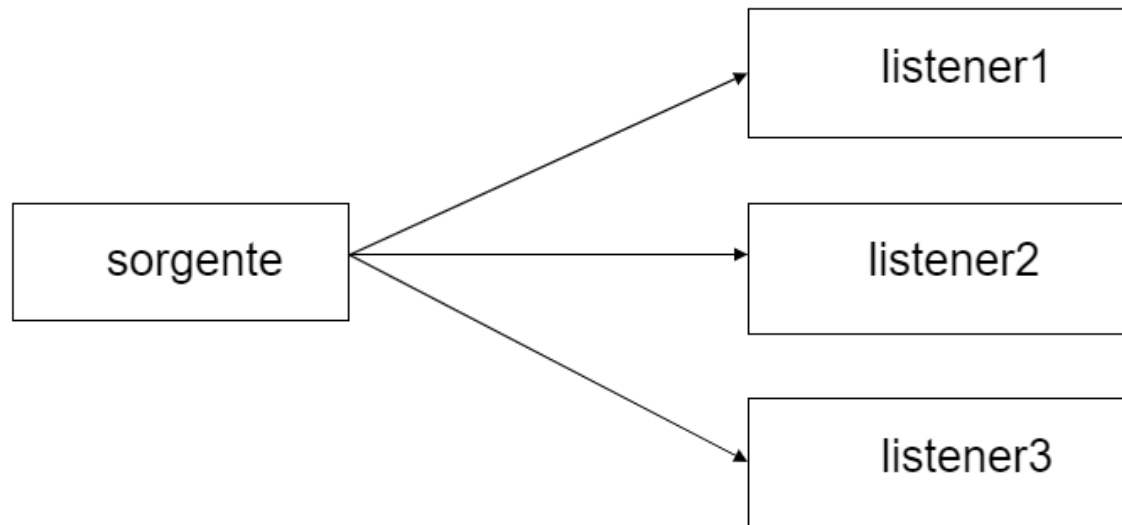


# Schemi di uso dei Listener - III



Talvolta è necessario associare più di un listener alla stessa sorgente di eventi:

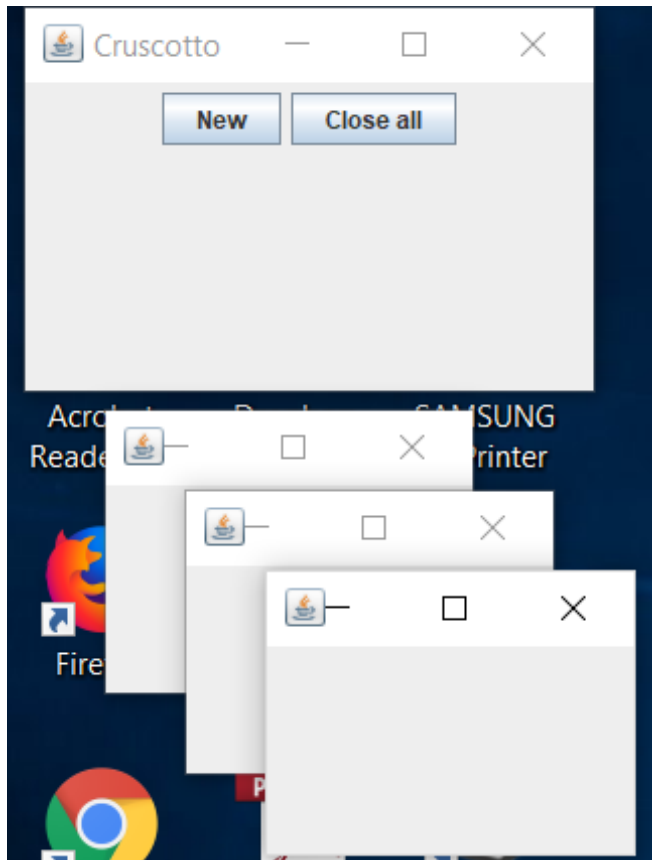
- per gestire diverse tipologie di evento generate dalla sorgente.
- per gestire in più modi gli stessi eventi (fare più cose), in parallelo.



# Esempio (schema 3) – MultiCastPanel



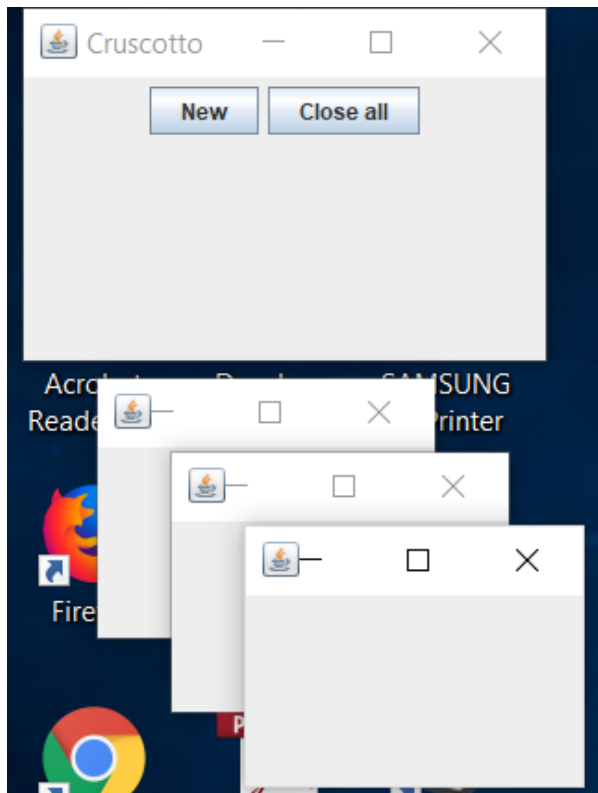
- il pulsante **New** genera una nuova finestra ogni volta che viene premuto.
- il pulsante **Close all** chiude tutte le finestre, con un solo click.



# MultiCastPanel - I



- **Un solo ActionListener ascolta gli eventi di «New»** e crea una finestra per ciascun click del bottone.
- **Ogni finestra è un diverso ActionListener di «Close all»** e quando riceve l'evento di click del bottone si chiude.



# MultiCastPanel - II



```
class MultiCastPanel extends JPanel {  
  
    public MultiCastPanel() {  
        JButton newButton = new JButton("New");  
        add(newButton);  
        final JButton closeAllButton = new JButton("Close all");  
        add(closeAllButton);  
  
        newButton.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent event) {  
                BlankFrame fr = new BlankFrame(closeAllButton);  
                fr.setVisible(true); }  
        });  
    }  
}
```

# MultiCastPanel - III



```
class BlankFrame extends JFrame {  
    public BlankFrame(final JButton closeButton) {  
        ActionListener closeListener = new ActionListener() {  
            public void actionPerformed(ActionEvent ev) {  
                // rimuove il listener prima di distruggere la finestra  
                closeButton.removeActionListener(closeListener);  
                dispose();  
            }  
        };  
        closeButton.addActionListener(closeListener);  
    }  
}
```



# Ringraziamenti

Grazie al Prof. Emerito Alberto Martelli del Dipartimento di Informatica dell'Università di Torino per aver redatto la prima versione di queste slides.