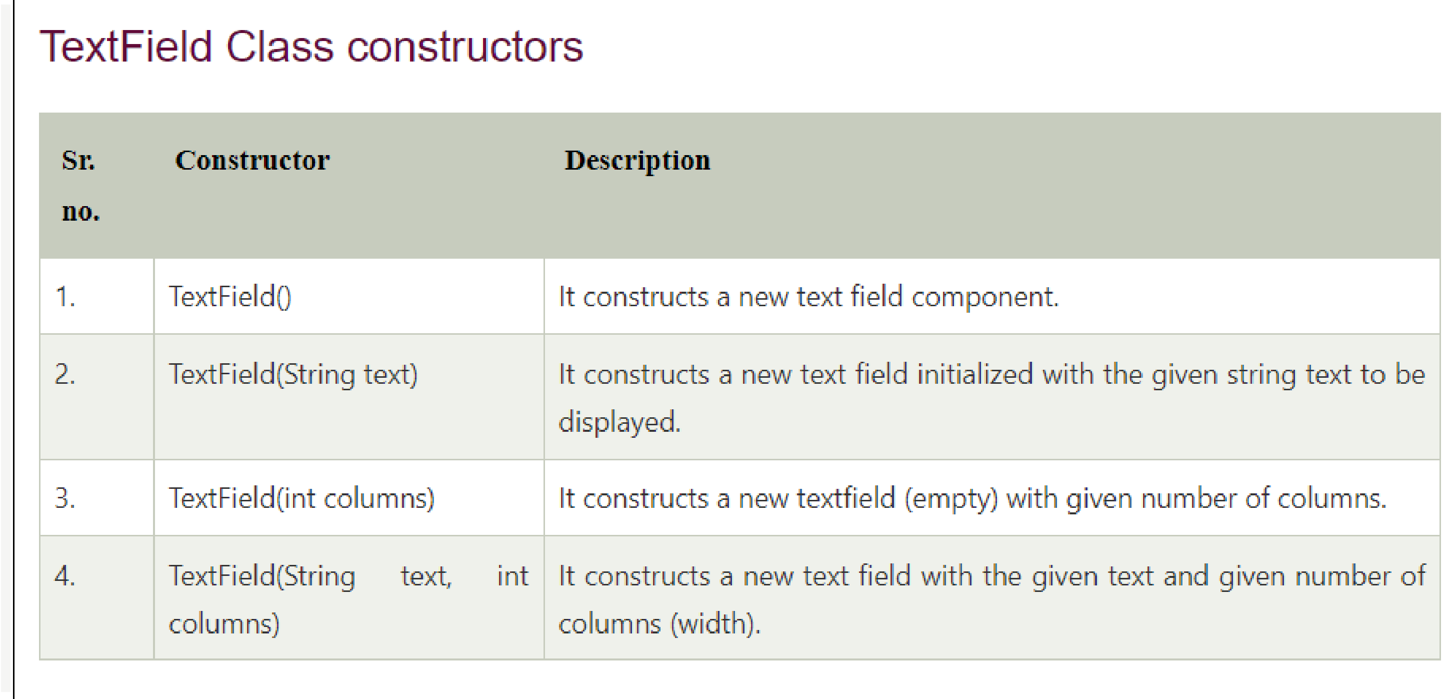
JAVA AWT TextField

The object of a **TextField** class is a text component that allows a user to enter a single line text and edit it. It inherits **TextComponent** class, which further inherits **Component** class.

When we enter a key in the text field (like key pressed, key released or key typed), the event is sent to **TextField**. Then the **KeyEvent** is passed to the registered **KeyListener.** It can also be done using **ActionEvent**; if the **ActionEvent** is enabled on the text field, then the **ActionEvent** may be fired by pressing return key. The event is handled by the **ActionListener** interface.

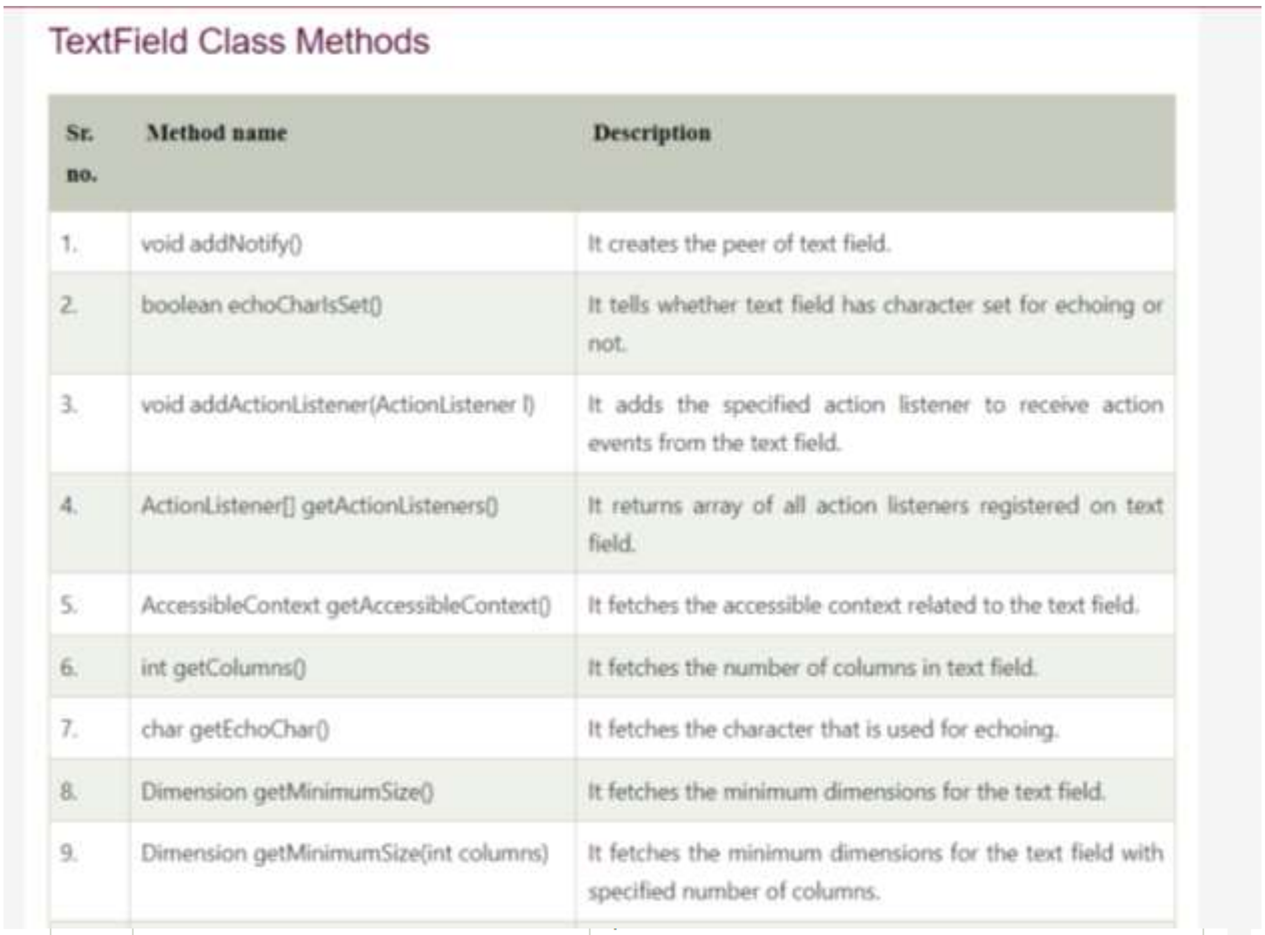
AWT TextField Class Declaration

public class TextField **extends** TextComponent



TextField Class constructors

Sr. no.	Constructor	Description
1.	TextField()	It constructs a new text field component.
2.	TextField(String text)	It constructs a new text field initialized with the given string text to be displayed.
3.	TextField(int columns)	It constructs a new textfield (empty) with given number of columns.
4.	TextField(String text, int columns)	It constructs a new text field with the given text and given number of columns (width).



TextField Class Methods

Sr. no.	Method name	Description
1.	void addNotify()	It creates the peer of text field.
2.	boolean echoCharIsSet()	It tells whether text field has character set for echoing or not.
3.	void addActionListener(ActionListener l)	It adds the specified action listener to receive action events from the text field.
4.	ActionListener[] getActionListeners()	It returns array of all action listeners registered on text field.
5.	AccessibleContext getAccessibleContext()	It fetches the accessible context related to the text field.
6.	int getColumns()	It fetches the number of columns in text field.
7.	char getEchoChar()	It fetches the character that is used for echoing.
8.	Dimension getMinimumSize()	It fetches the minimum dimensions for the text field.
9.	Dimension getMinimumSize(int columns)	It fetches the minimum dimensions for the text field with specified number of columns.
10.	Dimension getPreferredSize()	It fetches the preferred size of the text field.
11.	Dimension getPreferredSize(int columns)	It fetches the preferred size of the text field with specified number of columns.
12.	protected String paramString()	It returns a string representing state of the text field.
13.	protected void processActionEvent(ActionEvent e)	It processes action events occurring in the text field by dispatching them to a registered ActionListener object.
14.	protected void processEvent(AWTEvent e)	It processes the event on text field.
15.	void removeActionListener(ActionListener l)	It removes specified action listener so that it doesn't receive action events anymore.
16.	void setColumns(int columns)	It sets the number of columns in text field.
17.	void setEchoChar(char c)	It sets the echo character for text field.
18.	void setText(String t)	It sets the text presented by this text component to the specified text.

Method Inherited

The AWT TextField class inherits the methods from below classes:

1. java.awt.TextComponent
2. java.awt.Component
3. java.lang.Object

Java AWT TextField Example

TextFieldExample1.java

```
// importing AWT class
import java.awt.*;
public class TextFieldExample1 {
   // main method
   public static void main(String args[]) {
   // creating a frame
   Frame f = new Frame("TextField Example");
   TextField t1, t2;
   t1 = new TextField("Welcome to Javatpoint.");
   t1.setBounds(50, 100, 200, 30);
   t2 = new TextField("AWT Tutorial");
   t2.setBounds(50, 150, 200, 30);
   f.add(t1);
   f.add(t2);
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
}
}
```

Java AWT TextField Example with ActionListener

TextFieldExample2.java

```
// importing necessary libraries
import java.awt.*;
import java.awt.event.*;
// Our class extends Frame class and implements ActionListener interface
public class TextFieldExample2 extends Frame implements ActionListener {
   // creating instances of TextField and Button class
   TextField tf1, tf2, tf3;
   Button b1, b2;
   // instantiating using constructor
   TextFieldExample2() {
      // instantiating objects of text field and button
      // setting position of components in frame
      tf1 = new TextField();
      tf1.setBounds(50, 50, 150, 20);
      tf2 = new TextField();
      tf2.setBounds(50, 100, 150, 20);
      tf3 = new TextField();
      tf3.setBounds(50, 150, 150, 20);
      tf3.setEditable(false);
      b1 = new Button("+");
      b1.setBounds(50, 200, 50, 50);
      b2 = new Button("-");
      b2.setBounds(120,200,50,50);
```

```
    // adding action listener
    b1.addActionListener(this);
    b2.addActionListener(this);
    // adding components to frame
    add(tf1);
    add(tf2);
    add(tf3);
    add(b1);
    add(b2);
    // setting size, layout and visibility of frame
    setSize(300,300);
    setLayout(null);
    setVisible(true);
  }
  // defining the actionPerformed method to generate an event on buttons
  public void actionPerformed(ActionEvent e) {
    String s1 = tf1.getText();
    String s2 = tf2.getText();
    int a = Integer.parseInt(s1);
    int b = Integer.parseInt(s2);
    int c = 0;
    if (e.getSource() == b1){
      c = a + b;
    }
    else if (e.getSource() == b2){
      c = a - b;
```

```
        }
        String result = String.valueOf(c);
        tf3.setText(result);
    }
// main method
public static void main(String[] args) {
    new TextFieldExample2();
}
}
```