

JFileChooser per visualizzare **finestre modali** per **scegliere percorso** di file

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

class ScelteFile extends JFrame {    // nessun accesso ai file ... da gestire con successive letture/scritture

public ScelteFile(){

    JFileChooser fc = new JFileChooser(); // la directory di default è Documents

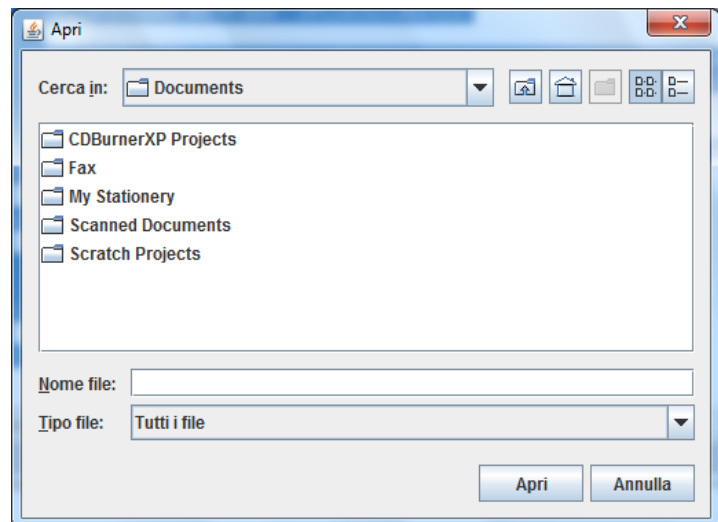
    int returnVal = fc.showOpenDialog(this); // apre finestra modale ...
                                                // visualizza un JFileChooser per apertura file
                                                // come parent è scelto this (se la chiamata è in contesto static, usare null)

    if(returnVal == JFileChooser.APPROVE_OPTION) {
        System.out.println("You chose to open this file: " + fc.getSelectedFile().getName());
    }
}
```

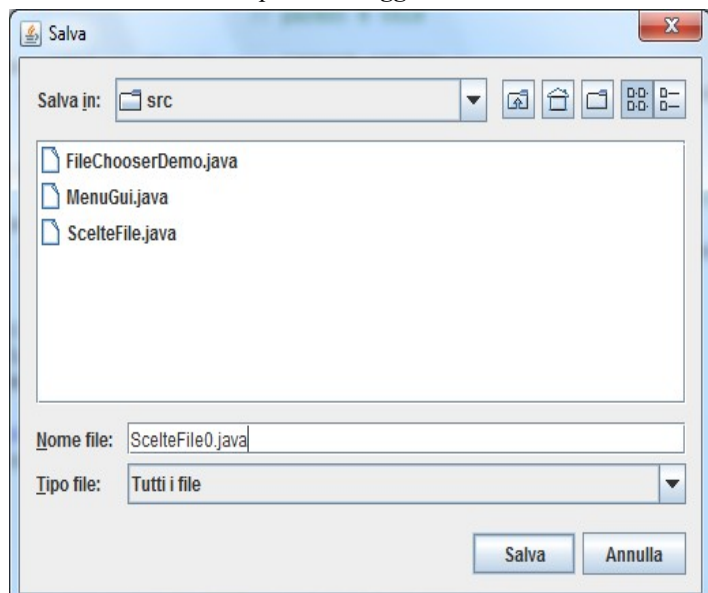
// con percorso

Cerca in:

// selezionato



```
returnVal = fc.showSaveDialog(null); // apre finestra modale ...
// visualizza un JFileChooser per salvataggio file.
```



```
    pack();
    setVisible(true);
    setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
}
public static void main (String arg[]){ new ScelteFile(); } // fine main
} // fine classe
```

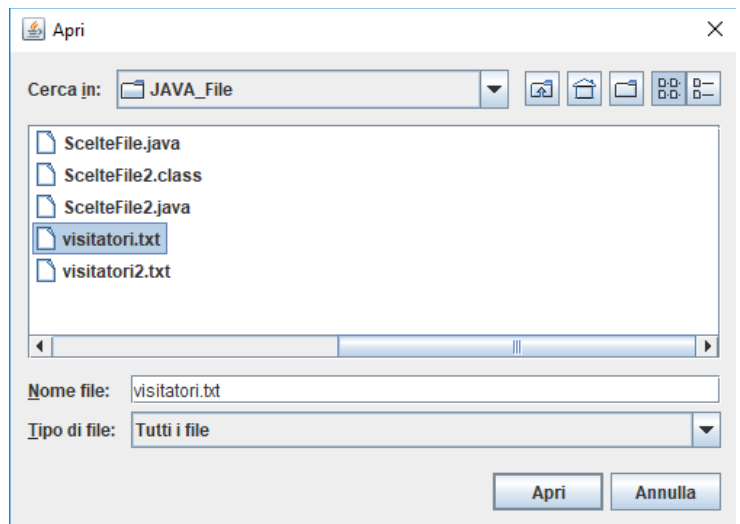


```
import java.awt.*;
import javax.swing.*;

class ScelteFile2 extends JFrame {
    // nessun accesso ai file
    //... da gestire con successiva lettura

    String dir="."; // la directory è quella corrente
    String nome = "visitatori.txt"; // di default
    String path = dir + nome ; // di default per aprire

    public ScelteFile2 () {
```



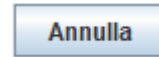
```
        JFileChooser fc = new JFileChooser(dir);

        int returnVal = fc.showOpenDialog(this);
        // apre finestra modale ...
        // visualizza un JFileChooser per apertura file.
        // come parent si è scelto this

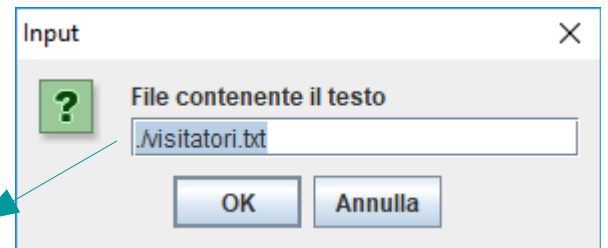
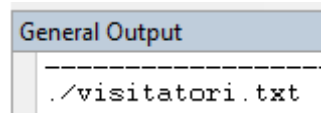
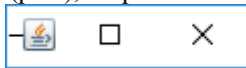
        if(returnVal == JFileChooser.APPROVE_OPTION) { // controllo sulla validità del nome o percorso del file
            nome=fc.getSelectedFile().getName();
            System.out.println("Hai scelto di aprire il file: " + nome);
            path=fc.getCurrentDirectory()+"\\"+nome; // doppio backslash necessario
            ///path=fc.getSelectedFile().getPath(); // altro modo per visualizzare il percorso completo
            System.out.println(path); // per test
        }
        else { //gestendo pressione Annulla o chiusura senza digitare il nome
            String s=JOptionPane.showInputDialog ("File contenente il testo", path); // da finestra modale
            // di tipo JOptionPane

            if(s!=null)
                path = s;
        }

        System.out.println (path); // per test in caso di Annulla / Annulla oppure chiusure con X
        pack();
        setVisible(true);
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
```



```
}
public static void main (String arg[]){
    new ScelteFile2();
} // fine main
} // fine classe
```



Sintesi: Crea il file chooser fc usando new JFileChooser();

1. Chiamare uno dei seguenti metodi che mostrano un dialogo modale contenente il file chooser


```
int scelta = fc.showOpenDialog(finestra);
int scelta = fc.showSaveDialog(finestra);
int scelta = fc.showDialog(finestra,"titolo");
```

 dove finestra è la componente da cui il dialogo deve dipendere.

2. Leggere il valore di ritorno, che può essere

JFileChooser.APPROVE_OPTION o

JFileChooser.CANCEL_OPTION che corrisponde alla pressione di o

e, in caso di approvazione, considerare **valido** il nome del file

```
if (scelta==JFileChooser.APPROVE_OPTION) {
    fc.getSelectedFile().getName(); //... poi si gestirà l'accesso al file
    // con metodi di scrittura /lettura
}
```

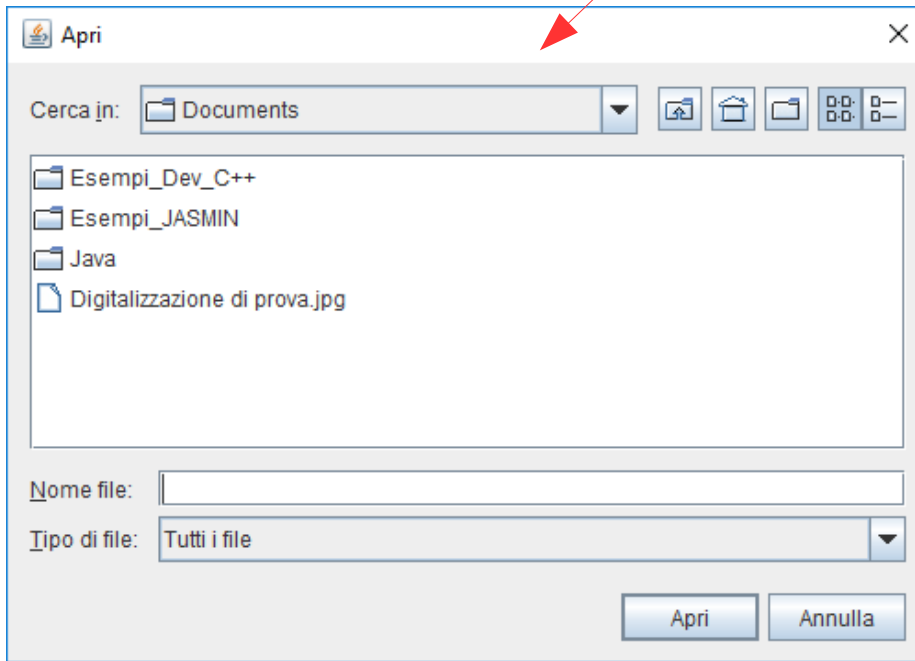
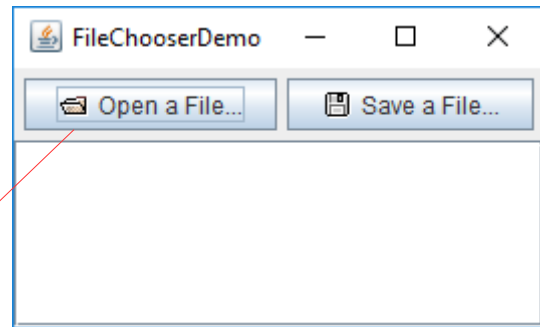
Altri metodi consentono di assegnare directory corrente, definire filtri, personalizzare.

Esempi dal Tutorial

FileChooserDemo

[sorgente](#) e immagini: [Open16.gif](#), [Save16.gif](#)

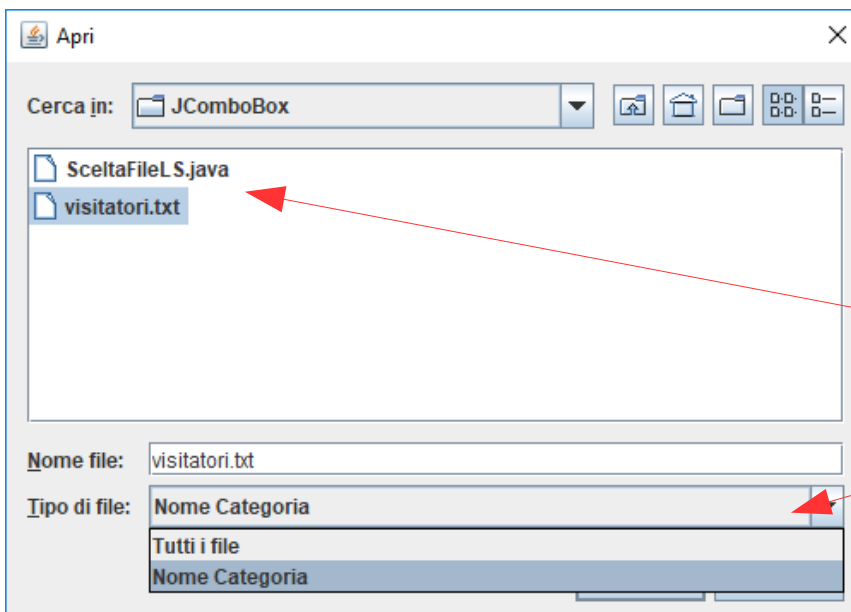
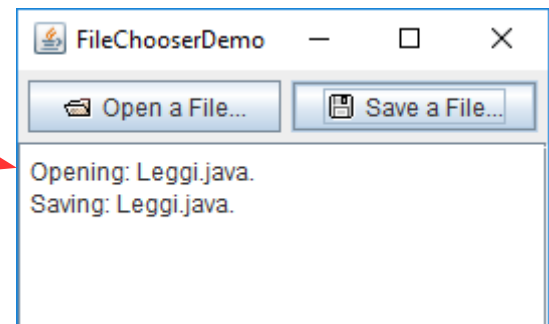
oppure il [progetto](#) in forma compressa



selezionando gli opportuni percorsi, si potrà *visualizzare* il *nome* del file scelto

nb: *nessun* accesso ai file

da **API**
se si desidera *filtrare* la visualizzazione del tipo di file creando una *categoria personalizzata*



```
import javax.swing.filechooser.*;
```

```
JFileChooser fc = new JFileChooser();
```

```
FileNameExtensionFilter filter =  
new FileNameExtensionFilter ("Nome Categoria",  
"txt", "java");
```

```
fc.setFileFilter(filter);
```

si visualizza la *categoria personalizzata*

Esempio di metodi per accesso a file di testo

```
/**
 * Metodo che legge da File con percorso passato come parametro
 * @param nome => stringa che è il percorso del file
 * @return sF => stringa che è il testo contenuto nel file letto
 */
public String leggiFile(String nome){
    String sF="";

    try{
        FileReader fr = new FileReader (nome);
        Scanner in = new Scanner(fr);
        while (in.hasNextLine()) { // se l'input termina hasNextLine() ritorna falso
            String line = in.nextLine(); // lettura di una riga alla volta
            System.out.println(line); // per test
            // possibile memorizzazione come elementi di una collezione
            sF = sF + line + "\n"; // accodamento in stringa con gestione dell' a capo
        }
        fr.close();
    } catch (FileNotFoundException fe){ System.out.println("File non trovato");
    } catch (IOException e){ }
    return sF;
}

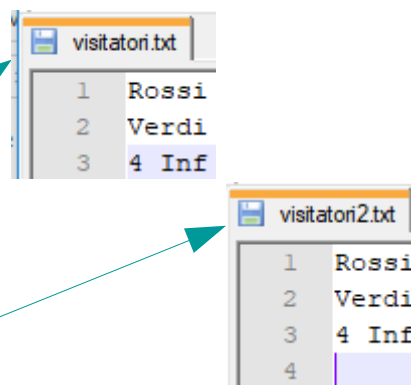
/**
 * Metodo che salva su File una stringa data come parametro
 * @param s => stringa da salvare su file
 * @param nome => stringa che è il percorso del file
 */
public void salvaSuFile(String s, String nome) {
    try{
        BufferedWriter outputWriter = new BufferedWriter(new FileWriter(nome)); // non accoda, riscrive
        outputWriter.write(s);// bufferizzato (unico accesso per flusso di caratteri)
        outputWriter.flush();
        outputWriter.close();// salva solo alla chiusura dello stream

    } catch(IOException ioe) { //Gestisco eccezioni I/O
        System.out.println("Errore: " + ioe.getMessage());
    }
}
}
```

>> **Progettare una GUI per leggere da file di testo con nome selezionabile con uso di JFileChooser e salvare il contenuto in altro file**

```
String dir=".";
String nome = "visitatori.txt";// di default
String path = dir + nome ; // di default per aprire

String nomeS = "visitatori2.txt";// di default
String pathS = dir + nomeS ; // di default per salvare
```



```
General Output
-----Configuration: <Default>
Hai scelto di aprire il file: visitatori.txt
Rossi
Verdi
4 Inf
Process completed.
```

>> **progettare** la visualizzazione del contenuto del file in un'area di testo o altro componente (**JComboBox**)