

Gestione **Thread** : scritta scorrevole che si modifica da campo di testo

**Esercizio:** realizzare un'applicazione con una scritta scorrevole che si modifica da campo di testo con uso di contenitori/componenti Swing evitando *sfarfallio*.

**Possibile soluzione:** si possono inserire due **PANNELLI** (in un Container impilando in verticale); il secondo per disegnare con **interfaccia Runnable**; gestione di **evento di azione** con interfaccia tipo ActionListener associata al campo di testo (alla pressione di INVIO) per cancellare il testo stesso. Nella colonna di destra le **modifiche** per gestire la possibilità che la scritta scorra per tutta la **larghezza** della cornice anche se l'utente ridimensiona la finestra.

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

public class TestoS implements ActionListener {
    private String nome;
    private JTextField t;
    private MioPanel disegno;
    private JPanel p;
    private JFrame f;
    private Container c;
    private JLabel l;

    public TestoS() { // costruttore

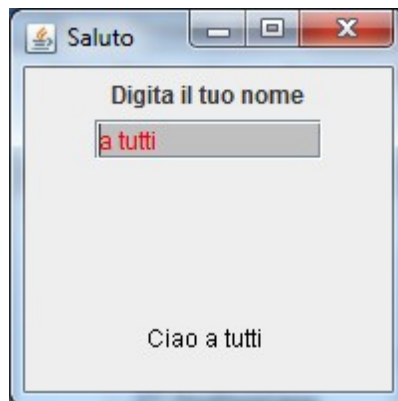
        nome = "a tutti";
        f = new JFrame ("Saluto");
        c = f.getContentPane();
        c.setLayout (new BorderLayout (c, BorderLayout.PAGE_AXIS)); // impila gli elementi in verticale
                                                // nel Container

        p = new JPanel();
        l = new JLabel ("Digita il tuo nome");
        t = new JTextField(nome, 10);
        t.setForeground(Color.red); // testo rosso
        t.setBackground(Color.lightGray); // sfondo grigio chiaro per la casella di testo
        t.addActionListener(this);
        disegno = new MioPanel(t);
        p.add(l);
        p.add(t);
        c.add(p);
        c.add(disegno);
        f.setSize(200,200); // con f.pack() bisogna ingrandire la finestra per vedere il "disegno"
        f.setLocation(0,0); // se non se ne impostano le dimensioni
        f.setResizable(false); // f.setResizable(true);
        f.setVisible(true);
        Thread runner = new Thread (disegno);
        runner.start(); // richiama il metodo run() del "thread" disegno
        f.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
    }
    public void actionPerformed(ActionEvent e) { // alla pressione di INVIO
        t.setText("");
    }
    public static void main(String [] args) {
        new TestoS();
    }
} // fine applicazione TestoS
```

```
// classe esterna
class MioPanel extends JPanel implements Runnable {
```

<pre>private String nome; private JTextField testo; private int Xpos;  public MioPanel (JTextField t) {     testo = t;     setForeground(Color.black); // scrive in nero }  // metodo obbligatorio per eseguire il thread public void run () {     while (true) {         for (Xpos =0; Xpos&lt;150; Xpos++) {              repaint();             try{                 Thread.sleep(80);             }catch(InterruptedException ie) { return;}         }     } }</pre>	<pre>private String nome; private JTextField testo; private int Xpos;  public MioPanel (JTextField t) {     testo = t;     setForeground(Color.black); // scrive in nero }  // larghezza scelta come // massimo valore x nello spostamento [*] public void run () {     while (true) {         for (Xpos =0; Xpos&lt;getWidth(); Xpos++) {              repaint();             try{                 Thread.sleep(80);             }catch(InterruptedException ie) {return;}         }     } }</pre>
--	---

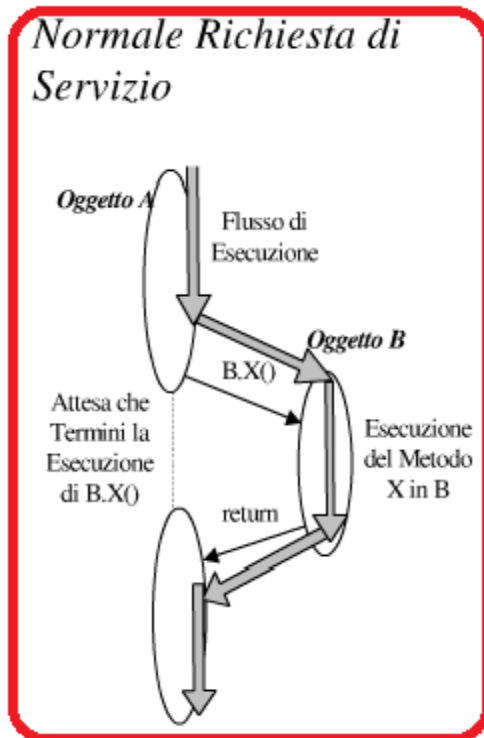
```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    nome = testo.getText();
    g.drawString("Ciao " + nome, Xpos, 50); // scelta dell'ordinata in modo da vedere meglio la scritta
}
}
```



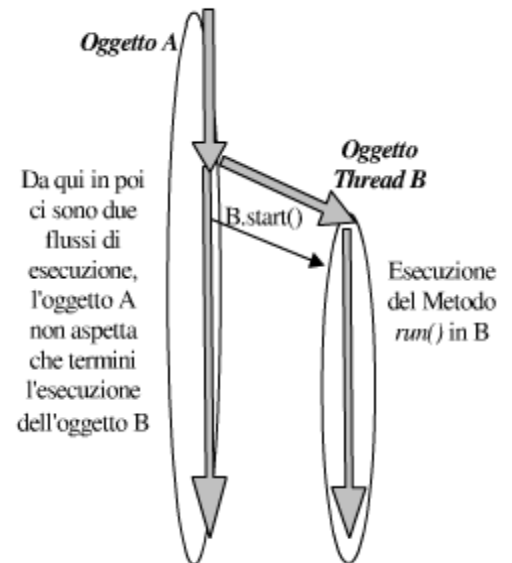
All'inizio la scritta è Ciao a tutti

nb: l'uso di una *tela* (oggetto di classe che estende Canvas) disegnando con metodo paint() causa **maggior effetto "flicker"** o *sfarfallio*.

**Confronto: scritta scorrevole che si modifica da campo di testo** (unico thread associato al main)



*Richiesta di Servizio start() a un Thread*



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

```
public class SenzaThread implements ActionListener {
    private String nome;
    private JTextField t;
    private MioPanel disegno;
    private JPanel p;
    private JFrame f;
    private Container c;
    private JLabel l;

    public SenzaThread() { // costruttore

        nome = "a tutti";
        f = new JFrame("Saluto");
        c = f.getContentPane();
        c.setLayout(new BorderLayout(c, BorderLayout.PAGE_AXIS)); // impila gli elementi in verticale
        p = new JPanel();
        l = new JLabel("Digita il tuo nome");
        t = new JTextField(nome, 10);
        t.setForeground(Color.red); // testo rosso
        t.setBackground(Color.lightGray); // sfondo grigio chiaro per la casella di testo
        t.addActionListener(this);
        disegno = new MioPanel(t);
        p.add(l);
        p.add(t);
        c.add(p);
        c.add(disegno);
        f.setSize(200,200); // con fpack bisogna ingrandire la finestra per vedere il "disegno"
```

```

f.setLocation(0,0); // se non se ne impostano le dimensioni
f.setResizable(true);
f.setVisible(true);
f.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE); // deve essere posta prima del ciclo di disegno
while(true)
    disegno.go(); // senza craere thread
}

public void cancellaNome() {
    t.setText("");
}
public void actionPerformed(ActionEvent e) {
    cancellaNome(); // alla pressione di INVIO
}

public static void main(String [] args) {
    new SenzaThread();
}

// classe esterna
class MioPanel extends JPanel {
    private String nome;
    private JTextField testo;
    private int Xpos;

    public MioPanel (JTextField t) {
        testo = t;
    }

    public void go () { // servizio richiesto

        for (Xpos =0; Xpos<getWidth(); Xpos++) { // ascissa massima
                                                    // in funzione della larghezza del pannello

            setForeground(Color.black); // scrive in nero
            repaint();
            try{
                Thread.sleep(80);
            } catch (InterruptedException ie) {return;}

        }
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        nome = testo.getText();
        g.drawString("Ciao " + nome, Xpos,50); // scelta dell'ordinata
                                                // in modo da vedere meglio la scritta
    }
}

```

**Normale Richiesta di Servizio**

