

让界面更加绚丽JavaSE6.0四种新功能 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/145/2021\\_2022\\_\\_E8\\_AE\\_A9\\_E7\\_95\\_8C\\_E9\\_9D\\_A2\\_E6\\_c104\\_145033.htm](https://www.100test.com/kao_ti2020/145/2021_2022__E8_AE_A9_E7_95_8C_E9_9D_A2_E6_c104_145033.htm)

在这篇文章中我接着介绍另外几种新的GUI功能。这些功能是：带有排序和过滤功能的JTable。增强的JTabbedPane组件 增强的打印功能 增强的拖放功能 带有排序和过滤功能的JTable 在Java SE 6中除了java.awt被更新外，javax.swing同时也有了很大的改进。

在C/S程序中我们会经常使用到"表"。如我们可以在查询数据库后将查询结果显示在表格中。在Java中显示表格使用的是JTable类。在以前的版本中，JTable只能简单地显示数据，并没有什么附加的处理功能，而在Java SE 6中的JTable增加了排序和过滤功能。用户可以单击列头进行排序，也可以根据某一行来过滤表中的数据。为了使JTable可以对数据进行，必须将RowSorter类和JTable进行关联。RowSorter是一个抽象类，它负责将JTable中的数据映射成可排序的数据。在真正使用时，我们将直接使用RowSorter的子类TableRowSorter。下面的代码显示了如何将TableRowSorter类和JTable相关联

```
TableModel model = new DefaultTableModel(rows, columns);
JTable table = new JTable(model);
RowSorter sorter = new TableRowSorter(model);
table.setRowSorter(sorter);
```

上面代码首先建立一个TableModel，然后将这个TableModel的实例同时传递给了JTable和RowSorter。下面是一个使用JTable排序的简单的例子。

```
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;

public class TestSortedTable {
    public static void main(String args[]) {
        JFrame frame = new JFrame("JTable的排序测
```

```
试"). frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE).  
// 表格中显示的数据 Object rows[][] = { { "王明", "中国", 44 }, { "  
姚明", "中国", 25 }, { "赵子龙", "西蜀", 1234 }, { "曹操", "北魏",  
2112 }, { "Bill Gates", "美国", 45 }, { "Mike", "英国", 33 } }. String  
columns[] = { "姓名", "国籍", "年龄" }. TableModel model = new  
DefaultTableModel(rows, columns). JTable table = new  
JTable(model). RowSorter sorter = new TableRowSorter(model).  
table.setRowSorter(sorter). JScrollPane pane = new  
JScrollPane(table). frame.add(pane, BorderLayout.CENTER).  
frame.setSize(300, 150). frame.setVisible(true). } }
```

图1和图2分别是按"姓名"进行升序和降序排列的显示结果。图1按"姓名"升序显示图2按"姓名"降序显示图3显示的是按"年龄"进行降序排列。但我们发现一个奇怪的问题，就是"年龄"字段并不是按数值类型进行排序的，而是按字符类型进行排序的。图3按年龄降序显示出现这种情况是因为在默认情况下DefaultTableModal的列是Object类型。而要想使JTable按数值进行排序，必须要覆盖DefaultTableModal的getColumnClass方法。TableModel model = new DefaultTableModel(rows, columns){ public Class getColumnClass(int column) { Class returnValue. if ((column >= 0) amp. (column { returnValue = getValueAt(0, column).getClass(). } else { returnValue = Object.class. } return returnValue. } }。图4显示了按"年龄"进行排序的界面，看看，是不是按数值进行排序了。图4按数值类型进行排序下面让我们来看看来何使用JTable进行过滤。我们可以通过convertRowIndexToModel方法进行过滤。下面的代码加在一个按钮中添加事件代码调用JTable的过滤功能

。 `button.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e) { String text = filterText.getText(). if (text.length() == 0) { sorter.setRowFilter(null). } else { sorter.setRowFilter(RowFilter.regexFilter(text)). } })`. 上面的代码并没有调用 `convertRowIndexToModel()` 方法，如果调用它，你就可以在表中进行相应的操作。在 `JTable` 中通过抽象类 `RowFilter` 类对行进行过滤。和排序不同，你可以不建立它们的子类，而使用这个抽象类的6个静态方法。 `andFilter` `dateFilter(RowFilter.ComparisonType type, Date date, int... indices)` `notFilter(RowFilter filter)` `numberFilter(RowFilter.ComparisonType type, Number number, int... indices)` `orFilter` `regexFilter(String regex, int... indices)` 其中 `andFilter()`、`orFilter()` 以及 `notFilter()` 方法的功能是将当前的过滤条件和其它的过滤条件进行组合。如在同时比较日期和数值时需要将日期过滤和数值过滤进行组合。这些组合是非常简单的。`RowFilter` 的类型比较允许你进行4种关系的比较，等于、不等于、大于或小于。我们可以通过指定某一列进行过滤，也可以对所有的列进行过滤。这其中最为有趣的也许是正则表达式过滤(regular expression filter，或简称为 `regex filter`)。使用这个过滤器可以对表中数据进行更高级的过滤。下面是实现一个简单过滤器的代码。

```
import javax.swing.*;import javax.swing.table.*;import java.awt.*;import java.awt.event.*;public class TestFilter{ public static void main(String args[]) { JFrame frame = new JFrame("JTable的过滤测试"). frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE). Object rows[][] = { { "王明", "中国", 44 }, { "姚明", "中国", 25 }, { "赵子龙", "西蜀", 1234 }, { "曹操", "北魏", 2112 }, { "Bill Gates", "美
```

```
国", 45 }, { "Mike", "英国", 33 } }. String columns[] = { "姓名", "国籍", "年龄" }. TableModel model = new DefaultTableModel(rows, columns) { public Class getColumnClass(int column) { Class returnValue. if ((column >= 0) amp. (column { returnValue = getValueAt(0, column).getClass(). } else { returnValue = Object.class. } return returnValue. } }. final JTable table = new JTable(model). final TableRowSorter sorter = new TableRowSorter(model). table.setRowSorter(sorter). JScrollPane pane = new JScrollPane(table). frame.add(pane, BorderLayout.CENTER). JPanel panel = new JPanel(new BorderLayout()). JLabel label = new JLabel("过滤"). panel.add(label, BorderLayout.WEST). final JTextField filterText = new JTextField(""). panel.add(filterText, BorderLayout.CENTER). frame.add(panel, BorderLayout.NORTH). JButton button = new JButton("过滤"). button.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent e) { String text = filterText.getText(). if (text.length() == 0) { sorter.setRowFilter(null). } else { sorter.setRowFilter(RowFilter.regexFilter(text)). } } }). frame.add(button, BorderLayout.SOUTH). frame.setSize(300, 250). frame.setVisible(true). }
```

图5是上面程序的运行界面。图5增强的JTabbedPane组件 JTabbedPane组件为我们提供了一种非常好的方法在窗体上显示很多的控件。我们可以将不同类别的控件放到不同的Tab页上，然后通过需要点击相应的Tab页。在传统的Tab页上只能防止文本的图标。而在Java SE 6中使我们可以直接将控件放到Tab上。我们可以通

过setTabComponentAt方法将控件放到Tab上。这个方法有两个参数，一个是Tab的索引，另一个是要放置的对象

。 JTabbedPane pane = new

JTabbedPane().pane.setTabComponentAt(1, component).

在JTabbedPane控件中有3个常用的方法

， setTabComponentAt(int index, Component comp),

getTabComponentAt(int index)

和indexOfTabComponent(Component)。最后一个方法将替

换Tab上的控件。下面的代码是一个关于JTabbedPane控件的

演示。 import javax.swing.\*.import javax.swing.table.\*.import

java.awt.\*.import java.awt.event.\*.public class TestTabbedPane{

static void addIt(JTabbedPane tabbedPane, String text) { JLabel label

= new JLabel(text). JButton button = new JButton(text). JPanel

panel = new JPanel(). panel.add(label). panel.add(button).

tabbedPane.addTab(text, panel). if(text.equals("tab4"))

tabbedPane.setTabComponentAt(tabbedPane.getTabCount() -

1,new JTextField("插入了文本控件")). else

tabbedPane.setTabComponentAt(tabbedPane.getTabCount() -

1,button). } public static void main(String args[]) { JFrame f = new

JFrame("JTabbedPane演示").

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE).

JTabbedPane tabbedPane = new JTabbedPane(). addIt(tabbedPane,

"tab1"). addIt(tabbedPane, "tab2"). addIt(tabbedPane, "tab3").

addIt(tabbedPane, "tab4"). addIt(tabbedPane, "tab5").

f.add(tabbedPane, BorderLayout.CENTER). f.setSize(400, 200).

f.setVisible(true). }} 100Test 下载频道开通，各类考试题目直接

下载。详细请访问 [www.100test.com](http://www.100test.com)