Programmazione in Java
e
gestione della grafica

Lezione 21

Parliamo (ancora) di …

Eventi della GUI

(Eventi del mouse e della tastiera)

Gestione degli eventi del mouse

- Eventi del Mouse
  - Creo un oggetto `MouseEvent`
  - Gestione con `MouseListeners` e `MouseMotionListeners`
  - `MouseInputListener` combina le due interfacce
  - Interfaccia `MouseWheelListener` dichiara un metodo `mouseWheelMoved` che riceve parametro `MouseWheelEvents` (sottoclasse di `MouseEvent`.

MouseListener and MouseMotionListener interface methods

Methods of interface `MouseListener`

`public void mousePressed( MouseEvent event )`

Called when a mouse button is pressed while the mouse cursor is on a component.

`public void mouseClicked( MouseEvent event )`

Called when a mouse button is pressed and released while the mouse cursor remains stationary on a component. This event is always preceded by a call to `mousePressed`.

`public void mouseReleased( MouseEvent event )`

Called when a mouse button is released after being pressed. This event is always preceded by a call to `mousePressed` and one or more calls to `mouseDragged`.

`public void mouseEntered( MouseEvent event )`

Called when the mouse cursor enters the bounds of a component.

Metodi delle interfacce `MouseListener` and `MouseMotionListener`

MouseListener and MouseMotionListener interface methods

`public void mouseExited( MouseEvent event )`

Called when the mouse cursor leaves the bounds of a component.

Methods of interface `MouseMotionListener`

`public void mouseDragged( MouseEvent event )`

Called when the mouse button is pressed while the mouse cursor is on a component and the mouse is moved while the mouse button remains pressed. This event is always preceded by a call to `mousePressed`. All drag events are sent to the component on which the user began to drag the mouse.

`public void mouseMoved( MouseEvent event )`

Called when the mouse is moved when the mouse cursor is on a component. All move events are sent to the component over which the mouse is currently positioned.

Metodi delle interfacce `MouseListener` and `MouseMotionListener`

- Le chiamate di metodo `mouseDragged` and `mouseReleased` sono passate a `MouseMotionListener` per la `Component` sul quale si è iniziata una operazione di trascinamento del mouse.
- La chiamata di metodo `mouseReleased` alla fine di una operazione di trascinamento è passata al `MouseListener` per la `Component` sul quale si è iniziata una operazione di trascinamento del mouse.

Vediamo file `mousetrackerframe.java`

Classi Adapter

- Molte interfacce event listener contengono moltissimi metodi e non sempre è pratico implementarli tutti (es. `WindowListener` ha 7 metodi per la gestione delle finestre!)
- Per risolvere questo, nel package `java.awt.event` e `java.swing.event` ci sono delle classi adapter

Classi Adapter per event listener

- Implementano interfacce event listener
- Forniscono implementazioni di default (con il corpo del metodo vuoto) per tutti i metodi di gestione degli eventi dell'interfaccia corrispondente.
- Si può estendere una classe adapter ed estendere solo I metodi che servono nel nostro caso.

Event-adapter class in `java.awt.event`	Implements interface
`ComponentAdapter`	`ComponentListener`
`ContainerAdapter`	`ContainerListener`
`FocusAdapter`	`FocusListener`
`KeyAdapter`	`KeyListener`
`MouseAdapter`	`MouseListener`
`MouseMotionAdapter`	`MouseMotionListener`
`WindowAdapter`	`WindowListener`

Classi Event-adapter e interfacce implementate del package `java.awt.event`.

Estensione di `MouseAdapter`

`MouseAdapter`

È una classe adapter per le interfacce `MouseListener` e `MouseMotionListener`

– se estendiamo questa classe possiamo fare overriding *solo* dei metodi che ci servono

Vediamo file `MouseDetailsFrame.java`

Esempio: una sottoclasse `JPanel` per disegnare con il mouse

- Fare overriding nella classe `JPanel`
  - Creiamo un'area dedicata al disegno
  - Usiamo il mouse per disegnare
  - Classe `Graphics` è usata per disegnare su una componente
  - Classe `Point` rappresenta una coppia *x*-*y* di coordinate

Metodo `paintComponent`

- Disegna su una componente Swing (`Jcomponent`)
- Fare overriding del metodo ci permette di creare disegni “nostri”
- Quando faccio overriding devo prima chimamare i metodi della superclasse

Vediamo file `PaintPanel.java`

Gestione degli eventi della tastiera

- Uso interfaccia `KeyListener`
- Devo dichiarare i metodi
  - `keyPressed`
  - `keyReleased`
  - `keyTyped`

Vediamo file `KeyDemoFrame.java`

`TextAreaFrame.java`