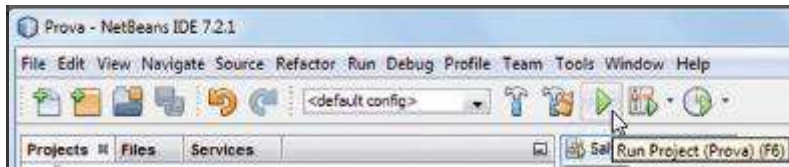


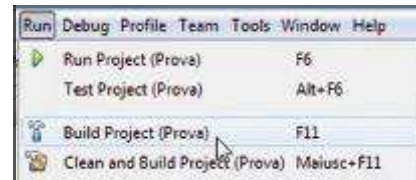
- 7 Run Project dal menu Run, oppure cliccando sull'apposita icona come mostrato dalla seguente figura:



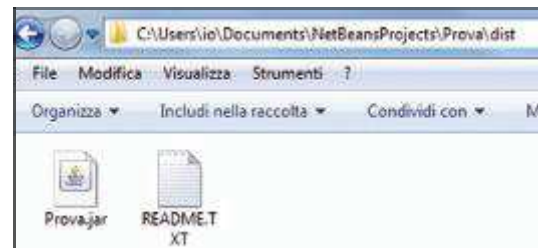
8



Per eseguire il progetto possiamo anche premere il tasto F6. L'eseguibile di tale programma è utilizzabile anche da altri utenti che non possiedono NetBeans. Tuttavia per compilare e creare l'eseguibile in formato .jar del programma dobbiamo attivare il menu Run, quindi selezionare Build Project: ►



La cartella del progetto, in questo caso chiamata prova, è una sottocartella di NetBeansProject. Tale cartella contiene, nella sottocartella dist, il file eseguibile in formato .jar: ►



In sintesi per creare una applicazione dobbiamo:

- 1 Creare un progetto: quando creiamo un progetto IDE dobbiamo anche creare l'ambiente che conterrà il necessario per l'esecuzione della nostra applicazione.
- 2 Aggiungere del codice al file sorgente generato: il file sorgente contiene del codice, scritto in linguaggio Java. Un progetto IDE quando viene creato fornisce uno scheletro del codice sorgente automaticamente generato. Occorrerà modificare tale codice per creare la nostra applicazione.
- 3 Compilare il file sorgente in un file .class: L'IDE invoca il compilatore java (javac), il quale traduce il sorgente in bytecode che la JVM è in grado di eseguire.
- 4 Eseguire il programma: L'IDE invoca il programma di lancio (java), il quale utilizzerà la JVM per eseguire l'applicazione.

## ■ Creare un'interfaccia grafica con NetBeans

Uno degli usi principali di NetBeans è quello che consente di creare un'interfaccia grafica di tipo GUI (Graphical User Interface) attraverso la tecnica Drag and drop, senza scrivere una riga di codice. Mediante il disegnatore di NetBeans possiamo trascinare un oggetto nella finestra (Frame) facendo generare al programma il codice necessario per la gestione dell'oggetto stesso. La procedura che segue mostra come realizzare una finestra inserendovi un pulsante.

1

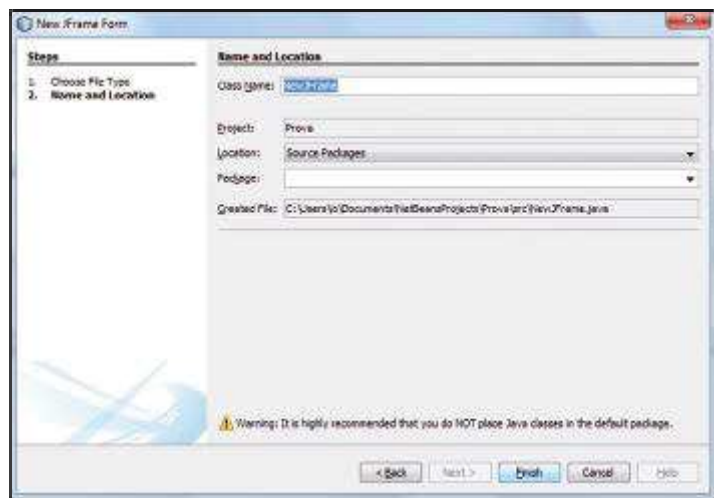
2

JFrame Form... ➤ New, quindi



3

Nome della Classe il nome da assegnare alla classe che contiene la finestra, in questo caso lasceremo il nome che appare normalmente (NewJFrame), quindi facciamo click su Finish: ➤

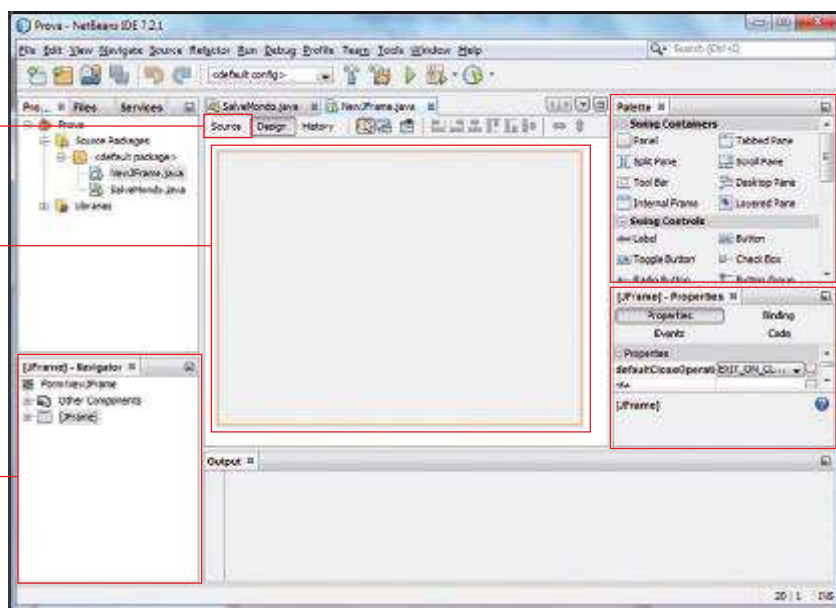


4

Codice sorgente

Area di disegno

Area di navigazione



Componenti Swing trascinabili

Proprietà oggetto selezionato

#### ► Area di navigazione

Sono presenti tutti gli oggetti che sono stati aggiunti nella nostra interfaccia grafica. Essi vengono organizzati ad albero dove un oggetto nodo può identificare un contenitore nel quale sono presenti altri oggetti. Questa finestra mette in luce la struttura dell'interfaccia grafica che stiamo realizzando. Inoltre semplifica la selezione di un determinato oggetto, soprattutto quando il progetto ne contiene molti.

#### ► Area di disegno

Si tratta di una finestra nella quale possiamo trascinare gli oggetti mediante una semplice operazione di drag and drop. Inoltre la finestra mostra il layout degli oggetti così come verranno visualizzati durante l'esecuzione del programma secondo la logica Wysiwyg (**What You See Is What You Get**).

#### ► Componenti Swing trascinabili

Questa finestra si chiama **Palette** e mostra l'elenco di tutti i componenti grafici (oggetti di classe `JSwing`) che possiamo utilizzare. Vengono mostrati suddivisi per categorie, per utilizzarli è sufficiente selezionare quello che ci interessa per poi trascinarlo nell'area disegno.

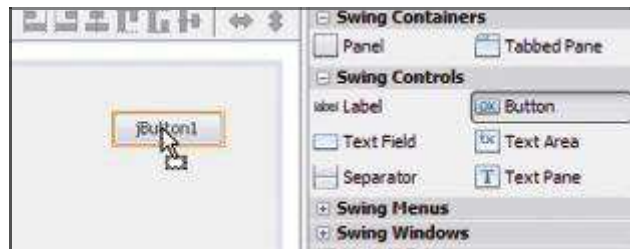
#### ► Proprietà oggetto selezionato

Consente di visualizzare e modificare tutte le proprietà, disponibili in fase di progettazione, di un determinato oggetto selezionato e di gestire gli eventi a esso associato. Come si può notare in quest'area sono presenti delle voci (**Properties, Binding, Events, Code**) che suddividono le caratteristiche dell'oggetto in esame.

#### ► Codice sorgente

Facendo click su questa sezione possiamo visualizzare il codice legato alla finestra.

5



6

**Palette**, nella categoria **Swing Controls**, selezioniamo e trasciniamo nell'area disegno un oggetto **Button** e un oggetto **Label**: ►



Possiamo ridimensionare un oggetto, una volta selezionato, cliccando e trascinando uno dei ganci che lo circonda, come nella figura: ►



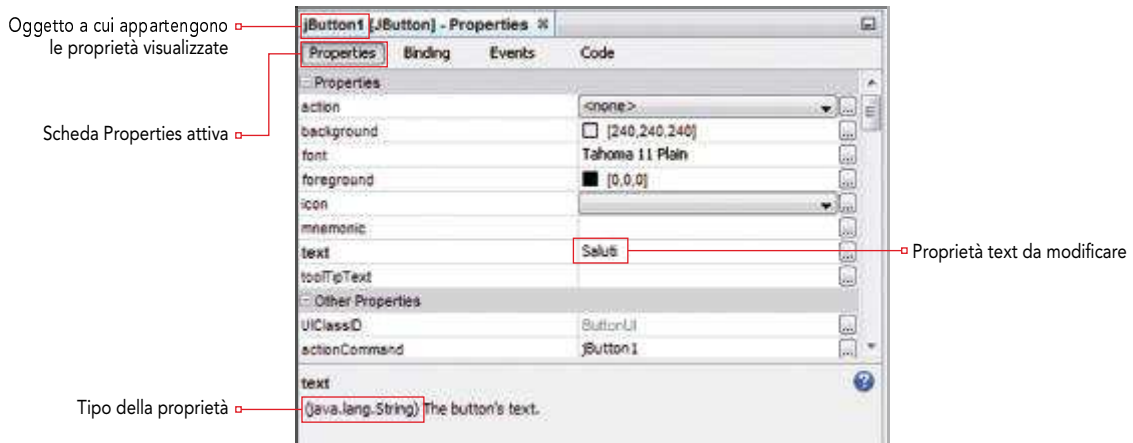
7

area di navigazione o direttamente dall'area disegno, tenendo presente che per selezionare gli oggetti nell'area disegno, il pulsante **Selection Mode** deve essere selezionato: ►



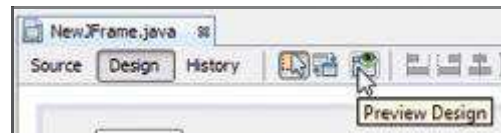
8

Properties sia attiva. Modifichiamo la proprietà text che consente di cambiare il testo che appare sul pulsante:

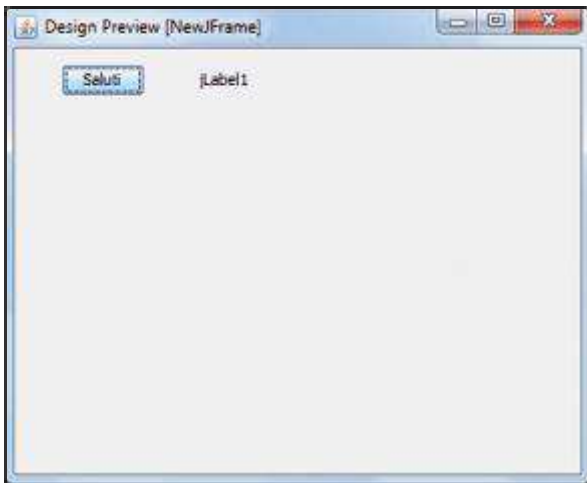


9

jButton1. Per vedere il risultato e ottenere così una anteprima dell'output del programma facciamo click su Preview Design: ►



Otteniamo la seguente finestra:



Preview Design mostra solo l'aspetto grafico della finestra di classe JFrame e quindi ogni eventuale funzionalità dei componenti dell'interfaccia, implementate col codice, non sono eseguite, l'unica eccezione è legata alla funzionalità della finestra come per esempio la chiusura e il ridimensionamento.

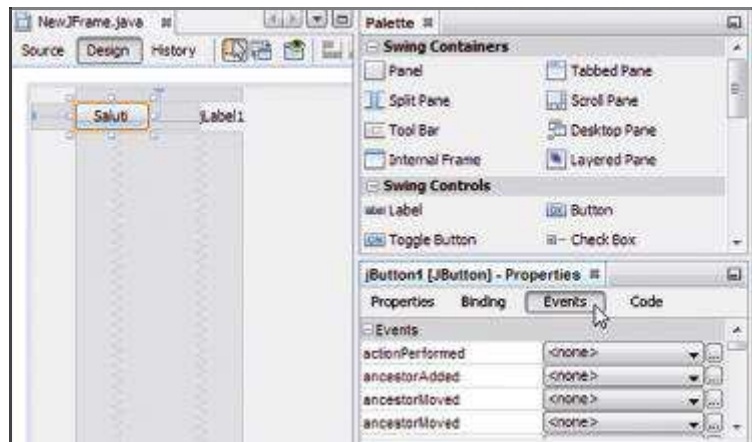
Adesso abbiamo solo imparato come inserire alcuni oggetti e modificarne le proprietà. Nel paragrafo seguente vedremo come associare l'evento e il codice relativo.

## ■ Associare il codice agli elementi grafici

In questo caso il codice che intendiamo associare modificherà il contenuto dell'etichetta jLabel1, alla pressione del pulsante jButton1 inseriremo un testo nell'etichetta. Per fare questo dobbiamo gestire l'evento actionPerformed del pulsante, inserendo un ascoltatore di eventi. Tuttavia NetBeans, lo farà automaticamente, come mostrato dalla procedura che segue:

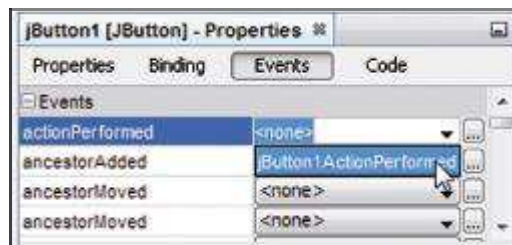
1

Properties attiviamo la scheda Events. ▶



2

actionPerformed, dobbiamo quindi selezionare l'oggetto da associare all'evento semplicemente facendo click su di esso: ▶



3

jButton1ActionPerformed()) a cui associare l'evento actionPerformed, che verrà eseguita ogni qualvolta si farà click sul pulsante jButton1:



La stessa operazione, cioè quella di associare l'evento al pulsante selezionato, la possiamo ottenere facendo doppio click sul pulsante.

4

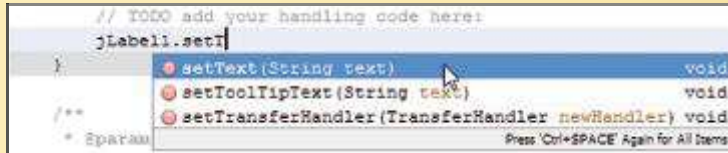
text dell'oggetto jLabel1, attraverso l'invocazione del metodo setText():

```

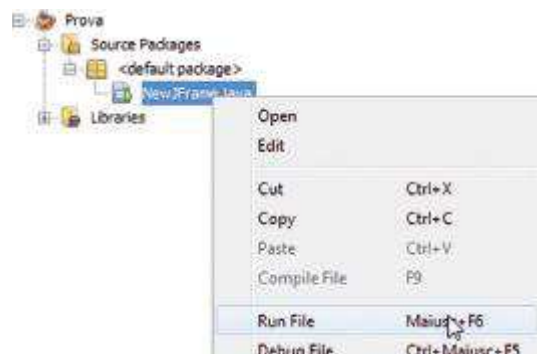
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jLabel1.setText("Hai fatto click sul pulsante: ciao!");
}

```

NetBeans offre un valido strumento chiamato **intellisense** che consente di avere un elenco dei metodi e delle proprietà dell'oggetto che stiamo digitando. Basta infatti scrivere il nome dell'oggetto seguito dal punto e automaticamente apparirà un elenco di metodi e proprietà dai quali selezionare la voce interessata facendo click su di essa:



5. Per eseguire il file dobbiamo fare click con il tasto destro del mouse sul file del programma (**NewJFrame.java**), quindi selezionare la voce **Run File**: ▶



6. Saluti otteniamo una scritta nell'etichetta posta accanto: ▶



7. **jLabel1**, per cancellarla dobbiamo eliminare la scritta di default dell'etichetta, che può essere effettuata in due modi:
- ▶ attraverso la scheda **Properties** dell'oggetto in fase di progettazione;
  - ▶ attraverso il costruttore della classe **NewJFrame** per modificarla in fase di **run time** a ogni esecuzione.

In questo caso proviamo la seconda strada per comprendere come scrivere nel costruttore della classe. Scriviamo nel costruttore il seguente codice:

```

10 public class NewJFrame extends javax.swing.JFrame {
11
12     /**
13      * Creates new form NewJFrame
14      */
15     public NewJFrame() {
16         initComponents();
17         jLabel1.setText(null);
18     }

```

Che sostanzialmente disattiva la scritta, associando a essa un valore **null**.

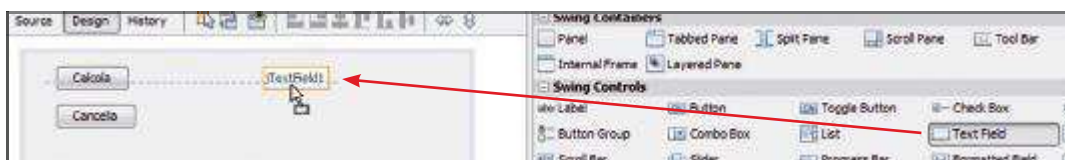
Possiamo vedere che nel costruttore appare la chiamata a un metodo chiamato  `initComponents()`. Questo metodo viene generato automaticamente dal programma e contiene tutto il codice dell'interfaccia grafica che abbiamo costruito nell'area disegno; il codice non può essere modificato direttamente ma solo attraverso l'area delle proprietà dei singoli oggetti.

```
public class NewJFrame extends javax.swing.JFrame {
    public NewJFrame() {
        initComponents();
        jLabel1.setText(null);
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        // ...
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        jLabel1.setText("Has fatto click sul pulsante: ciao!");
    }
    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if (" Nimbus ".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException | InstantiationException | IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
    }
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new NewJFrame().setVisible(true);
        }
    });
}
```

Costruttore: richiama il metodo  `initComponents()`. Qui dobbiamo impostare le proprietà iniziali del Frame

### ESEMPIO 16 *Una applicazione GUI: calcolo dell'area del rettangolo*

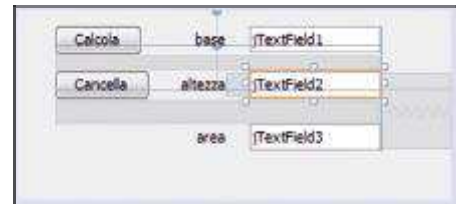
In questo esempio utilizziamo due pulsanti di classe  `Button` (Calcola e Cancella) e tre caselle di testo di classe  `TextField` per leggere base e altezza e visualizzare il risultato. La classe  `TextField` rappresenta una casella di testo ed è presente nella finestra laterale chiamata  `Palette` dove sono presenti anche tutti gli altri elementi grafici. Per inserire due caselle di testo non dobbiamo fare altro che trascinarle all'interno del form:



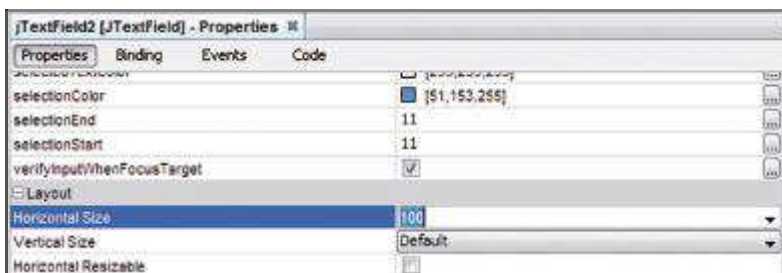
Dopo aver aggiunto gli oggetti caselle di testo, dobbiamo inserire alla proprietà  `text` il valore  `null` per evitare che mostrino il valore di default, assai poco gradevole esteticamente. Per fare questo scriviamo le seguenti righe di codice nel metodo costruttore della classe  `NewJFrame`:

```
public NewJFrame() {
    initComponents();
    jTextField1.setText(null);
    jTextField2.setText(null);
    jTextField3.setText(null);
}
```

Aggiungiamo adesso anche tre etichette (**Label**) per associare una scritta accanto alle caselle di testo, dopo aver modificato la proprietà delle tre etichette, sostituendo per esempio a `jLabel1` il testo “base”: ▶



Le caselle di testo devono essere dimensionate opportunamente per evitare che abbiano una dimensione troppo ridotta. Per fare questo dobbiamo agire sulla proprietà **Horizontal Size**, assegnando un valore pari a 100:



Una volta inseriti tutti gli oggetti associamo al primo pulsante (`jButton1`) il codice seguente:

```

121 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
122     //dichiarazione delle tre variabili double
123     double base, altezza, area;
124     //lettura della casella di testo che contiene la misura della base
125     base = Double.parseDouble(jTextField1.getText());
126     //lettura dalla casella di testo che contiene la misura dell'altezza
127     altezza = Double.parseDouble(jTextField2.getText());
128     //calcolo della misura della base
129     area = base*altezza;
130     //assegnazione risultato alla casella di testo dell'area
131     jTextField3.setText(Double.toString(area));
132 }

```

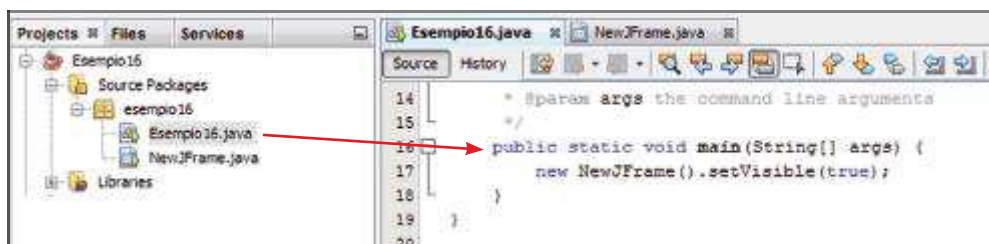
Associamo al secondo pulsante (`jButton2`) il codice seguente:

```

138 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
139     //cancella il testo delle tre caselle di testo
140     jTextField1.setText("");
141     jTextField2.setText("");
142     jTextField3.setText("");
143 }
144

```

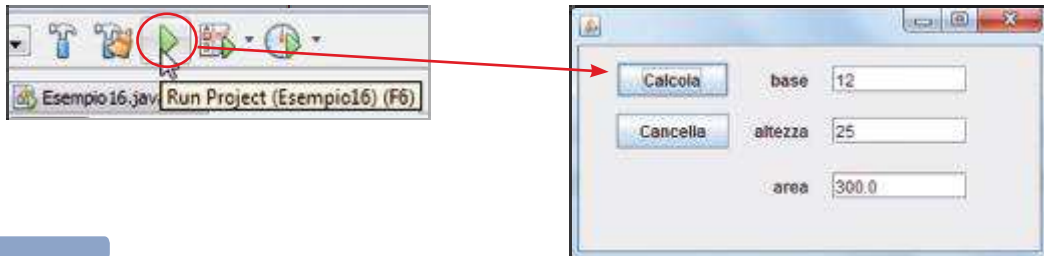
Infine per fare in modo che la finestra, presente nella classe `NewJFrame`, sia mostrata avviando il pulsante `Run Project`, dobbiamo scrivere il codice seguente nel `main` della classe che utilizzerà la finestra:





NewJFrame.

Adesso eseguiamo il nostro programma facendo click sul pulsante Run Project, l'esecuzione è mostrata di seguito:



**Prova adesso!**



**APRI IL PROGETTO** [Esercizio2](#)

- Creare un progetto GUI con NetBeans
- Aggiungere oggetti al frame
- Utilizzare Button, Label e TextField

- 1 Modifica il progetto in modo tale che sia in grado di convertire anche le temperature da gradi Fahrenheit a gradi Celsius.
- 2 Confronta la tua soluzione con quella presente nel progetto [Esercizio2\\_solux](#).

## ■ L'oggetto TextArea

Si tratta di una **lista di classe** `TextArea` che consente di contenere più righe di testo all'interno dello stesso elemento. È importante sottolineare che per aggiungere righe all'interno di una lista si usa il metodo `append()`. L'esempio seguente mostra come utilizzarlo per stampare a video un elenco di numeri.

### ESEMPIO 17 *Numeri primi*

Nell'esempio proposto vi è un metodo che riceve come argomento un numero `n` e restituisce un valore `true` o `false` a seconda che esso sia primo oppure no. Il metodo va inserito subito dopo **Generate Code**, come mostrato di seguito:

```
Generated Code
//metodo che restituisce true se il parametro n è primo
//o restituisce false se il parametro n non è primo
boolean numeroPrimo(int n) {
    int conta = 2;
    while (n%conta!=0 && conta<n) {
        conta++;
    }
    if (conta==n) {
        return true;
    }
    else {
        return false;
    }
}
```