13. GUI Programming :: The *JTextField* Class

In a GUI program, a **text field** is a component that displays a single line of text that can be edited by the user. To create a text field using Swing we use the *javax.swing.JTextField* class:

```
JTextField tf1 = new JTextField(15);
JTextField tf2 = new JTextField("This text appears in the text field.");
JTextField tf3 = new JTextField("Initial text", 30);
```

The first *JTextField*(**int**) constructor specifies the number of columns (i.e., width) used to determine the width of the text field; the text field will be empty.

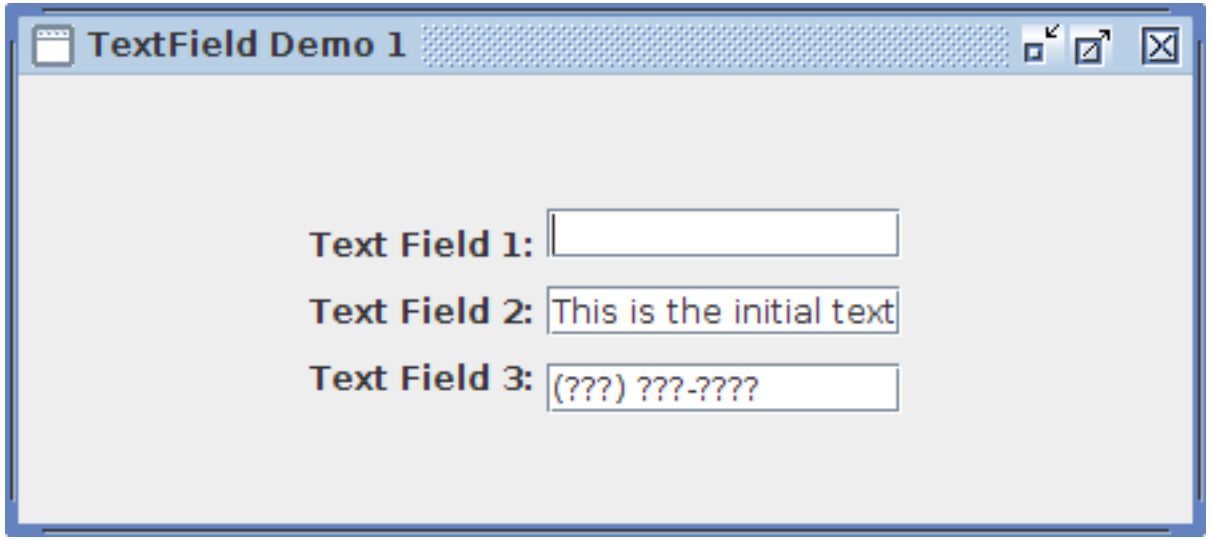
The second *JTextField*(*String*) constructor specifies the initial text to be displayed in the text field. The number of columns is based on the length of the text string argument.

The third *JTextField*(*String*, **int**) constructor is a combination of the first two; the text string will be displayed in the text field and the number of columns will be the second argument.

A *JTextField* object can be added to a *JPanel* like a *JLabel* or *JButton*.

13. GUI Programming :: The *JTextField* Class :: *TextFieldDemo1* Example

Let's write an application that creates a *JFrame* containing a *JPanel* laid out using *FlowLayout* (centered, 50 pixel vertical gap above). We will add three *JLabel*'s to a *JPanel* laid out using *GridLayout* (3 rows, 1 column, 10 pixel vertical gap between components) and three *JTextField*s to another *JPanel* also laid out using *GridLayout*. Our finished GUI will look like this:



13. GUI Programming :: The *JTextField* Class :: *TextFieldDemo1* (continued)

```
//*************************************************************************
// CLASS: TextFieldDemo1 (TextFieldDemo1.java)
//*************************************************************************
import java.awt.FlowLayout;
import java.awt.GridLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 * This application demonstrates how to create GUI text fields using the javax.
 * swing.JTextField class.
 */
public class TextFieldDemo1 {
  public static void main(String[] args) { new TextFieldDemo1().run(); }
  public void run() {
    // Use the Swing look and feel.
    JFrame.setDefaultLookAndFeelDecorated(true);

    // Create three JLabels.
    JLabel label1 = new JLabel("Text Field 1: ", JLabel.LEFT);
    JLabel label2 = new JLabel("Text Field 2: ", JLabel.LEFT);
    JLabel label3 = new JLabel("Text Field 3: ", JLabel.LEFT);
```

13. GUI Programming :: The *JTextField* Class :: *TextFieldDemo1* (continued)

```
// Create a JPanel for the JLabels and use the GridLayout with 3 rows and
// 1 column. Add the labels to the panel.
JPanel panelLabel = new JPanel();
panelLabel.setLayout(new GridLayout(3, 1, 0, 10));
panelLabel.add(label1);
panelLabel.add(label2);
panelLabel.add(label3);

// Create three JTextFields.
JTextField tf1 = new JTextField("");
JTextField tf2 = new JTextField("This is the initial text");
JTextField tf3 = new JTextField("(???) ???-????");

// Create a JPanel for the JTextFields and use the GridLayout with 3 rows
// and 1 column. Add the text fields to the panel.
JPanel panelTextField = new JPanel();
panelTextField.setLayout(new GridLayout(3, 1, 0, 10));
panelTextField.add(tf1);
panelTextField.add(tf2);
panelTextField.add(tf3);

// Create a main panel to hold the panelLabel and panelTextField panels.
// The layout for mainPanel will be FlowLayout. Add the panelLabel
// and panelTextField panels to mainPanel.
JPanel mainPanel = new JPanel();
mainPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 0, 50));
mainPanel.add(panelLabel);
mainPanel.add(panelTextField);
```

13. GUI Programming :: The *JTextField* Class :: *TextFieldDemo1* (continued)

```
        JFrame frame = new JFrame("TextField Demo 1");
        frame.setSize(450, 200);
        frame.add(mainPanel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```