

CALCULATOR PROJECT

Try it

BUAT PROJECT BARU

Name and Location

Project Name:

Project Location:

Project Folder:

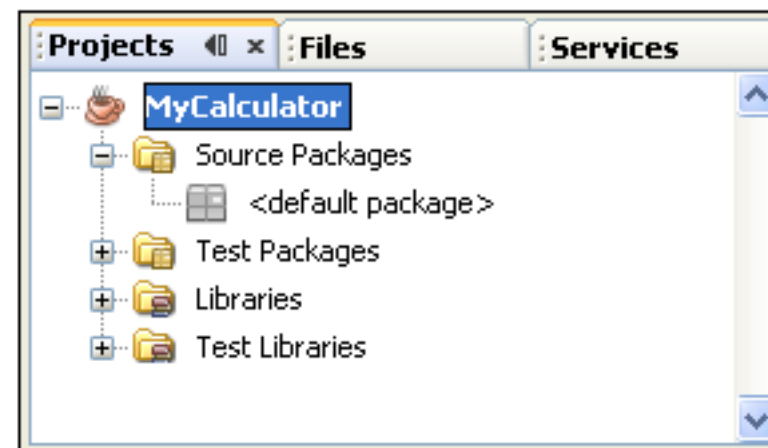
Use Dedicated Folder for Storing Libraries

Libraries Folder:

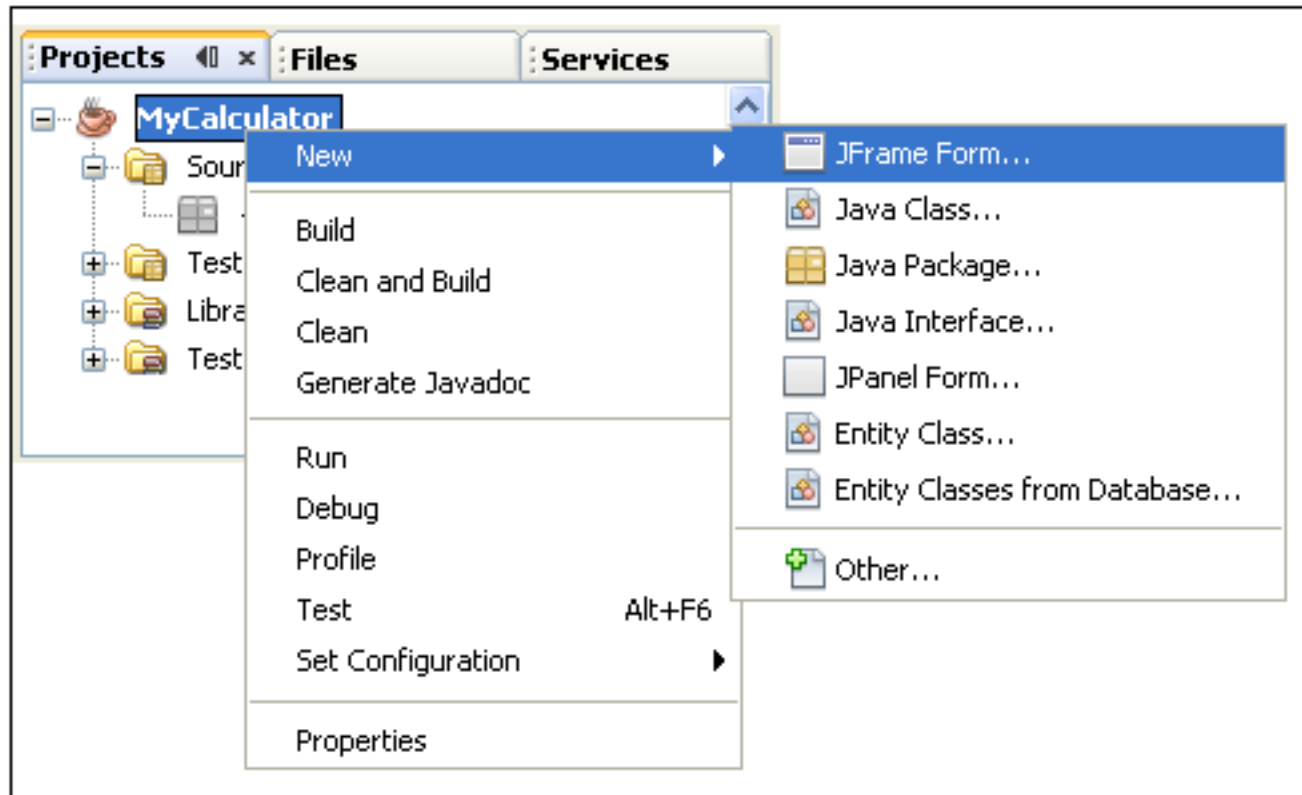
Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class

Set as Main Project



BUAT JFrame



Name and Location

Class Name:

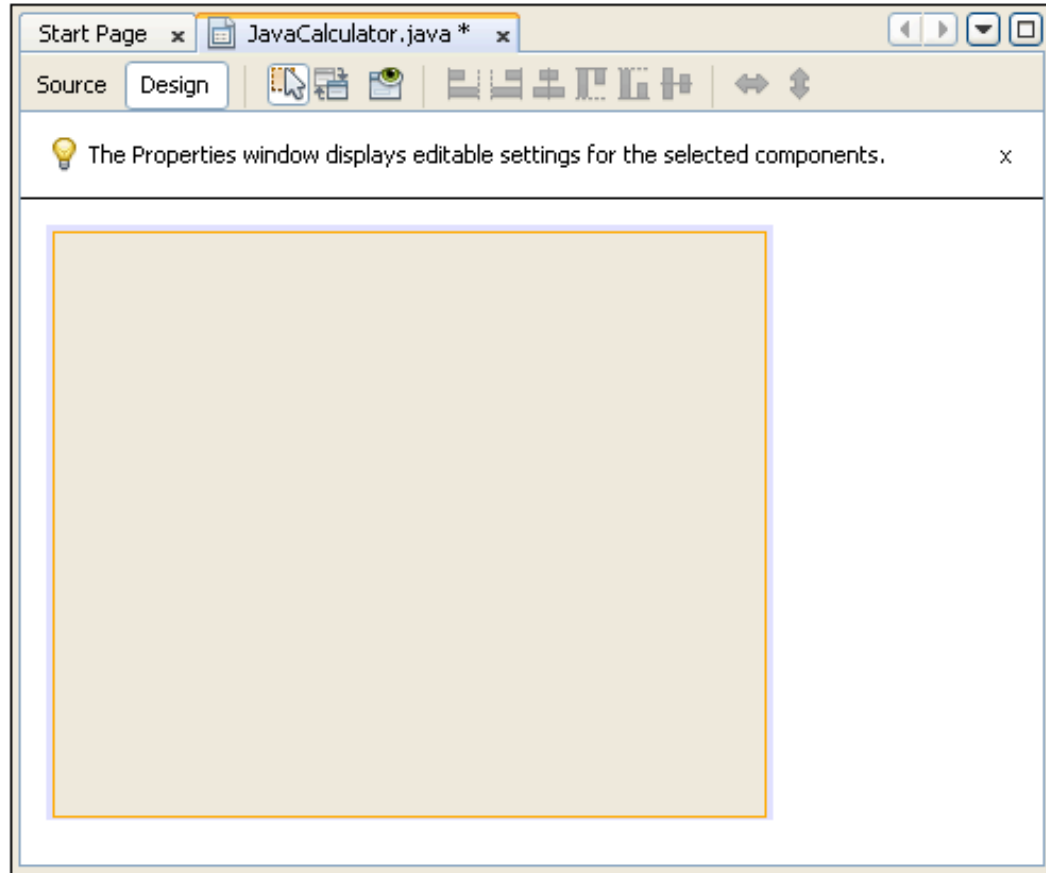
Project:

Location:

Package:

Created File:

JFRAME FORM



```
package jCalculator;

+ /**...*/
public class JavaCalculator extends javax.swing.JFrame {

- /** Creates new form JavaCalculator */
- public JavaCalculator() {
    initComponents();
}

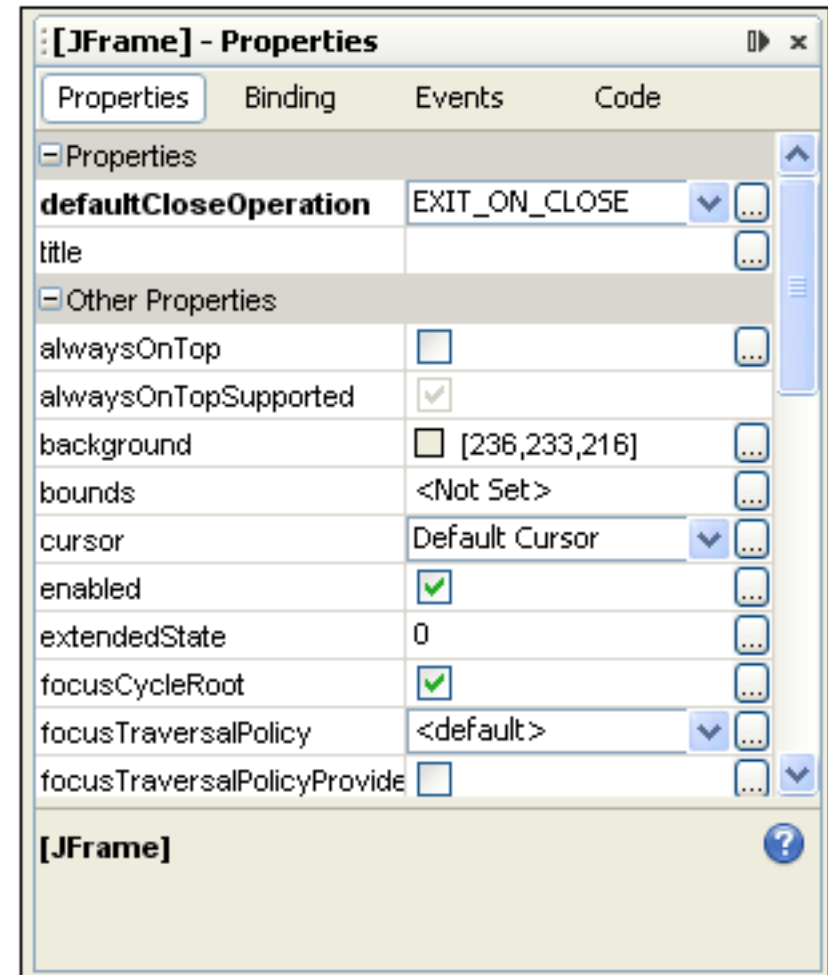
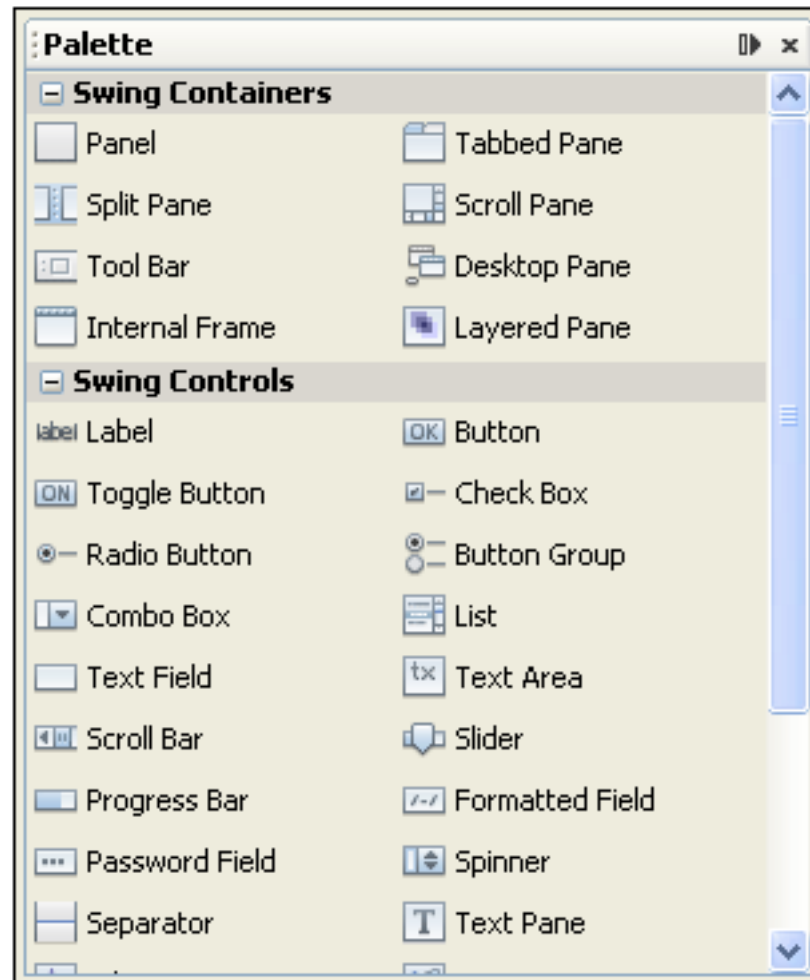
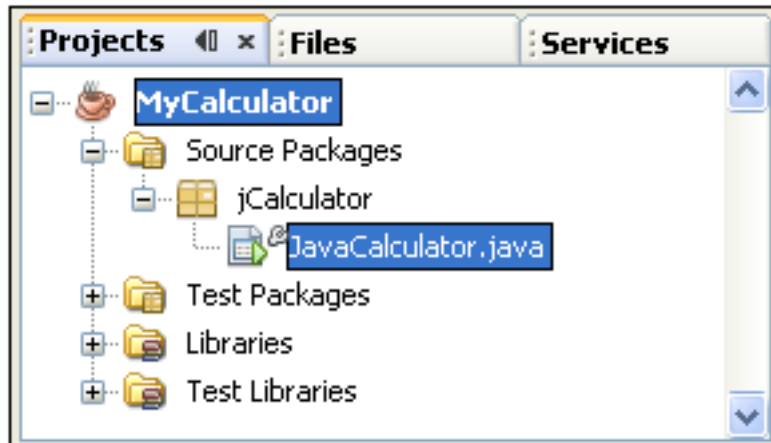
+ /**...*/
+ @SuppressWarnings("unchecked")
+ Generated Code

+ /**...*/
- public static void main(String args[]) {
-     java.awt.EventQueue.invokeLater(new Runnable() {
-         public void run() {
-             new JavaCalculator().setVisible(true);
-         }
-     });
}

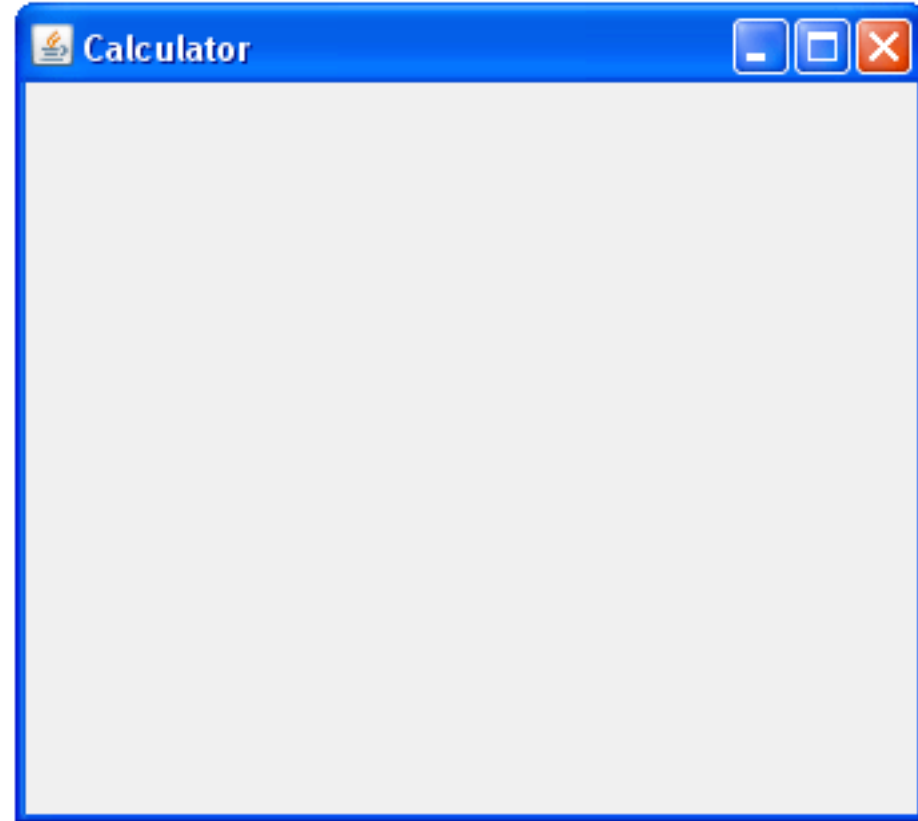
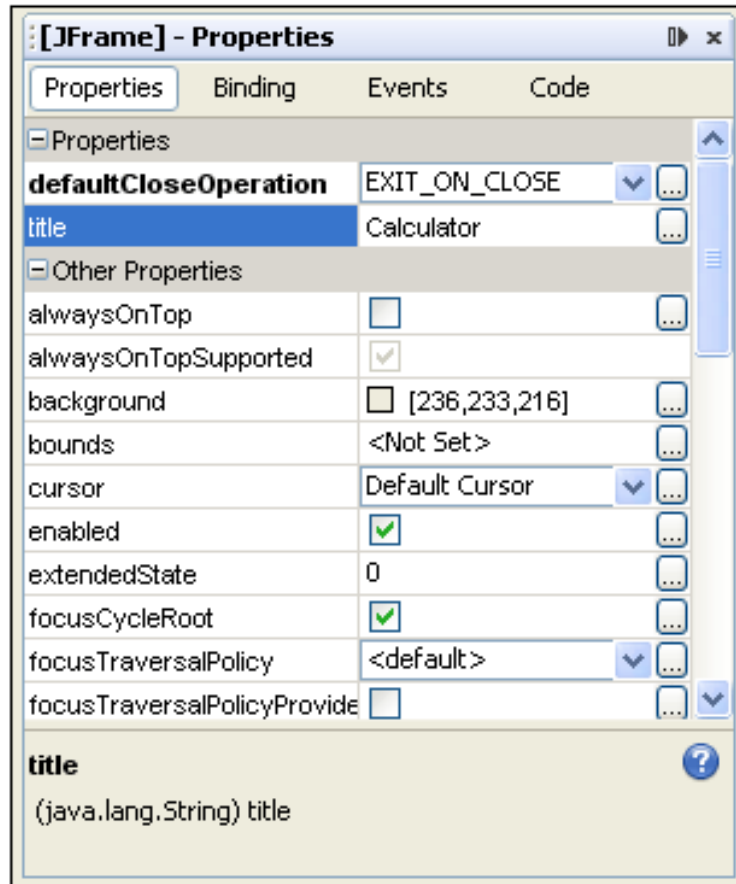
// Variables declaration - do not modify
// End of variables declaration

}
```

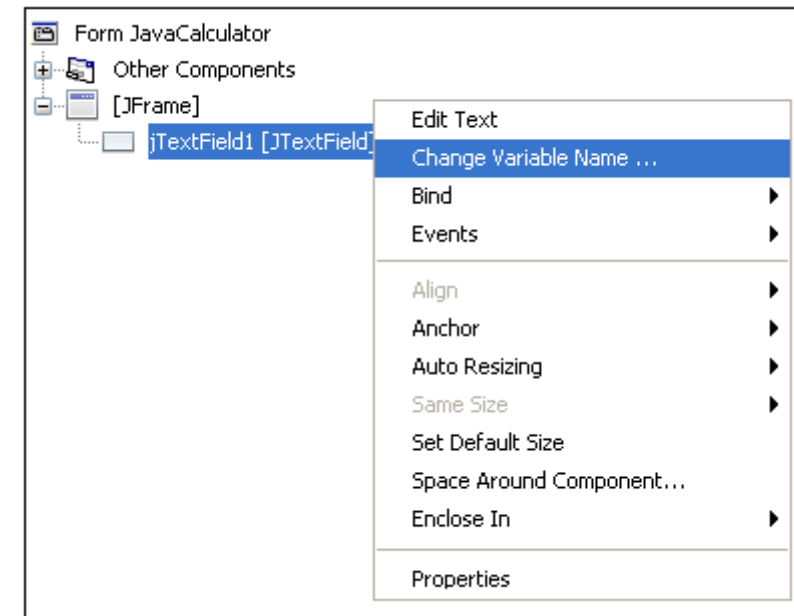
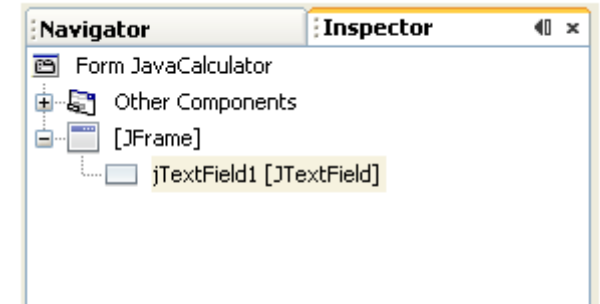
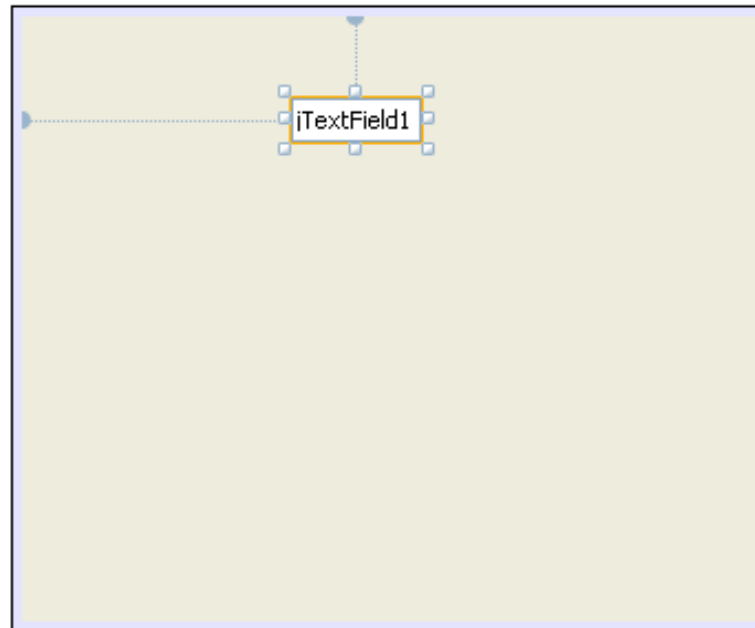
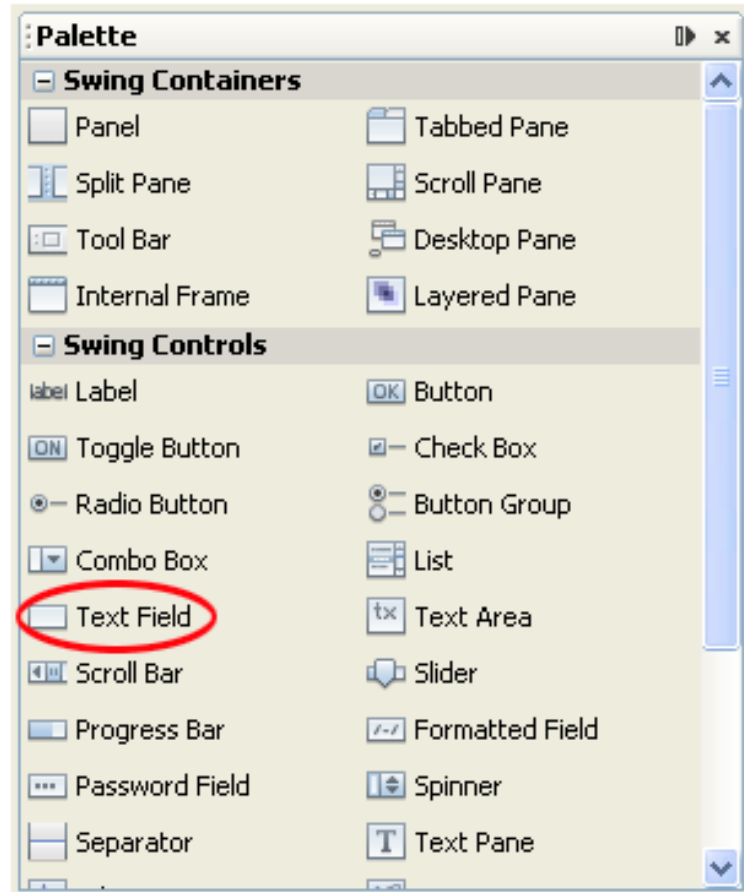
PROPERTIES



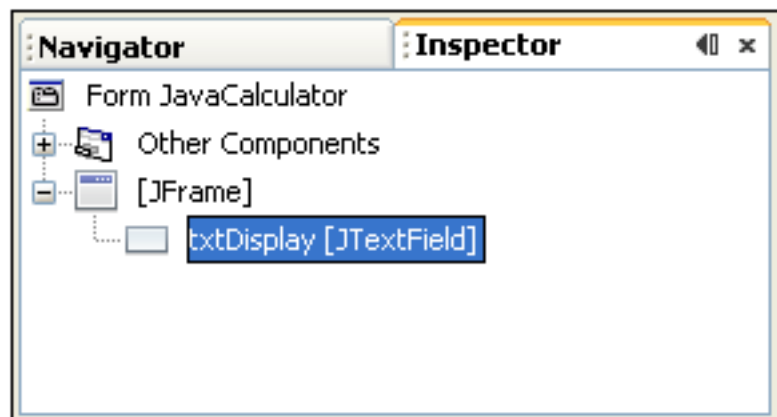
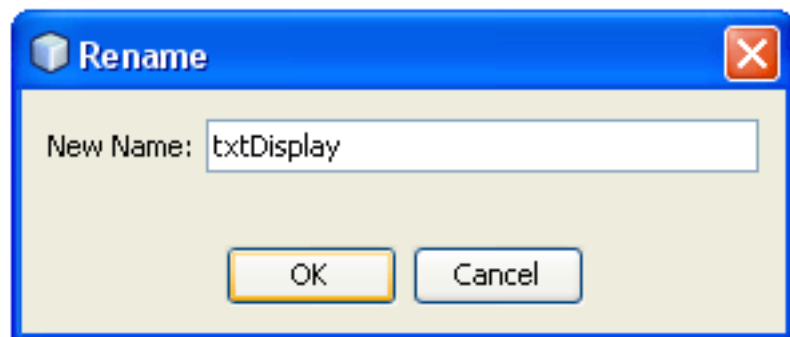
SET THE TITLE



TAMBAHKAN TEXT FIELD



TAMBAHKAN TEXT FIELD



```
/**...*/  
public static void main(String args[]) {  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new JavaCalculator().setVisible(true);  
        }  
    });  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JTextField txtDisplay;  
// End of variables declaration
```


TAMBAHKAN TEXT FIELD

txtDisplay [JTextField] - Properties

Properties Binding Events Code

Properties

background	<input type="checkbox"/> [255,255,255]	...
columns	0	...
document	<default>	...
editable	<input checked="" type="checkbox"/>	...
font	Tahoma 11 Plain	...
foreground	■ [0,0,0]	...
horizontalAlignment	LEADING	...
text	TextField1	...
toolTipText	null	...

Other Properties

UI	<default>	...
UIClassID	TextFieldUI	...
action		...

text (java.lang.String) the text of this component

txtDisplay [JTextField] - Properties

Properties Binding Events Code

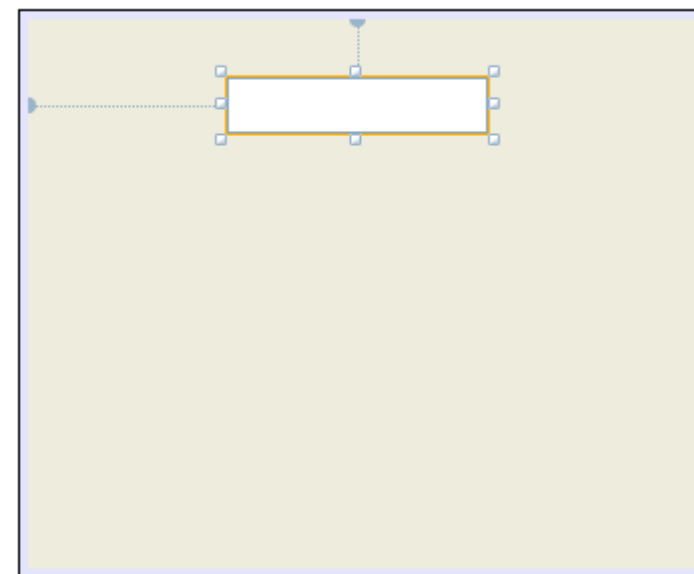
Properties

background	<input type="checkbox"/> [255,255,255]	...
columns	0	...
document	<default>	...
editable	<input checked="" type="checkbox"/>	...
font	Tahoma 11 Plain	...
foreground	■ [0,0,0]	...
horizontalAlignment	LEADING	...
text		...
toolTipText	null	...

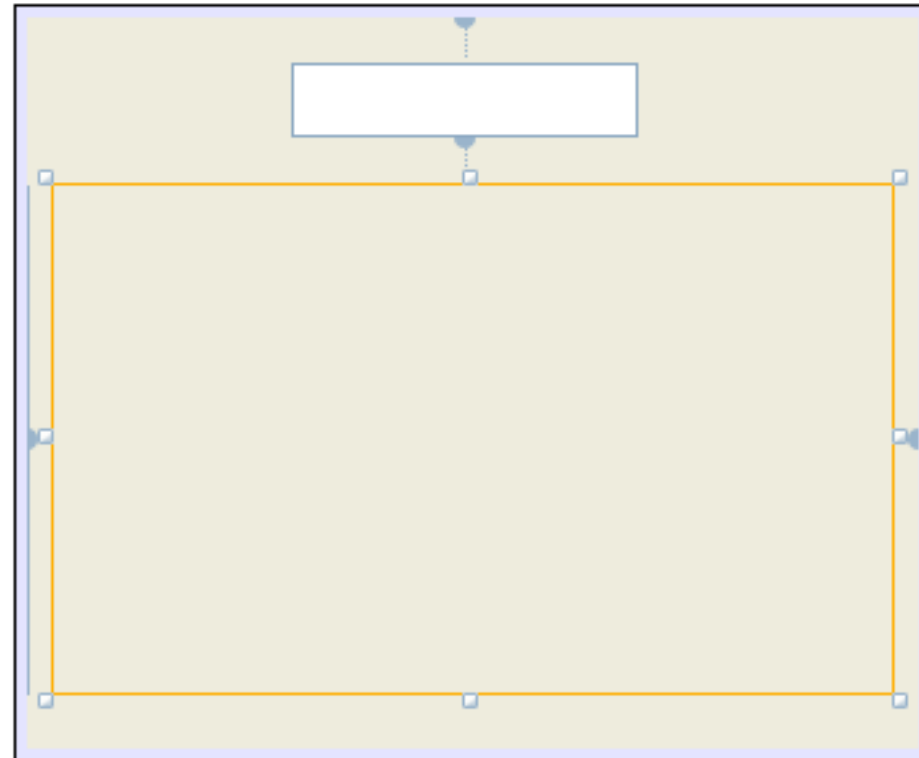
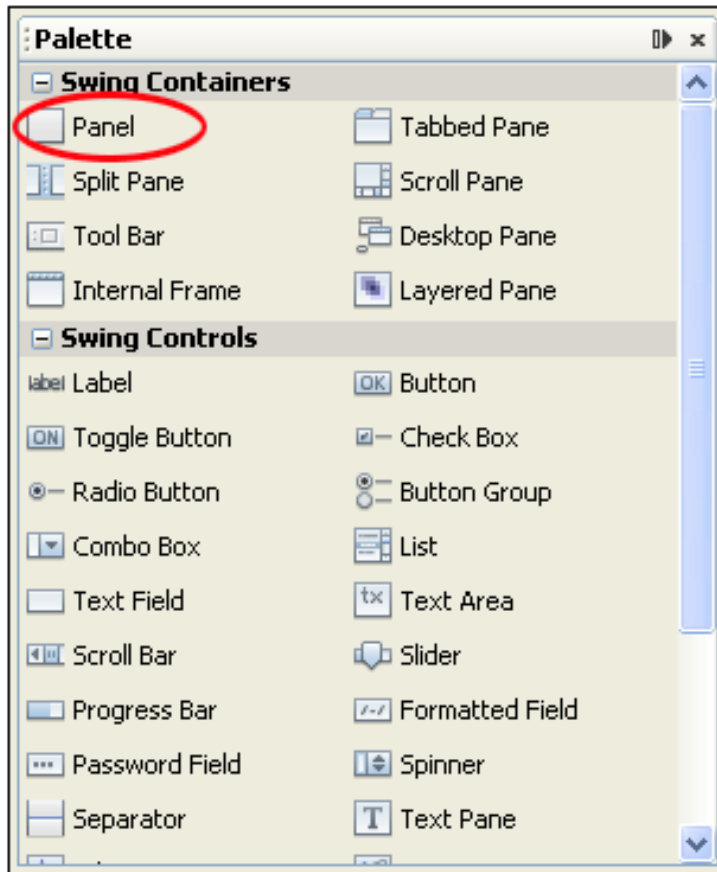
Other Properties

UI	<default>	...
UIClassID	TextFieldUI	...
action		...

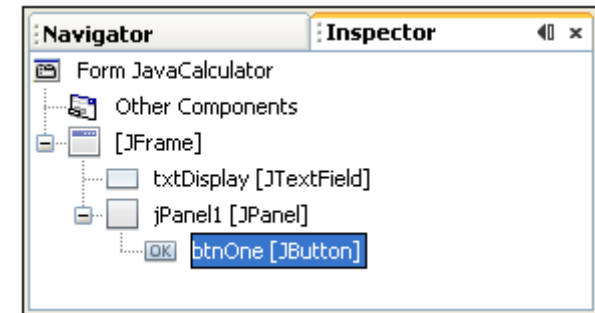
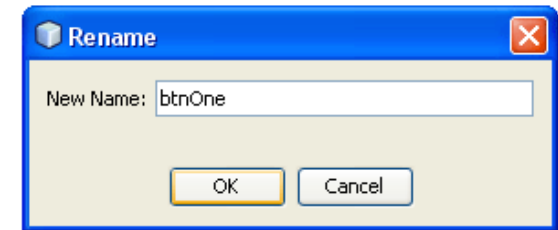
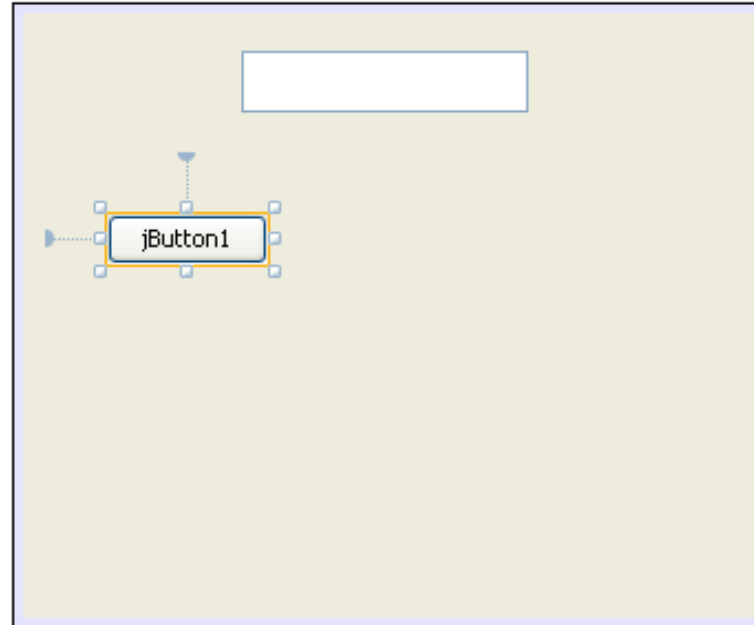
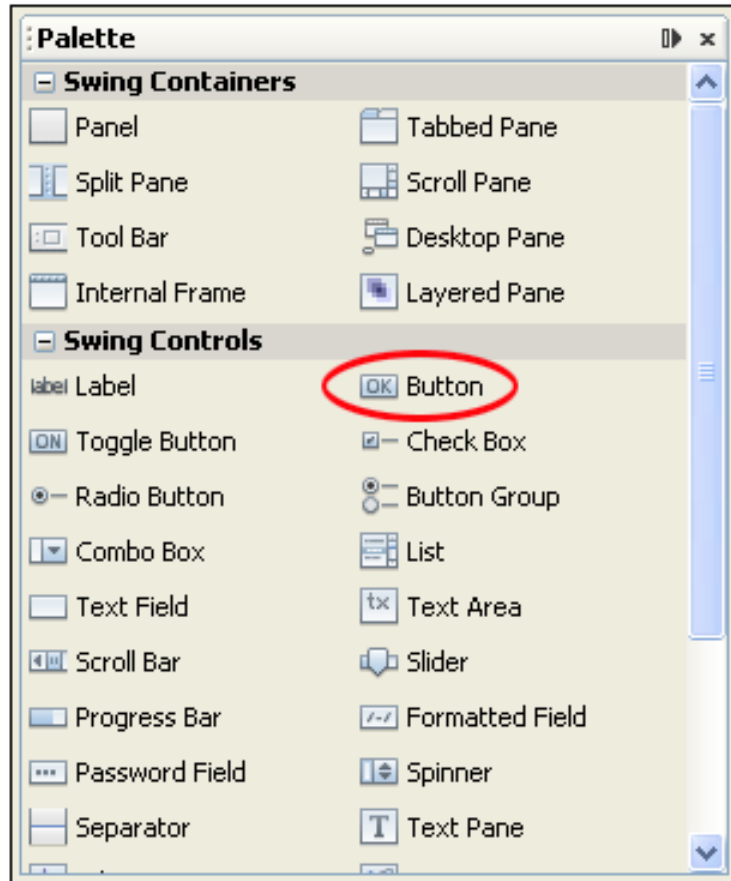
text (java.lang.String) the text of this component



TAMBAHKAN PANEL



TAMBAHKAN BUTTON



```
// Variables declaration - do not modify
private javax.swing.JButton btnOne;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField txtDisplay;
// End of variables declaration
```

UBAH PROPERTIES

btnOne [JButton] - Properties

Properties Binding Events Code

Properties

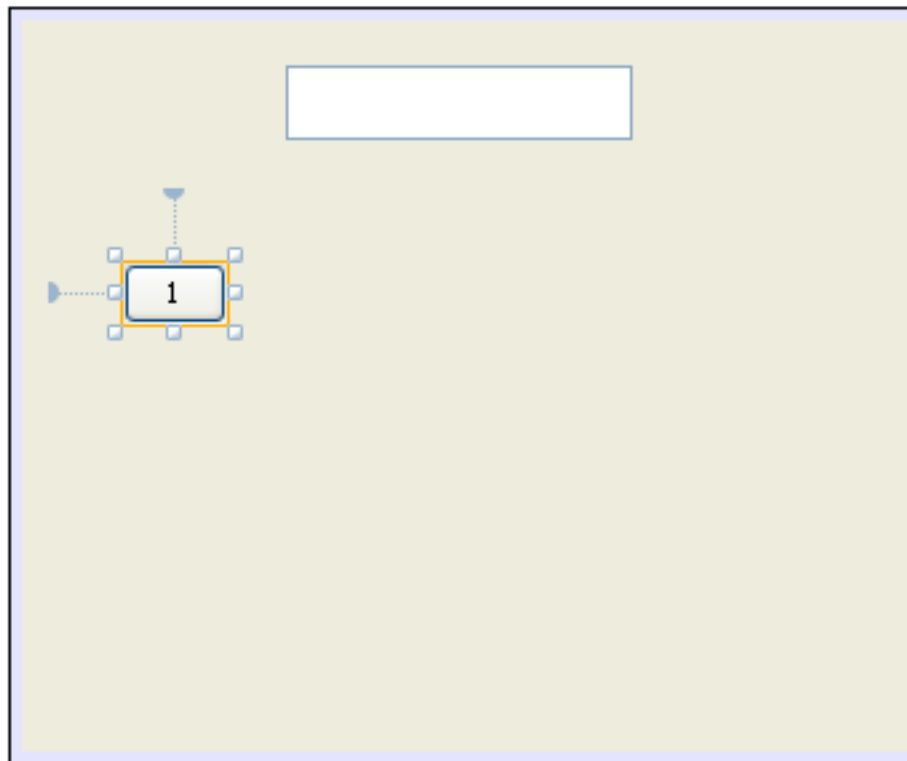
action		...
background	<input type="checkbox"/> [236,233,216]	...
font	Tahoma 11 Plain	...
foreground	<input type="checkbox"/> [0,0,0]	...
icon		...
mnemonic		...
text	1	...
toolTipText	null	...

Other Properties

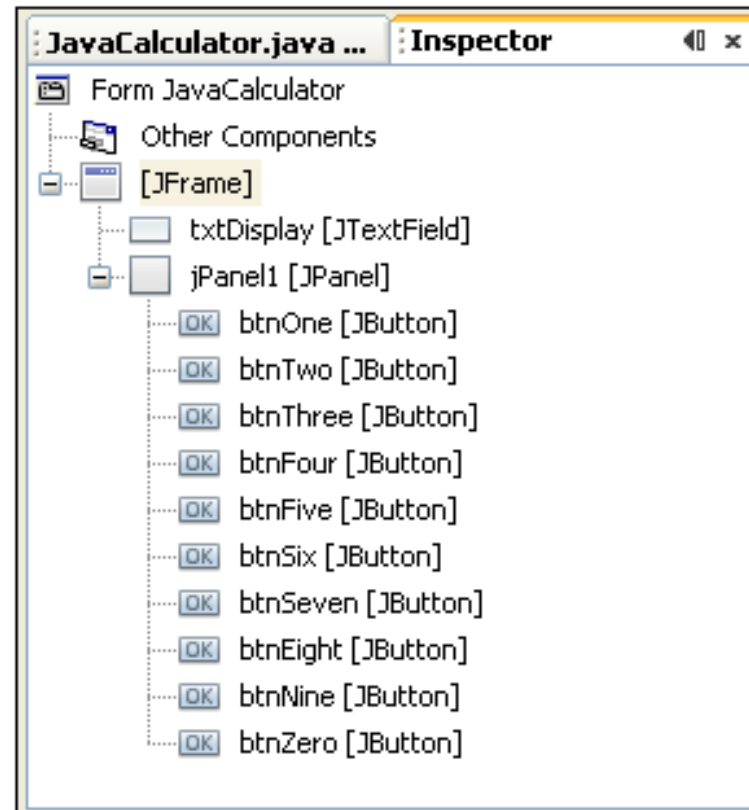
UIClassID	ButtonUI	...
actionCommand	1	...
alignmentX	0.0	...
alignmentY	0.5	...

text

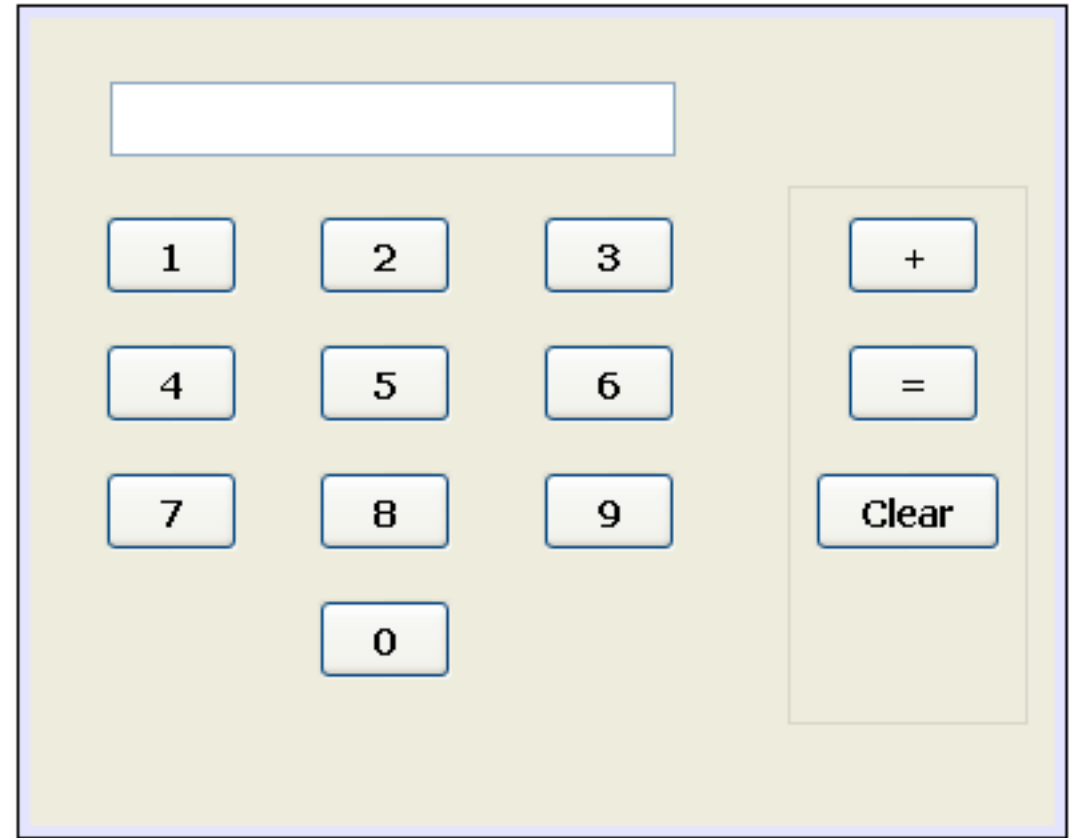
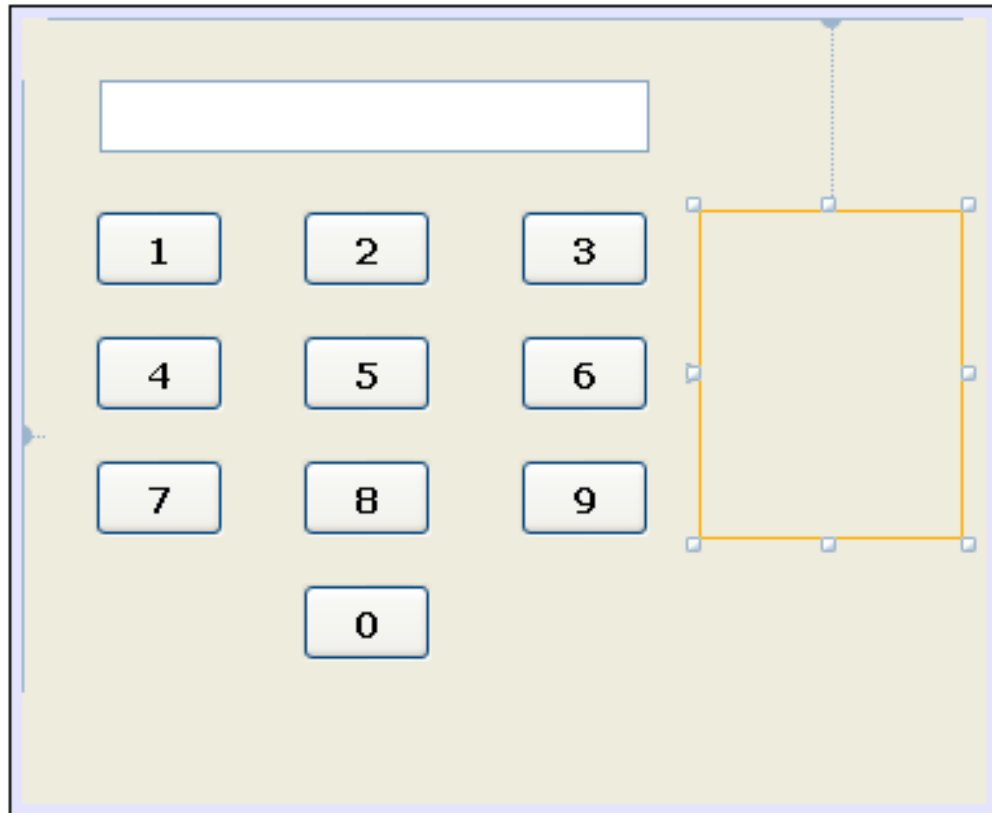
(java.lang.String) The button's text.



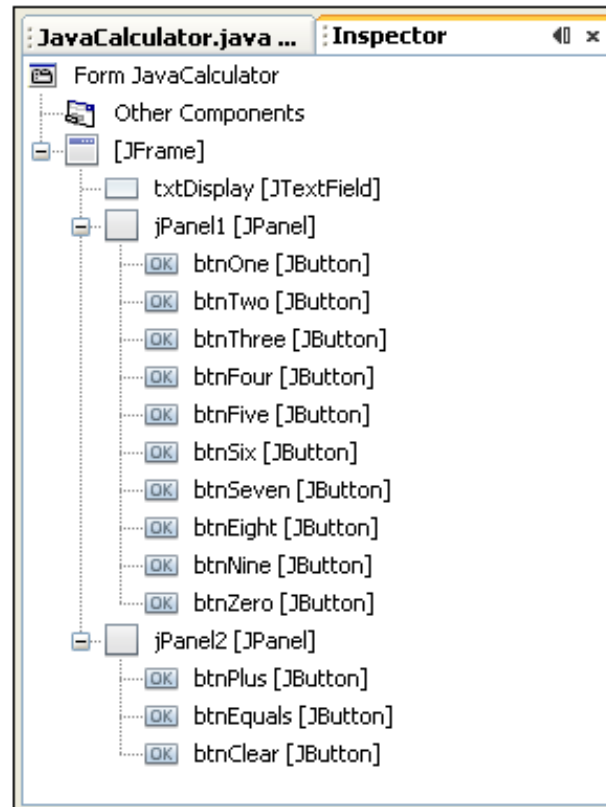
CEK INSPECTOR AREA



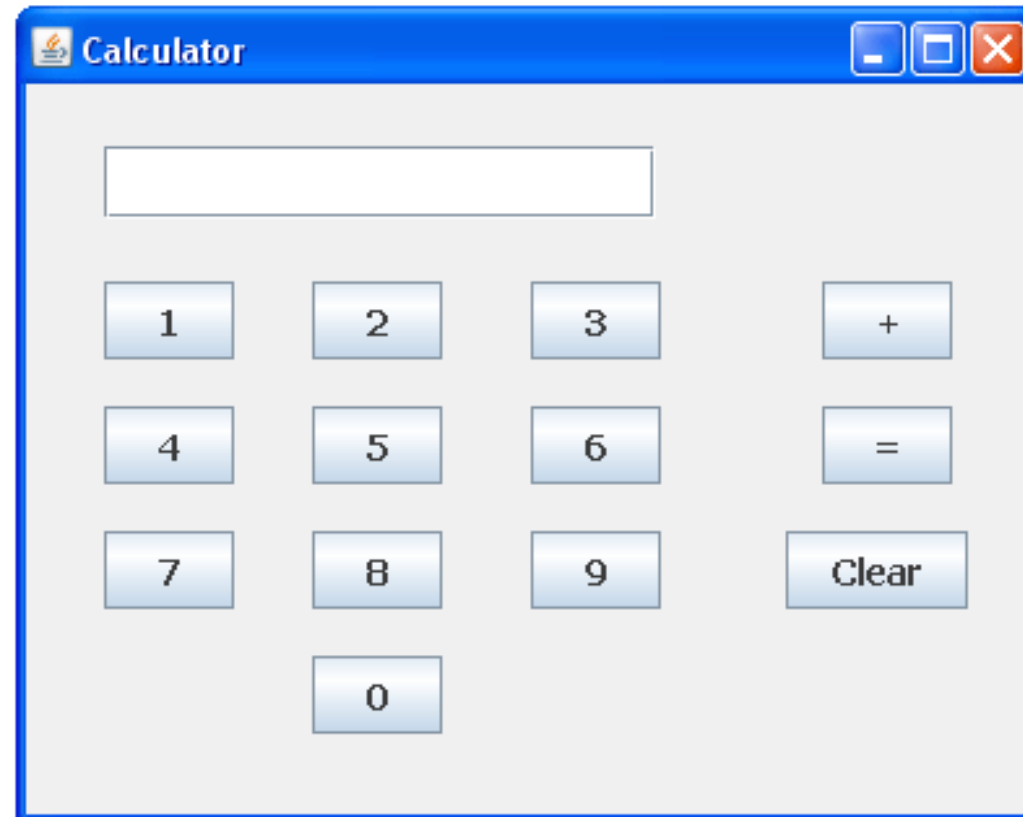
BUAT PANEL BARU



CEK INSPECTOR AREA



TAMPILAN GUI



MULAI MEMBUAT PROGRAM

The image illustrates the steps to add an event listener to a button in a Java Swing application. It shows the IDE's Navigator and Inspector panels. The Navigator displays a tree structure of components, with `btnOne` selected. The Inspector shows the 'Events' menu for `btnOne`, with the 'Action' event selected. A blue arrow points from the 'Action' event to the 'actionPerformed' event. Another blue arrow points from the 'actionPerformed' event to a code snippet.

```
private void btnOneActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

PENDAHULUAN

Untuk mengambil text dari button

```
String btnOneText = btnOne.getText( );
```

Untuk mengambil text dari text field

```
String textfieldText = txtDisplay.getText( );
```

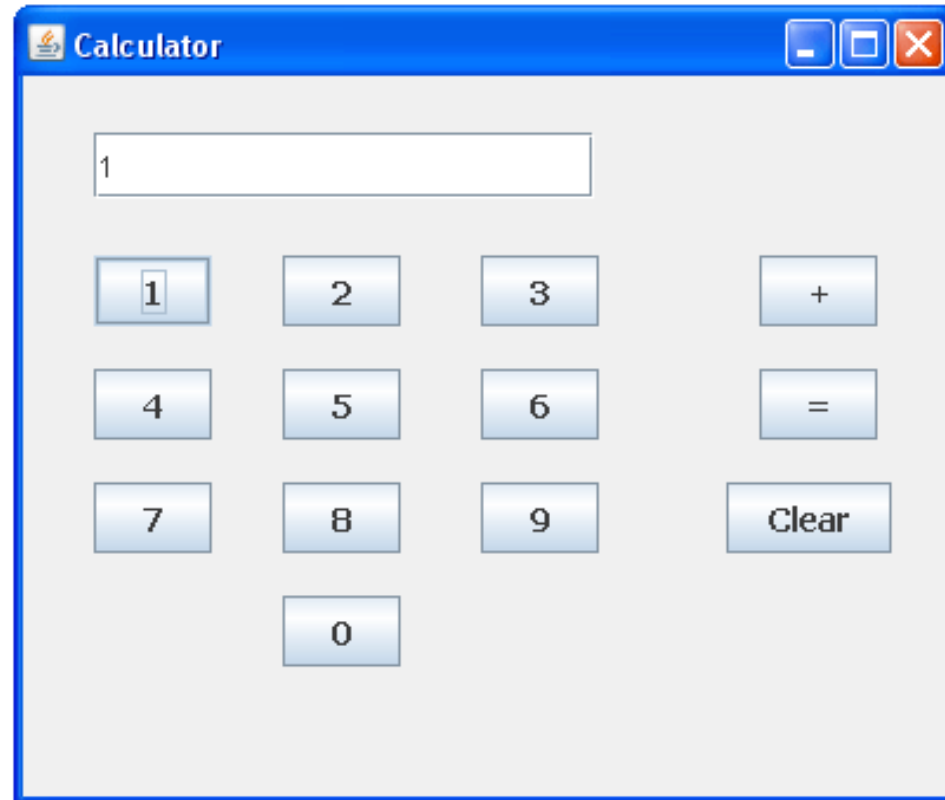
Untuk merubah text di text field menjadi text dari button

```
txtDisplay.setText( btnOneText );
```

Tuliskan di actionPerformed →

```
private void btnOneActionPerformed(java.awt.event.ActionEvent evt) {  
    String btnOneText = btnOne.getText( );  
    txtDisplay.setText(btnOneText);  
}
```

HASIL RUN



DISPLAY ANGKA BERIKUTNYA

Ubah program di actionPerformed menjadi

```
private void btnTwoActionPerformed( java.awt.event.ActionEvent evt ) {  
  
    String btnTwoText = txtDisplay.getText() + btnTwo.getText();  
    txtDisplay.setText( btnTwoText );  
  
}
```

Lakukan hal yang sama di semua button

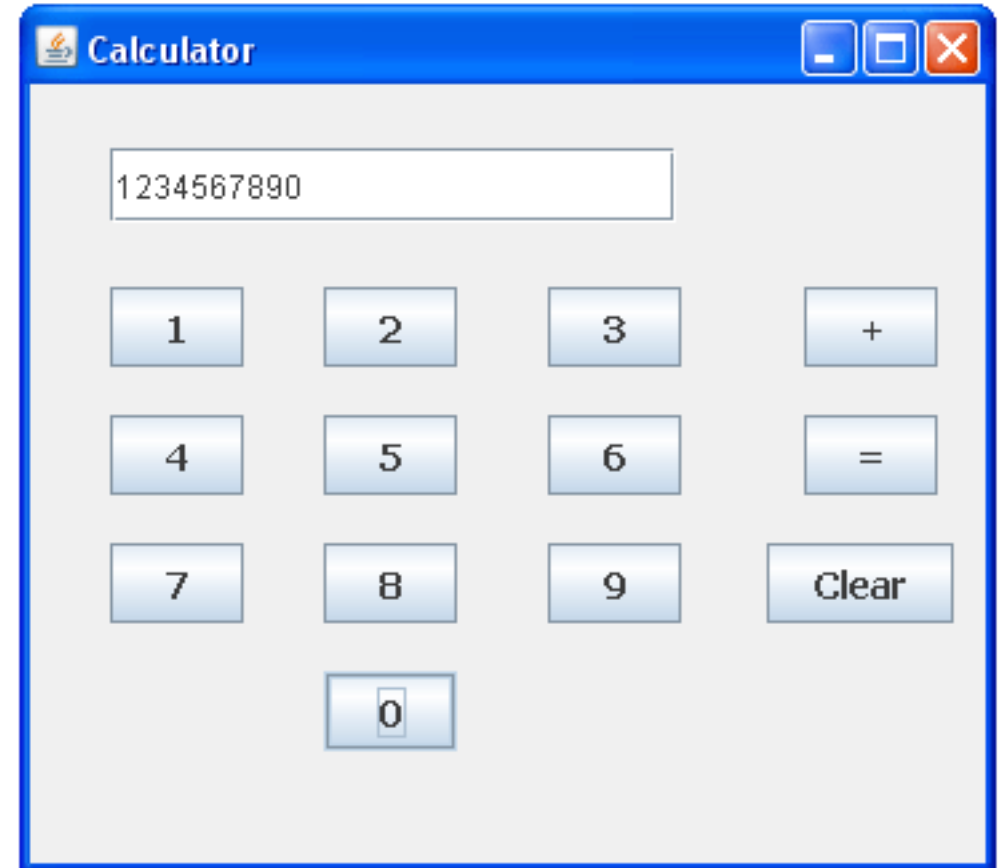
DISPLAY ANGKA BERIKUTNYA

```
private void btnOneActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String btnOneText = txtDisplay.getText() + btnOne.getText();  
    txtDisplay.setText(btnOneText);  
}
```

```
private void btnTwoActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String btnTwoText = txtDisplay.getText() + btnTwo.getText();  
    txtDisplay.setText(btnTwoText);  
}
```

```
private void btnThreeActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String btnThreeText = txtDisplay.getText() + btnThree.getText();  
    txtDisplay.setText(btnThreeText);  
}
```

```
private void btnFourActionPerformed(java.awt.event.ActionEvent evt) {  
    String btnFourText = txtDisplay.getText() + btnFour.getText();  
    txtDisplay.setText(btnFourText);  
}
```



TOMBOL +

Tambahkan variabel penampung di awal deklarasi

```
public class JavaCalculator extends javax.swing.JFrame {  
  
    private double total1 = 0.0;  
  
    /** Creates new form JavaCalculator */  
    public JavaCalculator() {  
        ...  
    }  
  
    /**...*/  
}
```

Klik 2x tombol “+” dari display GUI, kemudian tambahkan baris program berikut

```
total1 = total1 + Double.parseDouble( txtDisplay.getText( ) );  
txtDisplay.setText("");
```

TOMBOL “=”

Tambahkan variable “total2”

```
public class JavaCalculator extends javax.swing.JFrame {  
  
    private double total1 = 0.0;  
    private double total2 = 0.0;  
  
    public JavaCalculator() {  
        initComponents();  
    }  
}
```

Tambahkan baris program berikut di tombol “=”

```
total2 = total1 + Double.parseDouble( txtDisplay.getText( ) );  
txtDisplay.setText( Double.toString(total2) );  
total1 = 0;
```

TOMBOL “-”, “X”, “/”

Untuk proses ini gunakan statement if atau switch

Contoh kita gunakan switch

Letakan baris program berikut di actionPerformed tombol “=”

```
switch ( math_operator ) {  
    case '+':  
        total2 = total1 + Double.parseDouble(txtDisplay.getText());  
        break;  
    case '-':  
        total2 = total1 - Double.parseDouble(txtDisplay.getText());  
        break;  
    case '/':  
        total2 = total1 / Double.parseDouble(txtDisplay.getText());  
        break;  
    case '*':  
        total2 = total1 * Double.parseDouble(txtDisplay.getText());  
        break;  
}  
  
txtDisplay.setText(Double.toString(total2));  
total1 = 0;
```


JANGAN LUPA

Tambahkan variable char/string “math_operator” di deklarasi

```
public class JavaCalculator extends javax.swing.JFrame {  
  
    private double total1 = 0.0;  
    private double total2 = 0.0;  
    private char math_operator;  
  
    public JavaCalculator() {  
        initComponents();  
    }  
}
```

TOMBOL “CLEAR”

Untuk clear display, bisa dengan menggunakan baris program berikut

```
total2 = 0;  
txtDisplay.setText("");
```

FINISH

