STAMPARE SU JAVA:

Il programma precedentemente realizzato per l'implementazione degli algoritmi di Skyline, è stato in questa occasione maggiormente affinato mediante l'inserimento di istruzioni che consentissero di visualizzare e rendere operativo un pulsante di stampa.

Facciamo una breve premessa definendo il package **`javax.swing:`**
- **Swing** è stato introdotto dalla versione 1.2 di Java ed è costruito sopra un altro package per la gestione delle GUI che è AWT (Abstract Windowing Toolkit) rilasciato nel package **`java.awt`**.

Gli elementi fondamentali di una GUI sono classificati in Swing nel seguente modo:

- *i contenitori principali (top-level container):*
  - la semplice finestra dalle funzionalità limitate: **`JWindow`**
  - una finestra con caratteristiche avanzate: **`JFrame`**
  - le finestre di dialogo di ogni tipo: **`JDialog`**
  - le finestre di applet: **`JApplet`**
- *i contenitori intermedi (intermediate container o pane):*
  - pannello: **`JPanel`**
  - pannello con barre di scorrimento: **`JScrollPane`**
  - scheda con etichette: **`JTabbedPane`**
- *i componenti finali (component):*
  - etichette: **`JLabel`**
  - caselle di testo: **`JTextField`**
  - area di testo: **`JTextArea`**
  - caselle combinate (combo box): **`JComboBox`**
  - bottoni: **`JButton`**
  - ecc

Ogni contenitore top-level contiene un contenitore intermedio detto content pane (pannello del contenuto). I componenti finali non possono essere inseriti direttamente nel contenitore di alto livello ma solo in un contenitore intermedio.

Ci soffermeremo in particolare sui uno dei componenti finali: i bottoni.

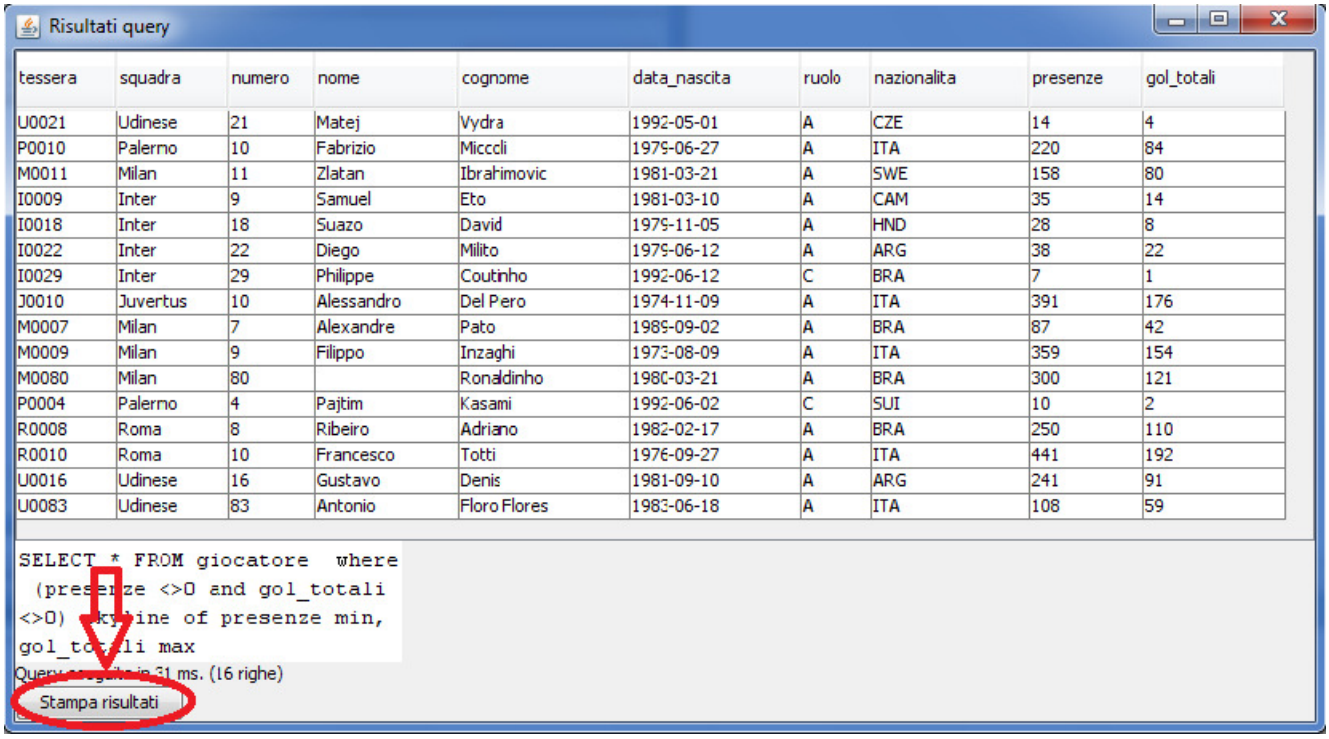
I pulsanti (bottoni) sono componenti dell'interfaccia Swing per confermare, annullare, le scelte effettuate in una finestra oppure per entrare o uscire da una finestra.

I costruttori base sono:

```
JButton() //crea un bottone senza nessun testo o icona sopra
JButton(Icon icon) // crea un bottone con una icona sopra
JButton(String text) // crea un bottone con testo sopra
JButton(String text, Icon icon) // crea un bottone con testo e icona sopra
```

In questo progetto abbiamo inserito un bottone con il costruttore del terzo tipo :

```
JButton (String text)
```



Tutto ciò che accade nella nostra interfaccia è interpretato come un ***evento*** (click del mouse, ecc).
Il modello utilizzato da Java per la gestione degli eventi è detto modello di delegazione: ogni componente visuale è una potenziale sorgente di eventi la cui gestione è delegata ad un oggetto ascoltatore con il compito di "ascoltare" e gestire gli eventi provenienti da quella sorgente.

Nel nostro caso:

```
JButton printresult = new JButton("Stampa risultati");
printresult.addActionListener(new ActionListener( )
{
  public void actionPerformed(ActionEvent e)
  {
        StampaElenco();
  }
} );
```

ActionEvent è un evento provocato dall'utente che agisce su un componente (click sul pulsante); vale per gli oggetti istanza delle classi: **`JButton, JRadioButton, JCheckBox, JComboBox, JTextfield.`**
L'oggetto ascoltatore associato è **ActionListener**. L'interfaccia ActionListener ha solo un evento: **`actionPerformed( ),`** quindi dovremo implementare solo il suo codice.

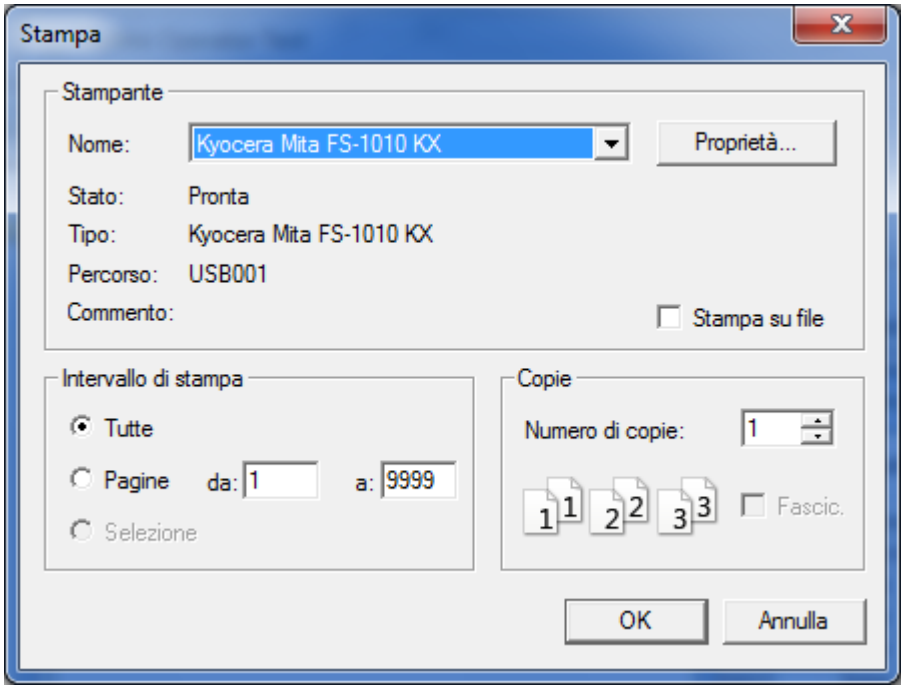
LA STAMPA

Java 1.2 supporta la stampa ad alta qualità tramite le classi nel pacchetto java.awt.print .

La classe PrinterJob è la classe principale del package in quanto è quella demandata al controllo della stampa. Questa classe non ha un costruttore, ma un sua istanza che può essere ottenuta invocando il metodo statico PrinterJob.getPrinterJob().

L'invio in stampa del PrinterJob tramite il comando printerJob.print() è condizionato dal valore ritornato dal metodo printerJob.printDialog(). Invocando questo metodo viene mostrata all'utente una dialog che gli consente di definire l'intervallo di pagine del documento da stampare ed il numero di copie da fare. Questo metodo ritorna *false* se l'utente decide di annullare la stampa, per cui in tal caso il metodo printerJob.print() non viene invocato.

Ecco cosa accadrà quando invocheremo PrintDialog :



L 'interfaccia Printable è costituita da un unico metodo la cui firma è:

```
int print(Graphics graphics, PageFormat pageFormat, int pageIndex)
```

in cui graphics è il contesto in cui la pagina da stampare verrà disegnata, mentre l'oggetto pageFormat contiene informazioni quali le dimensioni e l'orientamento della pagina. L'int ritornato da questo metodo deve essere uno dei due membri statici dell'interfaccia Printable, ovvero:

- `PAGE_EXISTS` che indica che la pagina è stata disegnata con successo
- `NO_SUCH_PAGE` che indica che il parametro page Index si riferisce ad una pagina non esistente (ovvero termina il processo di stampa).