

Java程序“字数统计器”设计报告

马玉坤-1150310618

2016年7月23日

目录

1	题目描述	2
2	总体设计思想	2
3	详细设计	2
3.1	底层容器设计	2
3.2	中间容器设计	3
3.3	事件监听	4
3.4	单词数目统计方法	4
4	具体实现	5
5	运行结果	7

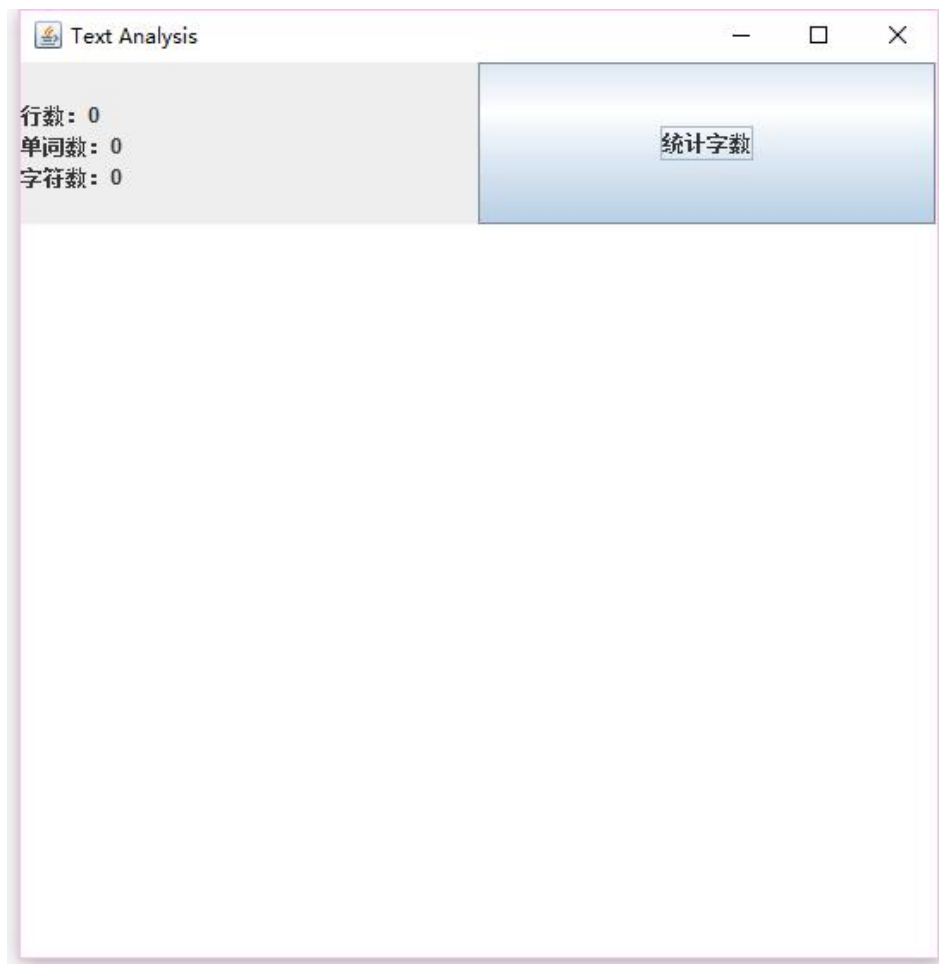
1 题目描述

编写应用程序，统计一个文本域输入文本的行数、单词数和字符数。可在图形界面中安排一个按钮、一个文本域和一个标签,点击按钮开始统计,在标签中显示结果。

2 总体设计思想

向窗体添加一个标签（用来显示统计信息），一个按钮（用来执行统计），一个文本域（用来供用户输入文本）。

标签和按钮在窗口上方，文本域在下方，用户输入一段文字后，点击“统计字数”按钮，用来显示统计信息的标签立刻显示出统计结果，分别对行数、单词数、字符数进行统计。



3 详细设计

3.1 底层容器设计

对于底层frame，将其布局管理器设为GridBagLayout，分为上下两行。上一行放一中间

容器(lbl_btn_back), 高度固定为40px, 宽度与窗口宽度相同。下一行放一文本域(text)占满窗口剩余空间。

```
1 // 将frame的布局管理器设为GridBagLayout, 两行一列
2 Container pane = frame.getContentPane();
3 pane.setLayout(new GridBagLayout());
4
5 GridBagConstraints c = new GridBagConstraints();
6 // 对于上下两个格子, x方向的空白区域全部填充
7 c.weightx = 1.0;
8 // 对于每个格子内部, 组件把格子空间全部占满
9 c.fill = GridBagConstraints.BOTH;
10 c.ipady = 40;
11
12 // 设置中间容器lbl_btn_back, 其上放置label和button
13 Panel lbl_btn_back = new Panel();
14 lbl_btn_back.setLayout(new GridLayout(1, 2));
15 c.gridx = 0;
16 c.gridy = 0;
17 c.gridwidth = 1;
18 c.gridheight = 1;
19 pane.add(lbl_btn_back, c);
20
21 // 设置供用户输入的文本域
22 JTextArea text = new JTextArea();
23 c.gridx = 0;
24 c.gridy = 1;
25 c.weightx = 1.0;
26 c.weighty = 1.0;
27 c.gridwidth = 2;
28 pane.add(text, c);
```

3.2 中间容器设计

在中间容器(lbl_btn_back)上, 使用布局管理器GridLayout分为一行两列。两个格子大小相同, 分别放置标签(label)和按钮(button)。

```
1 // 设置标签的初始内容, 并加入lbl_btn_back上
2 JLabel label
3 = new JLabel("<html>行数: 0<br>单词数: 0<br>字符数: 0</html>");
4 lbl_btn_back.add(label);
5
6 // 设置按钮的初始内容, 并加入lbl_btn_back上
7 JButton button = new JButton("统计字数");
```

```
8 lbl_btn_back.add(button);
```

3.3 事件监听

向button添加事件监听器监听actionPerformed事件。当按钮被点击时，更新标签内容。其中行数的统计使用text.getLineCount()，字符数的统计使用text.getText().length()，单词数的统计使用自定义方法int getWordCount(String s)。

```
1 // 绑定按钮事件，当用户点击按钮时更新label内容
2 button.addActionListener(
3     new ActionListener() {
4         public void actionPerformed(ActionEvent e) {
5             label.setText("<html>行数: " + text.getLineCount()
6                 + "<br>单词数: " + getWordCount(text.getText())
7                 + "<br>字符数: " + text.getText().length() + "</html>");
8         }
9     });
```

3.4 单词数目统计方法

在自定义的单词数目统计方法中，传入字符串s，返回s中被非英文字母隔开的连续英文字母的块数，也即满足 $s[i]$ 为字母 $\wedge (i = 0 \vee s[i - 1]$ 不为字母)的 i 的个数。例如字符串 $s = \text{"abc你def好ghi"}$ ，单词统计数目为3。

```
1 public static int getWordCount(String s) {
2     int ret = 0;
3     for (int i = 0; i < s.length(); i++) {
4         char ch = s.charAt(i);
5         if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
6             if (i == 0) {
7                 ret++;
8             } else {
9                 char prev_ch = s.charAt(i - 1);
10                if ((prev_ch < 'a' || prev_ch > 'z')
11                    && (prev_ch < 'A' || prev_ch > 'Z'))
12                    ret++;
13            }
14        }
15    }
16    return ret;
17 }
```

4 具体实现

```
1
2  import javax.swing.*;
3  import java.awt.*;
4  import javax.swing.JButton;
5  import javax.swing.JFrame;
6  import java.awt.event.*;
7
8  public static int getWordCount(String s) {
9      int ret = 0;
10     for (int i = 0; i < s.length(); i++) {
11         char ch = s.charAt(i);
12         if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
13             if (i == 0) {
14                 ret++;
15             } else {
16                 char prev_ch = s.charAt(i - 1);
17                 if ((prev_ch < 'a' || prev_ch > 'z')
18                     && (prev_ch < 'A' || prev_ch > 'Z'))
19                     ret++;
20             }
21         }
22     }
23     return ret;
24 }
25
26 public static void main(String[] args) {
27     // 设置frame标题, 默认大小
28     JFrame frame = new JFrame("Text Analysis");
29     frame.setSize(300, 400);
30     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31
32     // 将frame的布局管理器设为GridBagLayout, 两行一列
33     Container pane = frame.getContentPane();
34     pane.setLayout(new GridBagLayout());
35
36     GridBagConstraints c = new GridBagConstraints();
37     // 对于上下两个格子, x方向的空白区域全部填充
38     c.weightx = 1.0;
39     // 对于每个格子内部, 组件把格子空间全部占满
40     c.fill = GridBagConstraints.BOTH;
41     c.ipady = 40;
42 }
```

```

43      // 设置中间容器lbl_btn_back, 其上放置label和button
44      Panel lbl_btn_back = new Panel();
45      lbl_btn_back.setLayout(new GridLayout(1, 2));
46      c.gridx = 0;
47      c.gridy = 0;
48      c.gridwidth = 1;
49      c.gridheight = 1;
50      pane.add(lbl_btn_back, c);
51
52      // 设置供用户输入的文本域
53      JTextArea text = new JTextArea();
54      c.gridx = 0;
55      c.gridy = 1;
56      c.weightx = 1.0;
57      c.weighty = 1.0;
58      c.gridwidth = 2;
59      pane.add(text, c);
60
61      // 设置标签的初始内容, 并加入lbl_btn_back上
62      JLabel label
63      = new JLabel("<html>行数: 0<br>单词数: 0<br>字符数: 0</html>");
64      lbl_btn_back.add(label);
65
66      // 设置按钮的初始内容, 并加入lbl_btn_back上
67      JButton button = new JButton("统计字数");
68      lbl_btn_back.add(button);
69
70      // 绑定按钮事件, 当用户点击按钮时更新label内容
71      button.addActionListener(
72      new ActionListener() {
73          public void actionPerformed(ActionEvent e) {
74              label.setText("<html>行数: " + text.getLineCount() +
75                  "<br>单词数: " + getWordCount(text.getText()) +
76                  "<br>字符数: " + text.getText().length() + "</html>");
77          }
78      });
79
80      // 显示窗体
81      frame.setVisible(true);
82  }
83  }

```

5 运行结果

