

GUI in Java con l'AWT – 2

Stefano Mizzaro

Dipartimento di matematica e informatica
 Università di Udine
<http://www.dimi.uniud.it/mizzaro>
 mizzaro@dimi.uniud.it
 Programmazione, lezione 24
 22 febbraio 2006

Riassunto

- GUI in Java: AWT
- Concetti:
 - Componenti
 - Eventi
 - Ascoltatori
 - (Layout)
- Documentazione
 - Libri, Fotocopie
 - (Tutorial Sun)

Stefano Mizzaro - AWT2 2

Cosa manca

- Associazioni componenti – eventi
- Gestione layout
- Campionario componenti
 - (Component, Container, Window)
 - Frame, Panel, Button, Label, TextComponent, TextArea, TextField, (Canvas)
 - Per ogni componente:
 - Cosa fa/è, quali eventi genera, esempi di uso
 - Classi "ausiliarie": Color, (Font), Dimension
- (Disegno: Canvas e Paint. Applet)
- Eh...

Stefano Mizzaro - AWT2 3

Scaletta

- Componenti ↔ eventi
- Layout
- Campionario di componenti
- Progetto

Stefano Mizzaro - AWT2 4

Componenti-eventi

- Quali componenti generano quali eventi?
- Ogni componente può generare più eventi
- Ogni evento può essere generato da più componenti

Stefano Mizzaro - AWT2 5

Componenti, eventi ed eredità

- Eredità fra i componenti (Component genera eventi key ⇒ tutti i componenti lo faranno)
- Eredità fra gli eventi (se un componente genera eventi key ⇒ genera anche i "sopraeventi")

Stefano Mizzaro - AWT2 6

Componenti – eventi (1/2)

	action	adjustment	component	container	focus	item	key	mouse	mouse motion	text	window
Button	•		•		•	•	•	•			
Canvas			•		•	•	•	•			
Checkbox			•		•	•	•	•			
CheckboxMenuItem						•					
Choice			•		•	•	•	•			
Component			•		•	•	•	•			
Container			•	•	•	•	•	•			
Dialog			•	•	•	•	•	•			•

Componenti – eventi (2/2)

	action	adjustment	component	container	focus	item	key	Mouse	mouse motion	text	window
Frame			•	•	•		•	•	•		•
Label			•		•		•	•	•		
List	•		•		•	•	•	•	•		
MenuItem	•										
Panel			•	•	•		•	•	•		
Scrollbar		•	•		•		•	•	•		
TextArea			•		•		•	•	•	•	
TextField	•		•		•		•	•	•	•	
Window			•	•	•		•	•	•		•

Componenti - Eventi - Ascoltatori

- L'evento **XxxEvent** è gestito da **XxxListener**
- **XxxListener** va registrato nel componente
 - **addXxxListener** sono definiti nelle (ed ereditati da) classi "opportune":
 - **addWindowListener** è definito in **Window** ed ereditato da **Frame** e **Dialog**
 - **addActionListener** è definito in **Button**, **List**, **MenuItem**, **TextField**
- ...vedi fotocopie...

Stefano Mizzaro - AWT2 9

Scaletta

- Componenti ↔ eventi
- Layout
- Campionario di componenti
- Progetto

Stefano Mizzaro - AWT2 10

I gestori di layout

- Posizionamento dei componenti in un contenitore
- Ci sono vari gestori di layout (implementano **LayoutManager**)
- **setLayout(LayoutManager)** in **Container**
 - E quindi in tutte le sottoclassi...
 - Vediamone due

Stefano Mizzaro - AWT2 11

BorderLayout

- 5 posizioni (N, S, E, W, C)
 - **BorderLayout.NORTH**, ...
- Il **Center** cerca di essere più grande possibile
- I componenti vengono "stirati"
- Default di **Frame**
- Vediamo la documentazione
- Vediamo un esempio (**ProvaLayout.java**)

Stefano Mizzaro - AWT2 12

FlowLayout

- Come le parole di un testo
- Centrato, ma si può cambiare (`FlowLayout.LEFT, ...`)
- Default di `Panel`
- Vediamo la documentazione
- I componenti **non** vengono stirati (utile mettere un `Panel` in un `Frame...`)
- Vediamo un esempio (`ProvaFlowLayout.java`)

Stefano Mizzaro - AWT2 13

Esempio di uso dei layout

```

...
Panel p = new Panel();
p.setLayout(new BorderLayout());
p.add(new Label("Nord"), BorderLayout.NORTH);
p.add(new Label("Sud"), BorderLayout.SOUTH);
p.add(new Label("Est"), BorderLayout.EAST);
p.add(new Label("Ovest"), BorderLayout.WEST);
p.add(new Label("Centro"), BorderLayout.CENTER);
...

```

Stefano Mizzaro - AWT2 14

Pregi dei layout

- Meglio del posizionamento assoluto
- Il layout manager posiziona e dimensiona automaticamente (quando serve) i componenti in un contenitore
- **NON** serve invocare i metodi di un gestore di layout: vengono invocati automaticamente quando necessario

Stefano Mizzaro - AWT2 15

Scaletta

- Componenti ↔ eventi
- Layout
- Campionario di componenti
- Progetto

Stefano Mizzaro - AWT2 16

Tutti i componenti

<ul style="list-style-type: none"> ■ <code>Button</code> ■ <code>(Canvas)</code> ■ <code>Checkbox</code> ■ <code>CheckboxGroup</code> ■ <code>Choice</code> ■ <code>Dialog</code> ■ <code>Frame</code> ■ <code>Label</code> ■ <code>List</code> ■ <code>Panel</code> ■ <code>Scrollbar</code> ■ <code>ScrollPane</code> 	<ul style="list-style-type: none"> ■ <code>TextComponent</code> ■ <code>TextArea</code> ■ <code>TextField</code> ■ Componenti per menu: <ul style="list-style-type: none"> ■ <code>Menu</code> ■ <code>MenuItem</code> ■ <code>CheckboxMenuItem</code> ■ <code>PopupMenu</code> ■ <code>MenuBar</code> ■ <code>MenuShortcut</code>
---	---

Stefano Mizzaro - AWT2 17

abstract Component

- Il componente più generico (sopraclasse)
- Metodi principali:
 - `void setVisible(boolean)`
 - `void setBounds(int int int int)`
 - `Dimension getSize()`
 - `void setSize(Dimension d)`
 - `void setSize(int width, int height)`
 - `void setFont(Font)`
 - `void setBackground(Color) (Foreground)`
 - `Color getBackground() (Foreground)`
 - `void addXxxListener(XxxListener)`

Stefano Mizzaro - AWT2 18

3 classi "ausiliarie"

- **Color**
- **Dimension**
- **(Font)**

Stefano Mizzaro - AWT2 19

Color

- Ogni istanza rappresenta un colore
- Definisce alcuni colori "già pronti" (costanti `Color.BLACK`, `Color.WHITE`, ...)
- Vedi documentazione API
- Metodi: costruttori (sovraccarichi: RGB,...)

```
Component c;
c = new ...;
c.setBackground(new Color(0,0,255));
c.setBackground(Color.BLUE);
```

Stefano Mizzaro - AWT2 20

Dimension

- Larghezza (**width**) e altezza (**height**) di un **Component** in un unico oggetto
- Metodi:
 - Costruttore
 - `double getWidth()`
 - `double getHeight()`
- Vedi documentazione API

Stefano Mizzaro - AWT2 21

abstract Container

- Contenitore di componenti
- Metodi:
 - `void add(Component)`
 - `void setLayout(LayoutManager)`
 - `void addContainerListener(...)`

Stefano Mizzaro - AWT2 22

Window

- Finestra generica
- Di solito non si usa, si usano le sottoclassi
- Metodi
 - `void pack()`
 - `void dispose()`
 - `void addWindowListener(WindowEvent)`

Stefano Mizzaro - AWT2 23

Frame

- Finestra principale di un'applicazione
- Metodi principali:
 - Costruttore
 - `void setTitle(String)`
 - `void setMenuBar(MenuBar)`
- Uso tipico: già visto...

Stefano Mizzaro - AWT2 24

Panel

- Contenitore generico: usato per contenere altri componenti
- **Panel** dentro **Frame**
- **Panel** dentro **Panel**
- Metodi:
 - Costruttore:
 - `public Panel ()`
 - `public Panel (LayoutManager)`
- Layout di default: **FlowLayout**

Stefano Mizzaro - AWT2 25

Uso tipico di Panel

```
import java.awt.*;
class ProvaPanel extends Frame {
    public ProvaPanel () {
        super ("Layout");
        add(new Button("Nord"), BorderLayout.NORTH);
        add(new Button("Sud"), BorderLayout.SOUTH);
        add(new Button("Est"), BorderLayout.EAST);
        add(new Button("Ovest"), BorderLayout.WEST);
        Panel p = new Panel();
        p.add(new Button("Centro"));
        add(p, BorderLayout.CENTER);
        setVisible(true);
        pack();
    }
    ...
}
```

`add(new Button("Centro"), BorderLayout.CENTER);`

Stefano Mizzaro - AWT2 26

Button

- Pulsante che l'utente può premere
- Con etichetta testuale
- Quando premuto genera un **ActionEvent**
- Che va gestito con un **ActionListener**
- L'**ActionEvent** ha il metodo **getActionCommand** che restituisce una **String** che può essere usata per individuare il bottone
 - Anche se di solito meglio 1 pulsante 1 ascoltatore

Stefano Mizzaro - AWT2 27

Metodi di Button

- **setLabel** (anche con il costruttore)
- **getLabel**
- **setActionCommand** (di default la label)
- **getActionCommand**
- **addActionListener (ActionListener l)**
- ...

Stefano Mizzaro - AWT2 28

Uso tipico di Button (già visto)

```
class ... {
    ...
    Button b = new Button("Exit");
    this.add(b);
    Ascoltatore a = new Ascoltatore();
    b.addActionListener(a);
    ...
    class Ascoltatore implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            //... Cosa fare quando si preme il pulsante
            //qui ci può essere e.getActionCommand()
        }
    }
}
```

Stefano Mizzaro - AWT2 29

Label

- Etichetta, testo non selezionabile
- Presenta informazioni
- Genera solo gli eventi di **Component**
- Di solito non ascoltati

Stefano Mizzaro - AWT2 30

Metodi di `Label`

- `void setText(String)`
 - (anche costruttore)
- `void setAlignment(int)`
 - `Label.LEFT/CENTER/RIGHT` (anche costruttore)
- `String getText()`
- `int getAlignment()`

Stefano Mizzaro - AWT2 31

`TextComponent`

- Generico componente contenente testo
- Costruttore con accessibilità di package, quindi non istanziabile dalla ns. applicazione
- Genera eventi `TextEvent` gestiti con `TextListener (textValueChanged)`
- Si usano le due 2 sottoclassi (che ereditano):
 - `TextArea`
 - `TextField`

Stefano Mizzaro - AWT2 32

Metodi di `TextComponent`

- `void addTextListener(TextListener)`
- `String getText()`
- `void setText(String)`
- `void setEditable(boolean)`
- `boolean isEditable()`
- `String getSelectedText()`
- `void selectAll()`
- ...

Stefano Mizzaro - AWT2 33

`TextArea`

- Area di testo su più righe
- Genera gli stessi eventi di `TextComponent`

Stefano Mizzaro - AWT2 34

Metodi di `TextArea`

- Quelli di `TextComponent`
- Altri (vedi javadoc API):
 - Costruttori
 - `void append(String)`
 - `int getColumns()`
 - `int getRows()`
 - `void setColumns(int)`
 - `void setRows(int)`

Stefano Mizzaro - AWT2 35

Uso tipico di `TextArea`

```
class ... {
    TextArea t = new TextArea("Testo iniziale",5,20);
    ...
    class AscoltatoreTesto implements TextListener {
        public void textValueChanged(TextEvent e) {
            if(t.getText().equals("Basta"))
                t.setEditable(false);
        }
    }
    ...
    ... {
        t.addTextListener(new AscoltatoreTesto());
    }
}
```

- (Cfr. `ProvaTextArea.java`)
- Oppure `setText(...)` per visualizzare messaggi su un `TextArea` non editabile

Stefano Mizzaro - AWT2 36

TextField

- Campo di testo (1 riga)
- Oltre agli eventi di **TextComponent**
- Genera anche **ActionEvent** quando si preme **return**
 - E il **getActionCommand** restituisce il testo inserito

Stefano Mizzaro - AWT2 37

Uso tipico di TextField (1/2)

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame {
    private TextField f;
    private TextArea t;
    public MyFrame(){
        super("Prova Text");
        t = new TextArea("Hai inserito:\n",5,20);
        t.setEditable(false);
        f = new TextField("Inserisci testo");
        f.selectAll();
        f.addActionListener(new Ascoltatore());
        this.add(f, BorderLayout.NORTH);
        this.add(t, BorderLayout.CENTER);
        this.pack();
        this.setVisible(true);
    }
    ...
}
```

Stefano Mizzaro - AWT2 38

Uso tipico di TextField (2/2)

```
...
class Ascoltatore implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        t.append(e.getActionCommand() + "\n");
        f.selectAll();
    }
}

public class ProvaText {
    public static void main (String[] args) {
        MyFrame f = new MyFrame();
    }
}
```

- Nota: comodità della classe interna

Stefano Mizzaro - AWT2 39

Senza classe interna

```
class MyFrame {
    ...
    f.addActionListener(new Ascoltatore(f, t));
    ...
}

class Ascoltatore implements ActionListener {
    private TextField f;
    private TextArea t;
    public Ascoltatore (TextField f, TextArea t) {
        this.f = f; this.t = t;
    }
    public void actionPerformed(ActionEvent e) {
        t.append(e.getActionCommand() + "\n");
        f.selectAll();
    }
}
```

Stefano Mizzaro - AWT2 40

Altri componenti (che non vediamo...)

- Canvas
- Checkbox
- CheckboxGroup
- Choice
- Dialog
- List
- Scrollbar
- ScrollPane
- Componenti per menu
 - MenuComponent
 - MenuItem
 - CheckbocMenuItem
 - PopupMenu
 - MenuBar
 - MenuShortcut

Stefano Mizzaro - AWT2 41

Riassunto

- Componenti ↔ eventi
- Layout
- Campionario di componenti
 - (Component, Container, Window)
 - Frame, Panel, Button, Label, TextComponent, TextArea, TextField
 - Per ogni componente:
 - Cosa fa/è, quali eventi genera, esempi di uso
- Classi "ausiliarie"
 - Color, (Font), Dimension

Stefano Mizzaro - AWT2 42

Riassunto del corso

- Introduzione alla programmazione
 - Accento sulla qualità
- In dettaglio
 - La programmazione strutturata
 - Il linguaggio Java
- Solo accennati
 - Programmazione OO
 - API

Stefano Mizzaro - AWT2 43

Esame

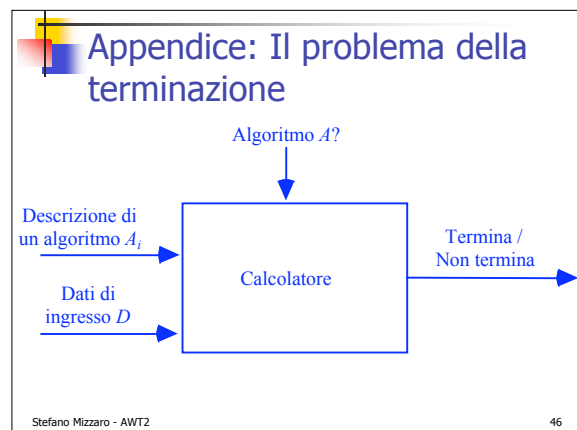
- Traccia esecuzione: condizione necessaria
- Scritto + progetto (facoltativo) + orale
- Voto max. senza progetto: 27
- Progetto dà incremento di 0-3 punti
 - Solo se voto scritto ≥ 21 !!
 - Se voto < 21 , non consegnato, non presentato \Rightarrow progetto annullato (come non fatto)
- Se voto scritto $\leq 10 \Rightarrow -5$ all'appello succ.!!
- Scelta obbligatoria fra 1o scritto e 2o compitino
 - **FATE ATTENZIONE QUANDO VI ISCRIVETE SU SINDY!!!!**

Stefano Mizzaro - AWT2 44

Progetto

- Cfr. descrizione sulle mie pag. Web
 - (o su quelle di Coppola)
 - Consegnato allo scritto E spedito per email
- Altri progetti per gli appelli successivi
- Orale collegato al progetto (non allo scritto)
 - Se consegnate il progetto e non vi presentate all'orale, il progetto viene annullato
- Dovete vedervi per conto vostro qcosa (di mooolto semplice) delle API

Stefano Mizzaro - AWT2 45



Una soluzione?

- Beh, potrei far eseguire A_i sui dati e attendere
- Se termina so che termina
- Se dopo 2 ore non ha terminato... so che non ha terminato in due ore... ma quanto devo aspettare?!
- Non va!
- Teorema: A non esiste

Stefano Mizzaro - AWT2 47

