

Letture da file di testo con Scanner, visualizzazione in area di testo, salvataggio di modifiche

```
// lettura da file con visualizzazione in area di testo
// modifica del testo
// salvataggio su file del contenuto dell'area di testo
```

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

```
class GUI {
```

```
    private JTextArea ta = new JTextArea();
    private String sl;
```

```
    public void leggiFile() {
        // LETTURA da file di testo
```

```
        try {
            FileReader fr = new FileReader("FileIni.txt");
            Scanner sc = new Scanner(fr);
            while(sc.hasNext()) {
                ta.append(sc.nextLine());
                ta.append("\n");
            }
            fr.close(); // chiusura File
        } catch (IOException e) {}
```

```
    }
    public void scriviFile() {
        // SCRITTURA su file di testo
```

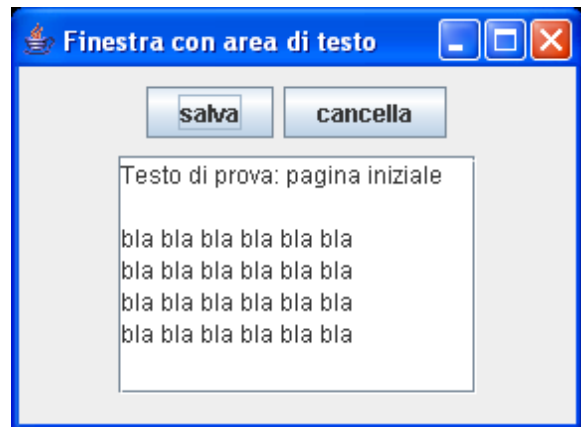
```
        try {
            FileWriter fw = new FileWriter("FileOut.txt");
            PrintWriter out = new PrintWriter(fw);
            sl = ta.getText(); // salva in String tutto il testo
            out.print(sl);
            fw.close(); // chiusura File
        } catch (IOException e) {}
```

```
    }
    public void visualizza(){
```

```
        JFrame f = new JFrame("Finestra con area di testo");
        Container c = f.getContentPane();
        JPanel p = new JPanel();
         JButton b1 = new  JButton ("salva");
         JButton b2 = new  JButton ("cancella");
        b1.addActionListener (new  mioAscoltatore()); // gestione eventi di azione
        b2.addActionListener (new  mioAscoltatore());
        p.add (b1);
        p.add (b2);
        p.setPreferredSize(new Dimension(180, 35));
        ta.setEditable (true);
         JScrollPane listPane = new  JScrollPane(ta);
        listPane.setPreferredSize(new Dimension(180, 120));
        c.setLayout(new  FlowLayout()); //Layout the content pane
        c.setPreferredSize(new Dimension(280, 180));
        c.add(p);
        c.add(listPane);
        f.pack();
        f.setVisible(true);
```

```
    }
```

```
// classe interna
```



```

class mioAscoltatore implements ActionListener {

    public void actionPerformed (ActionEvent ev) {

        String Bottone = ev.getActionCommand();    // testo mostrato da pulsante
        if (Bottone.equals ("salva") )
            scriviFile();
        else if (Bottone.equals ("cancella") )
            ta.setText ("");
        }
    } // fine class GUI

public class EsempioEventi {

    public static void main(String[] args) {

        GUI o = new GUI();
        o.leggiFile();
        o.visualizza();
    }
} // fine applicazione che usa class GUI

```

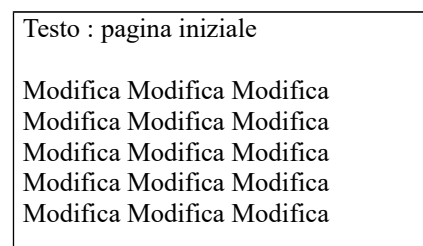
Effetto alla pressione del bottone “cancella”:



Effetto di **modifiche** (uso Ctr C e Ctr V per copia/incolla):



Effetto alla pressione del bottone “salva” :
salvataggio su file di testo FileOut.txt



Attività: migliorare il progetto prevedendo uso di [dialog box](#) per scegliere il nome del file

