



Graphics

Disegno a mano libera

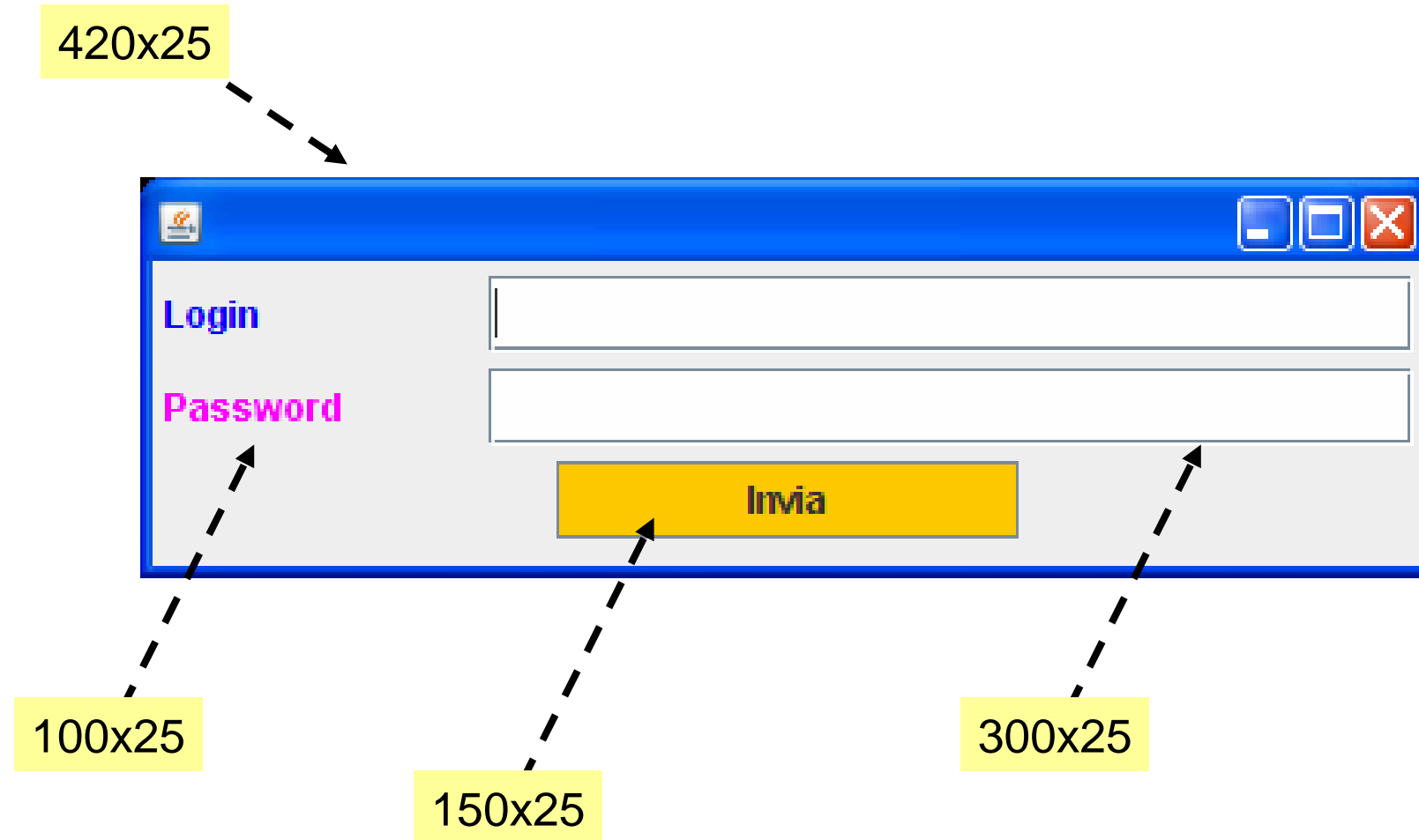
Tornando ai JFrame



A screenshot of a Java Swing JFrame window. The window has a blue title bar with standard Windows-style window controls (minimize, maximize, close) on the right. The main content area has a light gray background. On the left side, the word "Login" is written in blue text, and "Password" is written in magenta text. To the right of "Login" is a white text input field. To the right of "Password" is another white text input field. Below these two input fields is a yellow button with the text "Invia" in black.

Tempo 20 minuti

Esercizio su JFrame – le dimensioni





Soluzione

```
import javax.swing.*;
import java.awt.*;

public class NewFinestra{

    public static void main(String[] args) {
        JFrame f = new JFrame();
        JPanel p = new JPanel();
        f.setContentPane(p);

        f.setSize(420, 130);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel labelLogin = new JLabel("Login");
        labelLogin.setPreferredSize(new Dimension(100,25));
        p.add(labelLogin);
        labelLogin.setForeground(Color.BLUE);

        JTextField textfieldLogin = new JTextField();
        p.add(textfieldLogin);
        textfieldLogin.setPreferredSize(new Dimension(300,25));

        JLabel labelPassword = new JLabel("Password");
        labelPassword.setPreferredSize(new Dimension(100,25));
        labelPassword.setForeground(Color.MAGENTA);
        p.add(labelPassword);

        JTextField textfieldPassword = new JTextField();
        p.add(textfieldPassword);
        textfieldPassword.setPreferredSize(new Dimension(300,25));

        final JButton button = new JButton("Invia");
        p.add(button);
        button.setBackground(Color.ORANGE);
        button.setPreferredSize(new Dimension(150,25));

        f.setVisible(true);
    }
}
```

Per casa

- Aggiungere una nuova riga con una JCheckBox con la scritta “login sicuro” di colore verde
- Aggiungere un secondo pulsante accanto al primo con sfondo rosso e la scritta “cancella”
- Modificare il colore del textfield in modo che i caratteri della login siano blu e quelli della password magenta
- Cambiare il colore della selezione della textfield della password giallo con sfondo rosso





Introduzione

- Utilizzo dell'oggetto Graphics e Graphics2D per il disegno
- Realizzazione di un pannello per il disegno
 - Override del metodo paintComponent(Graphics)
 - Utilizzo delle primitive di disegno
 - Aggiunta del pannello in un JFrame



JComponent

- Finora i componenti grafici utilizzati erano già definiti nelle Swing di Java
 - JButton, JPanel, JCheckBox, JTextField
 - Estendono tutti da JComponent
 - Il modo in cui vengono disegnati è definito nel metodo `paintComponent(...)` – ciascuna classe sovrascrive l'implementazione di JComponent secondo le proprie esigenze
- Creiamo un nostro componente grafico personalizzato
 - Il disegno del componente lo effettuiamo attraverso primitive geometriche di disegno



Come si disegna un JComponent

1. Si definisce una classe che estende JComponent

2. Sovrascrittura del metodo

```
public void paintComponent(Graphics g)
```

1. Chiamata a super.paintComponent(g)

2. Cast a Graphics2D

3. Utilizzo delle primitive di disegno

3. Utilizzo del nuovo componente in modo analogo agli altri componenti Swing (JPanel, JButton etc)

1. Per esempio setContentPane(...) in un JFrame

→ Quando viene invocato il metodo paintComponent e da chi?

→ Prova a fare il ridimensionamento della finestra



Graphics2D

- public abstract void **draw**(Shape s)
 - Disegna il contorno dell'oggetto Shape specificato.
- public abstract void **fill**(Shape s)
 - Disegna l'interno dell'oggetto Shape specificato.
- public abstract void **drawString**(String str, int x, int y)
 - Disegna un stringa
- public abstract void **setStroke**(Stroke s)
 - Imposta il pattern di disegno.
- public abstract void **setColor**(Color c)
 - Imposta il Colore del disegno
- public abstract void **setPaint**(Paint paint)
 - Evoluzione del precedente, permette effetti di colore evoluti



Shape (forme geometriche)

- **Ellipse2D.Double** (double x, double y, double w, double h)
- **Rectangle2D.Double**(double x, double y, double w, double h)
- **Line2D.Double** (double x1, double y1, double x2, double y2)
- **RoundRectangle2D.Double** (double x, double y, double w, double h, double arcw, double arch)
 - arcw, arch: larghezza e altezza dell'arco
- **Path2D.Double**()
 - void moveTo(double x, double y)
 - void lineTo(double x, double y)
 - void curveTo(double x1, double y1, double x2, double y2, double x3, double y3)

→ Javadoc di Shape per conoscere altre sottoclassi



Stroke (pattern delle linee)

- **BasicStroke** (float width)

- Utilizzato (per ora) per definire la dimensione della linea



Colori per il disegno e il riempimento

- Il metodo setColor(...) permette di impostare un colore uniforme
- Il metodo setPaint(...) invece utilizza l'interfaccia Paint

- Color


- GradientPaint

- GradientPaint(float x1, float y1, Color color1, float x2, float y2, Color color2)
- GradientPaint(float x1, float y1, Color color1, float x2, float y2, Color color2, boolean cyclic)

- RadialGradientPaint

- RadialGradientPaint(float cx, float cy, float radius, float[] fractions, Color[] colors)
 - Radius: raggio del gradiente
 - fractions: valori tra 0 e 1 (crescenti) pari al numero di colori che identifica le percentuali dei vari colori usati - con 2 colori {0f, 0.5f} c'e' una distribuzione uniforme

→ Aspettare gli array



Math.random() (java.util.*)

- Genera un numero random tra 0 e 1 (escluso)
- **public static double random()**
 - ritorna un valore double maggiore o uguale a 0.0 e minore di 1.0.
- Come fare a generare un numero tra 0 e MAX?
 - `Math.random()*MAX`
- E per ottenere un valore Intero?
 - `(int)Math.random()*MAX`



Esercizio

- Disegnare in una finestra 800x800 due cerchi uno verde(pieno) e uno rosso (solo contorno)



Soluzione

```
import javax.swing.*;
import java.awt.*;
import java.awt.geom.*;
public class Disegno extends JComponent{

    public Disegno() {

    }

    public void paintComponent(Graphics g){
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D)g;
        //qui metto il mio disegno

        Ellipse2D.Double cerchio=
            new Ellipse2D.Double(100,50,20,20);
        g2.setColor(Color.GREEN);
        g2.fill(cerchio);

        Ellipse2D.Double cerchio2=
            new Ellipse2D.Double(300,300,20,20);
        g2.setStroke(new BasicStroke(4));
        g2.setColor(Color.RED);
        g2.draw(cerchio2);
    }

    public static void main(String[] aaa){
        JFrame f=new JFrame();
        Disegno d=new Disegno();
        f.setContentPane(d);
        f.setSize(800,800);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



Esercizio per casa

- Disegnare una Mongolfiera (vedi figura pagina successiva)
 - Utilizzare le primitive di disegno fill e draw sulle Shape viste a lezione
 - Utilizzare setColor(...) per impostare il colore di disegno
 - Utilizzare drawString per la stampa del testo



Esercizio per casa/2





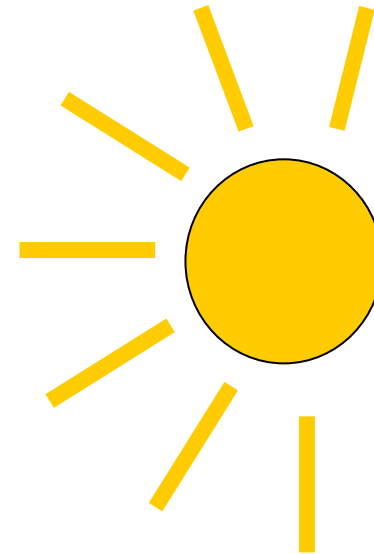
Disegnare

- Disegnare in una finestra 800x800 i seguenti oggetti in posizione x,y random all'interno della area specificata con colori differenti a piacere
 1. 10 cerchi di raggio 20
 2. 10 rettangoli 20x10
 3. 10 linee in posizione $(x1,y1),(x2,y2)$ random
 4. 10 rettangoli con sfondo e bordo di colori differenti e dimensioni 20x20
 5. Un poligono a 30 punti random
- Eseguire un passo alla volta e vederne il risultato



... continua

1. Modificare il disegno in modo da utilizzare colori sfumati
2. Poi aggiungere a dx il sole



Esercizio per casa

- Disegnare una strada con 5 Camion uno di seguito all'altro

