

Applet (e altro dell'AWT...)

Stefano Mizzaro

Dipartimento di matematica e informatica
 Università di Udine
<http://www.dimi.uniud.it/mizzaro/>
 mizzaro@uniud.it
 Programmazione, lezione 22
 19 maggio 2015

Riassunto

- Programmazione strutturata
- OO: TDA, scambio messaggi, eredità, polimorfismo, OO in Java
- Rassegna API
 - Documentazione Javadoc delle API
 - Sorgenti delle API
 - **Object, String, Math, System**
 - Eccezioni
 - File
 - GUI in Java: AWT
 - Componenti, Eventi, Ascoltatori, Layout

Stefano Mizzaro - Applet

2

Scaletta

- Esame
- Applet
 - Esempi e restrizioni
 - Disegno dei componenti, metodo `paint()` e classe **Graphics**
 - Eventi del mouse
 - **Choice**
 - Menu

Stefano Mizzaro - Applet

3

Esame

- Scritto + autovalutazione + [progetto] + orale
 - [Progetto facoltativo. Ma consigliato]
 - Voto max. senza progetto: 27
 - Orale obbligatorio per tutti
 - Scritto valido per tutto l'anno
- Progetto dà incremento di 0 – 3 punti
 - Solo se voto scritto ≥ 21 !! (era: 24)
 - Se voto < 21 , non consegnato, non presentato \Rightarrow progetto annullato (come non fatto)
- + bonus autovalutazione: [-4, +4]
 - Scadenza: 19 giugno 2015
- Traccia esecuzione: condizione necessaria

Stefano Mizzaro - Applet

4

Progetto

- Progetto di gruppo (3 persone)
- Cfr. Blog del corso
 - Ca. 2 settimane prima dello scritto
 - Consegnato in forma cartacea 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
- Dovrete vedervi per conto vostro qcosa (di mooolto semplice) delle API
- Indicazioni più precise sulla descrizione del progetto

Stefano Mizzaro - Applet

5

Scaletta

- Esame
- Applet
 - Esempi e restrizioni
 - Disegno dei componenti, metodo `paint()` e classe **Graphics**
 - Eventi del mouse
 - **Choice**
 - Menu

Stefano Mizzaro - Applet

6

Gli applet

- Applet = "piccola applicazione"
- Non un programma indipendente, ma un programma integrato in una pagina web, in esecuzione sulla JVM del browser
- Restrizioni di sicurezza

Stefano Mizzaro - Applet

7

La classe Applet

- `java.applet.Applet` è una sottoclasse di `java.awt.Panel`
- Da sovrascrivere per creare i nostri applet
- Non c'è un main:
 - i metodi di `Applet` vengono chiamati (implicitamente, dal browser)
 - e vanno quindi sovrascritti dal programmatore

Stefano Mizzaro - Applet

8

Il nostro primo applet

```
import java.applet.*;
import java.awt.*;
public class CiaoATuttiApplet extends Applet {
    public void paint (Graphics g) {
        g.drawString("Ciao a tutti!", 5, 25);
    }
}
```

Stefano Mizzaro - Applet

9

Il file html

```
<html>
<head>
  <title>Salve a tutti voi</title>
</head>
<body>
  <p>Il mio primo applet dice:
    <applet code="CiaoATuttiApplet.class"
            width="150" height="25">
  </applet></p>
</body>
</html>
```

Stefano Mizzaro - Applet

10

Come funziona

- Il browser ha una JVM, in grado di eseguire codice Java
- Abbiamo semplicemente sovrascritto `paint()` ...
 - (ereditato da `Component`)
- ...che viene invocato implicitamente
 - "Quando c'è bisogno" (di disegnare l'applet)
- `paint(Graphics g)` ...
 - `g` è il "contesto grafico" ("la cosa su cui disegnare")
 - Ha vari metodi (cfr. doc. API) fra cui `drawString`
- N.B. Tutti i componenti vengono disegnati automaticamente!

Stefano Mizzaro - Applet

11

L'appletviewer

- Altro "attrezzo" del JDK
- Visualizza applet, ma non html
- Ha bisogno di un file con il tag "`applet`"
 - "Trucco": codice html dentro a commenti in file java
- ...a volte i browser fanno scherzi strani...

Stefano Mizzaro - Applet

12

Uso dell'appletviewer

```
/*<applet code="CiaoATuttiApplet1.class"
   width="150" height="25"></applet>*/
import java.applet.*;
import java.awt.*;
public class CiaoATuttiApplet1 extends Applet {
    public void paint (Graphics g) {
        g.drawString("Ciao a tutti!",5,25);
        g.fillOval(0,0,100,100);
    }
}
```

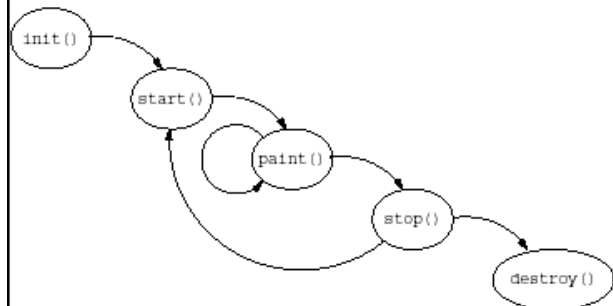
```
>javac CiaoATuttiApplet.java
>appletviewer CiaoATuttiApplet.html
```

```
>javac CiaoATuttiApplet1.java
>appletviewer CiaoATuttiApplet1.java
```

Stefano Mizzaro - Applet

13

"Ciclo di vita" di un applet



Stefano Mizzaro - Applet

14

Metodi di applet

- Metodi (vuoti) da **Applet**:
 - **init**: inizializzazione (~costruttore)
 - **start**: inizio esecuzione
 - **stop**: fine esecuzione, arresto
 - **destroy**: rilascio risorse
 - (leggere variazioni fra i vari browser...)
- Da **Component**:
 - **paint**: quello che l'applet mostra
- Invocazioni "automatiche"

Stefano Mizzaro - Applet

15

Invocazione metodi di Applet

```
/*<applet code="MetodiApplet.class" width="10"
   height="10"></applet>*/
import java.applet.Applet;
import java.awt.*;
public class MetodiApplet extends Applet {
    public void init() {
        System.err.println("init...");
    }
    public void start() {
        System.err.println("start...");
    }
    public void stop() {
        System.err.println("stop...");
    }
    public void destroy() {
        System.err.println("destroy...");
    }
    public void paint(Graphics g) {
        System.err.println("paint...");
    }
}
```

```
>javac MetodiApplet.java
>appletviewer MetodiApplet.java
```

Stefano Mizzaro - Applet

16

Applet per disegnare col mouse

- Vediamo un applet che ci consente di disegnare trascinando il mouse
- Avremo bisogno di:
 - Un applet contenente un **Canvas**
 - Ascoltatori di **eventi sul mouse** (click e drag) associati al **Canvas**
- Schema di disegno differente:
 - Non implemento **paint**
 - Al drag, direttamente nell'ascoltatore, disegno un segmento
 - Uso **getGraphics** (ereditato da **Component**) per ottenere il **Graphics**

Stefano Mizzaro - Applet

17

ScribbleApplet1.java (1/2)

```
/*<applet code="ScribbleApplet1.class"
   width="500" height="400"></applet>*/
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

public class ScribbleApplet1 extends Applet {
    private Canvas c = new MyCanvas();
    public void init() {
        this.setLayout(new BorderLayout());
        this.add(c, BorderLayout.CENTER);
    }
}
```

Stefano Mizzaro - Applet

18

ScribbleApplet1.java (2/2)

```
class MyCanvas extends Canvas {
    private int lastX, lastY;
    public MyCanvas() {
        addMouseListener(new AscoltaClick());
        addMouseMotionListener(new AscoltaDrag());
    }
    class AscoltaClick extends MouseAdapter {
        public void mousePressed(MouseEvent e) {
            lastX = e.getX(); lastY = e.getY();
        }
    }
    class AscoltaDrag extends MouseMotionAdapter {
        public void mouseDragged(MouseEvent e) {
            int x = e.getX(); int y = e.getY();
            getGraphics().drawLine(lastX,lastY,x,y);
            lastX = x; lastY = y;
        }
    }
}

```

```
>javac ScribbleApplet1.java
>appletviewer ScribbleApplet1.java
```

Stefano Mizzaro - Applet

Eventi del mouse (1/2)

- Non rispettano la regola dell'AWT
"1 evento ↔ 1 ascoltatore"
- Qui è
"1 evento ↔ 2 ascoltatori"
- La classe per gli eventi:
 - **MouseEvent**
- Le 2 interfacce per gli ascoltatori:
 - **MouseListener** e **MouseMotionListener**
- Ci sono anche i 2 adapter:
 - **MouseAdapter** e **MouseMotionAdapter**

Stefano Mizzaro - Applet

20

Eventi del mouse (2/2)

- **MouseListener**
 - (5 metodi con parametro **MouseEvent**)
 - Pulsante mouse premuto (**mousePressed**)
 - Pulsante mouse rilasciato (**mouseReleased**)
 - Click (premutato + rilasciato) (**mouseClicked**)
 - Cursore entra nel componente (**mouseEntered**)
 - Cursore lascia il componente (**mouseExited**)
- **MouseMotionListener**
 - (2 metodi con parametro **MouseEvent**)
 - Mouse mosso (**mouseMoved**)
 - Mouse trascinato (**mouseDragged**)
- **Component:**
 - **addMouseListener** e **addMouseMotionListener**

Stefano Mizzaro - Applet

21

Esercizi

- Rifarlo senza classi interne
- Rifarlo usando la classe `java.awt.Point`
- Esercizio(*): farlo modificando il `paint()` di `MyCanvas` (così il disegno è permanente)
- Rifare con `Frame` anziché `Applet`

Stefano Mizzaro - Applet

22

Aggiungiamo i colori

- Scelta del "colore della penna"
- Potremmo farlo con dei pulsanti, 1 per colore
- Ma non è il componente giusto
- Usiamo invece `java.awt.Choice`

Stefano Mizzaro - Applet

23

ScribbleAppletColor (1/2)

```
/*<applet code="ScribbleAppletColor.class"
    width="500" height="400"></applet>*/
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class ScribbleAppletColor extends Applet {
    private Canvas c = new MyCanvas();
    private Choice ch = new Choice();
    private static final Color[] colors =
        {Color.BLACK, Color.RED, Color.GREEN, Color.BLUE};
    private static final String[] colorNames =
        {"Nero", "Rosso", "Verde", "Blu"};
    public void init(){
        this.setLayout(new BorderLayout());
        this.add(c, BorderLayout.CENTER);
        for (int i = 0; i < colorNames.length; i++)
            ch.addItemListener(new AscoltaColore());
        this.add(ch, BorderLayout.EAST);
    }
}

```

Stefano Mizzaro - Applet

24

ScribbleAppletColor (2/2)

```
class AscoltaColore implements ItemListener {
    public void itemStateChanged(ItemEvent e) {
        c.setForeground(
            colors[ch.getSelectedIndex()]);
    }
}

class MyCanvas extends Canvas {
    // idem...
}
```

```
>javac ScribbleAppletColor.java
>appletviewer ScribbleAppletColor.java
```

Stefano Mizzaro - Applet

Commenti

- Alla scelta sul **Choice** **ch...**
- ... viene estratto l'indice della scelta...
- ... che viene usato per settare il colore di primo piano corrispondente sul **Canvas** **c**
- Soliti Ex.: rifarlo senza classe interna per **AscoltaColore**, con disegno permanente, ECC.

Stefano Mizzaro - Applet

26

Choice

- "Scelta" di un'alternativa da un elenco
- Metodi (vedi documentazione API):
 - Costruttore
 - `add(String)`
 - `addItemListener(ItemListener)`
 - `int getSelectedIndex()`
 - `String getSelectedItem()`
- **ItemEvent**: evento generato alla scelta
- **ItemListener**: ascoltatore
 - `itemStateChanged(ItemEvent)`

Stefano Mizzaro - Applet

27

Restrizioni sugli applet

- Un applet non può:
 - accedere al file system del client
 - aprire connessioni di rete con un calcolatore che non sia il web server
 - stampare
 - creare processi figli
- Tutte le finestre aperte da un applet riportano l'indicazione che sono state create da un applet (esempio →)

Stefano Mizzaro - Applet

28

Applet che apre un Frame

```
/*<applet code="OpenFrameApplet.class" width="10"
height="10"></applet>*/
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class OpenFrameApplet extends Applet {
    public void init() {
        final Frame f = new Frame();
        MenuBar mb = new MenuBar();
        Menu m = new Menu("File");
        MenuItem mi = new MenuItem("Esci");
        m.add(mi);
        mb.add(m);
        f.setMenuBar(mb);
        f.setVisible(true);
        f.pack();
        mi.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                f.dispose();
            }
        });
    }
}
```

```
>javac OpenFrameApplet.java
>appletviewer OpenFrameApplet.java
```

Commenti

- **MenuItem, Menu, MenuBar, add, setMenuBar** (vedi documentazione API)
- Menu solo nei **Frame**
 - `setMenuBar()` è definito in **Frame**
- Selezione voce da menu (~ pulsanti):
 - **ActionEvent** → **ActionListener** → **actionPerformed**
- **final Frame f** perché ascoltatore con classe interna che accede a variabile locale
 - Alternativa (forse migliore): variabile d'istanza – anche non final (ma **private**)

Stefano Mizzaro - Applet

30

Riassunto

- Esame
- Applet
 - Esempi e restrizioni
 - Disegno dei componenti, metodo `paint()` e classe `Graphics`
 - Eventi del mouse
 - **Choice**
 - Menu

Stefano Mizzaro - Applet

31

Riassunto del corso

- Introduzione alla programmazione
 - Accento sulla qualità
- In dettaglio
 - La programmazione strutturata
 - Il linguaggio Java
- Solo accennati
 - Programmazione OO
 - API
- Cenni computabilità (prox. volta)

Stefano Mizzaro - Applet

32