

软件构造第七章作业

- 右图显示的是一个计算器软件的界面。请按照这个界面或读者所使用的操作系统中计算器软件的界面，使用Java Swing编写出界面布局并实现类似的软件功能。

代码块

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5
6 public class Calculator extends JFrame implements ActionListener {
7
8     private JTextField display = new JTextField("0");
9     private String operator = "";
10    private double firstNum = 0;
11    private boolean startNew = true;
12
13    public Calculator() {
14        setTitle("计算器");
15        setSize(300, 400);
16        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17        setLayout(new BorderLayout());
18
19        //显示屏
20        display.setFont(new Font("Arial", Font.BOLD, 32));
21        display.setHorizontalAlignment(SwingConstants.RIGHT);
22        display.setEditable(false);
23        add(display, BorderLayout.NORTH);
24
25        //按钮布局
26        String[] buttons = {
27            "CE", "C", "%", "/",
28            "7", "8", "9", "*",
29            "4", "5", "6", "-",
30            "1", "2", "3", "+",
31            "±", "0", ".", "="
32        };
33
34        JPanel panel = new JPanel(new GridLayout(5, 4, 5, 5));
35        for (String text : buttons) {
36            JButton btn = new JButton(text);
37            btn.setFont(new Font("Arial", Font.PLAIN, 20));
```

```
38         btn.addActionListener(this);
39         panel.add(btn);
40     }
41
42     add(panel, BorderLayout.CENTER);
43 }
44
45 @Override
46 public void actionPerformed(ActionEvent e) {
47     String cmd = e.getActionCommand();
48
49     //数字键
50     if ("0123456789".contains(cmd)) {
51         if (startNew || display.getText().equals("0")) {
52             display.setText(cmd);
53         } else {
54             display.setText(display.getText() + cmd);
55         }
56         startNew = false;
57         return;
58     }
59
60     //小数点
61     if (cmd.equals(".")) {
62         if (!display.getText().contains(".")) {
63             display.setText(display.getText() + ".");
64         }
65         return;
66     }
67
68     //CE清除当前输入
69     if (cmd.equals("CE")) {
70         display.setText("0");
71         startNew = true;
72         return;
73     }
74
75     //C全部清除
76     if (cmd.equals("C")) {
77         display.setText("0");
78         firstNum = 0;
79         operator = "";
80         startNew = true;
81         return;
82     }
83
84     //±取反
```

```
85     if (cmd.equals("±")) {
86         double value = Double.parseDouble(display.getText());
87         display.setText(String.valueOf(-value));
88         return;
89     }
90
91     //运算符
92     if ("+-*/%".contains(cmd)) {
93         firstNum = Double.parseDouble(display.getText());
94         operator = cmd;
95         startNew = true;
96         return;
97     }
98
99     //等号
100    if (cmd.equals("=)) {
101        double secondNum = Double.parseDouble(display.getText());
102        double result = 0;
103
104        switch (operator) {
105            case "+": result = firstNum + secondNum; break;
106            case "-": result = firstNum - secondNum; break;
107            case "*": result = firstNum * secondNum; break;
108            case "/":
109                if (secondNum != 0) result = firstNum / secondNum;
110                else { display.setText("错误"); return; }
111                break;
112            case "%": result = firstNum % secondNum; break;
113        }
114
115        display.setText(String.valueOf(result));
116        startNew = true;
117    }
118 }
119
120 public static void main(String[] args) {
121     SwingUtilities.invokeLater(() -> {
122         Calculator calc = new Calculator();
123         calc.setVisible(true);
124     });
125 }
126 }
127 }
```