

JTextArea Tab Tuşu Kullanımı

JTextArea'da Tab tuşuna basıldığında seçili satırların 4 birim ileri yönde hareket ettirilmesini ve Shift + Tab tuşuna basıldığında tersinil işlemin yapılmasını sağlamak için aşağıdaki kodları kullanabiliriz:

```
1  import javax.swing.*;
2  import javax.swing.text.AttributeSet;
3  import javax.swing.text.BadLocationException;
4  import javax.swing.text.DefaultStyledDocument;
5  import java.awt.event.KeyAdapter;
6  import java.awt.event.KeyEvent;
7
8
9  public class Test extends JFrame {
10     private JTextArea txtContent = new JTextArea(30, 150);
11
12     public Test() {
13         add(txtContent);
14         txtContent.setDocument(new TabDocument());
15         addKeyListener();
16         pack();
17         setLocationRelativeTo(null);
18         setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
19         setVisible(true);
20     }
21
22     private static class TabDocument extends DefaultStyledDocument {
23         @Override
24         public void insertString(int offs, String str, AttributeSet a) throws BadLocationException {
25             str = str.replaceAll("\t", "");
26             super.insertString(offs, str, a);
27         }
28     }
29     private void addKeyListener() {
30         txtContent.addKeyListener(new KeyAdapter() {
31             private boolean isSelectedText = false;
32             private int selectionStart = 0;
33
34             @Override
35             public void keyPressed(KeyEvent e) {
36                 if (e.getKeyCode() == KeyEvent.VK_TAB && e.isShiftDown()) {
37                     selectionStart = txtContent.getSelectionStart();
38                     new TabBackSpace(txtContent).actionPerformed(null);
39                     isSelectedText = true;
40                 } else if (e.getKeyCode() == KeyEvent.VK_TAB) {
41
42                     if (txtContent.getSelectedText() != null) {
43                         selectionStart = txtContent.getSelectionStart();
44                         new TabSpace(txtContent).actionPerformed(null);
45                         isSelectedText = true;
46
47                     } else {
48                         txtContent.insert("    ", txtContent.getCaretPosition());
49                     }
50                 }
51             }
52
53             public void keyReleased(KeyEvent e) {
54                 if (e.getKeyCode() == KeyEvent.VK_TAB && isSelectedText) {
55                     int caretPosition = txtContent.getCaretPosition();
56                     txtContent.setSelectionStart(selectionStart);
57                     txtContent.setSelectionEnd(caretPosition);
58                 }
59             }
60         });
61     }
62     public static void main(String[] args) {
63         new Test();
64     }
65 }
```

Not: 36. satırdaki e.isShiftDown() metodu, Shift tuşuna basıldığı zaman true değerini alır.

TabSpace sınıfı

```
import javax.swing.*;
import java.awt.event.ActionEvent;

/**
 * TabSpace
 */
public class TabSpace extends AbstractAction {
    private JTextArea txtContent;

    public TabSpace(JTextArea txtContent) {
        this.txtContent = txtContent;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String selectedText = txtContent.getSelectedText();
        int selectionStart = txtContent.getSelectionStart();
        int selectionEnd = txtContent.getSelectionEnd();
        String[] lines = selectedText.split("\n");
        String result = "";
        for (String line : lines) {
            result += "    " + line + "\n";
        }

        result=result.replaceAll("\n$", "");
        txtContent.replaceRange(result, selectionStart, selectionEnd);
    }
}
```

TabBack Sınıfı

```
import javax.swing.*;
import java.awt.event.ActionEvent;

/**
 * TabBackSpace
 */
public class TabBackSpace extends AbstractAction {
    private JTextArea txtContent;

    public TabBackSpace(JTextArea txtContent) {
        this.txtContent = txtContent;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String selectedText = txtContent.getSelectedText();
        int selectionStart = txtContent.getSelectionStart();
        int selectionEnd = txtContent.getSelectionEnd();
        String[] lines = selectedText.split("\n");
        String result = "";
        for (String line : lines) {
            result += line.replaceAll("^(\s{4})", "") + "\n";
        }
        txtContent.replaceRange(result.replaceAll("(\\n)$", ""), selectionStart, selectionEnd);
    }
}
```