package 计算器;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

/\*

\* 制作计算器

\* 1.要求有正常计算功能

\* 2.限制为不能出现 00 1.0. .3333

\*/

public class computor extends JFrame implements ActionListener {

JButton b0, b;

JPanel panel1, panel2;

TextField text1; // 制作控件

public static void main(String[] args) {

computor jframe = new computor(); // 初始化

jframe.setResizable(false); // 面板不可拉伸

jframe.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

jframe.setVisible(true); // 可见状态

}

public computor() {

setTitle("计算器");

setLocation(500, 500);

setSize(360, 440); // 定义计算器界面大小

Dimension dm = new Dimension(50, 50);

Dimension tx = new Dimension(210, 40); // 定义两种控件大小的格式

panel1 = new JPanel();

panel2 = new JPanel(); // 定义两个面板

text1 = new TextField();

text1.setFont(new Font("", Font.BOLD, 30));

text1.setPreferredSize(tx); // 定义计算器显示屏的文本框

b0 = new JButton("C");

b0.setFont(new Font("", Font.BOLD, 22));

b0.setPreferredSize(new Dimension(50, 40));

b0.addActionListener(this); // 清零键样式制作

b = new JButton("X");

b.setFont(new Font("", Font.BOLD, 24));

b.setForeground(Color.RED);

b.setPreferredSize(new Dimension(50, 40));

b.addActionListener(this); // 退位键样式制作

panel1.add(text1);

panel1.add(b0);

panel1.add(b); // 界面北面板添加控件

String S = "789+456-123\*0.=/";

panel2.setLayout(new GridLayout(4, 4)); // 南面板定义网格布局，便于安放按钮

for (int i = 0; i < 16; i++) {

String temp = S.substring(i, i + 1);

JButton bt = new JButton();

bt.setText(temp);

bt.setFont(new Font("", Font.BOLD, 30));

bt.setPreferredSize(dm);

if (i == 3 | i == 7 | i == 11 | i == 14 | i == 15)

bt.setForeground(Color.RED);

bt.addActionListener(this);

panel2.add(bt);

} // 循环定义16个按钮，设置格式，监听器，面板添加

add(panel1, BorderLayout.NORTH);

add(panel2, BorderLayout.CENTER);

// 面板分南北两个

}

String num1 = "";

String num2 = "";

String operator = ""; // 定义全局变量，存储操作数跟操作符

public void actionPerformed(ActionEvent e) { // 监听动作执行

String result = null;

String click = e.getActionCommand();

if ("C".indexOf(click) != -1)

text1.setText(""); // 清零功能实现

else if ("X".indexOf(click) != -1 && text1.getText().isEmpty() == false)

text1.setText(text1.getText().substring(0, text1.getText().length() - 1)); // 退格功能实现

else if (".0123456789".indexOf(click) != -1) {

if ((text1.getText().indexOf(".") != -1 && click.equals("."))

|| (text1.getText().isEmpty() == true && click.equals("."))

|| (text1.getText().isEmpty() == true && click.equals("X"))) // 空内容限制退格功能

// .功能限制实现

;

else if (text1.getText().length() == 1 && text1.getText().indexOf("0") != -1 && click.equals("0"))

;

// 0功能限制实现

else {

text1.setText(text1.getText() + click);

} // 正常点击功能实现

} else if ("+-\*/".indexOf(click) != -1) { // 运算符存储及操作数记录

num1 = text1.getText();

operator = click;

text1.setText("");

} else if ("=".indexOf(click) != -1) {

num2 = text1.getText();

Double a = Double.valueOf(num1);

Double b = Double.valueOf(num2);

Double R = null;

switch (operator) { // switch实现加减乘除选择

case "+":

R = a + b;

break;

case "-":

R = a - b;

break;

case "\*":

R = a \* b;

break;

case "/":

R = a / b;

break;

default:

break;

}

text1.setText(R.toString()); // 计算结果输出显示屏上面

}

}

}