# 单元10 源代码

# 1.CalculateGuiClass类

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

/\*\*

\*

\* @author cqWrb

\* 此类是计算器的主类，用于显示图形界面和对事件操作的类

\*/

public class CalculateGuiClass

{

//定义成员变量---图形界面的所有图元

private JFrame calculateFrame;

private JPanel mainPanel,displayPanel,keyPanel;

private String numberKey="1234567890";

private String operaterKey="+-\*/().<C=";//<---删除显示区中运算表达式中最后一个的字符，C----清空显示区中的运算表达式

private JTextField calTxt;

private JButton[] keyButtons;

private int btnNumbers=20;//按简单计算器中的钮数量

/\*\*

\* 构造器 重载

\*/

public CalculateGuiClass()

{

this("简单计算器",500,350);

}

public CalculateGuiClass(String title)

{

this(title,500,350);

}

public CalculateGuiClass(String title,int width,int height)

{

initialGUI(title,width,height);

}

/\*\*

\* 初始化方法-----将界面中的所有成员初始化，且实现其显示布局

\*

\*/

private void initialGUI(String title,int width,int height)

{

//定义窗体界面:创建窗体对象，设置窗体大小及样式背景等，设置窗体响应，设置窗体显示属性

calculateFrame=new JFrame(title);

calculateFrame.setSize(width,height);

calculateFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//此处调用绝对布局方法------如何写绝对布局方法？

absoluteLayout();

//创建计算器上的按钮数组及实例化数组中的所有按钮

keyButtons=new JButton[btnNumbers];

for(int i=0;i<btnNumbers/5;i++)//此循环是根据上面提供的数字、运算符字符串进行按钮字符串创建-----此代码不具有扩展性

{

if(i<3)//对前四行中的按钮创建

{

for(int j=0;j<3;j++)

{ char keyName=numberKey.charAt(i\*3+j);

keyButtons[i\*5+j]=new JButton(keyName+"");

}

for(int j=0;j<2;j++)

{

char keyName=operaterKey.charAt(i\*2+j);

keyButtons[i\*5+j+3]=new JButton(keyName+"");

}

}

else//对最后一行的按钮创建

{

char charKeyName=numberKey.charAt(numberKey.length()-1);

keyButtons[3\*5]=new JButton(charKeyName+"");

for(int j=1;j<5;j++)

{

charKeyName=operaterKey.charAt(3\*2-1+j);

keyButtons[3\*5+j]=new JButton(charKeyName+"");

}

}

}

//创建显示文本区对象，设置文本区对象的属性

calTxt=new JTextField();

calTxt.setEditable(false);

calTxt.setFont(new Font("隶书",1,30));

calTxt.setPreferredSize(new Dimension(displayPanel.getWidth()-3,60));

calTxt.setHorizontalAlignment(JTextField.RIGHT);

//将显示区中的组件添加到显示块

displayPanel.add(calTxt);

//将按钮组添加到按钮区

for(int i=0;i<keyButtons.length;i++ )

{

keyButtons[i].setBorder(BorderFactory.createRaisedBevelBorder());

keyButtons[i].setFont(new Font("隶书",1,20));

keyPanel.add(keyButtons[i]);

}

//将主显示区添加到窗体构件实现显示

calculateFrame.add(mainPanel);

//重新绘制窗口

calculateFrame.setVisible(true);

//在事件源上注册监听

ButtonActionListener btnActionObj=new ButtonActionListener();

for(JButton b:keyButtons)

b.addActionListener(btnActionObj);

}

/\*\*

\* 布局管理器设置-----窗体框架布局设计-----采用绝对布局管理器

\* 主显示区域设计为无布局管理器-----

\* 显示区域设置高度为20

\*/

private void absoluteLayout()

{

//对窗体中的组件进行布置设置----区域块

mainPanel=new JPanel();

keyPanel=new JPanel();

displayPanel=new JPanel();

mainPanel.setLayout(null);

//设置主显示区域与Jframe外的边界宽度，主要是便于后面的显示扣减

int boundsWidth=15;

int boundsHeight=30;

displayPanel.setBounds(0,0, calculateFrame.getWidth()-boundsWidth,50);//显示区域的绝对布局

keyPanel.setBounds(0,displayPanel.getHeight(),calculateFrame.getWidth()-boundsWidth,calculateFrame.getHeight()-displayPanel.getHeight()-boundsHeight);//键盘区域的绝对布局

keyPanel.setLayout(new GridLayout(4,5,3,2));

displayPanel.setLayout(new FlowLayout());

mainPanel.add(displayPanel);

mainPanel.add(keyPanel);

}

//定义 事件处理器类

class ButtonActionListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

JButton btn=(JButton)e.getSource();

char inputChar=btn.getText().charAt(0);

if(inputChar!='<' && inputChar!='=' && inputChar!='=')

calTxt.setText(calTxt.getText()+inputChar);

else if(inputChar=='=')

{

Double result=CalculateClass.calculate(calTxt.getText());

calTxt.setText(result.toString());

}

}

}

public static void main(String[] args)

{

new CalculateGuiClass();

}

}

## 2. CalculateClass类

import java.util.\*;

/\*\*

\*

\* @author cqWrb

\*/

public class CalculateClass {

// 主函数

final static int N = 100; //设置最长字串运算表达式的长度

/\*\*

\* 对运算符的优先比较方法

\*/

private static int com(String x) {

if (x.equals("\*") || x.equals("/")) {

return 2;

}

if (x.equals("\*") || x.equals("/")) {

return 1;

} else {

return -1;

}

}

/\*\*

\* 计算方法---参数传入的是字符串形式的计算式，返回的计算结果为双精度类型数据

\* 运算表达式中可以包含加减乘除和括弧运算符 如"(3.5+6)\*(45/38-12)"

\*/

public static double calculate(String Input) {

String example = new String("()+\*/-");

String[] input = new String[N];

String[] Post = new String[N];

int n = Input.length();

int k;

int count = 0;

int pre = -1;

int now;

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

if (example.indexOf(Input.charAt(i)) != -1) {

if (pre != -1 && i != pre) {

input[count++] = Input.substring(pre, i);

}

if (pre == -1) {

input[count++] = Input.substring(0, i);

}

input[count++] = Input.substring(i, i + 1);

pre = i + 1;

}

}

if (pre < n) {

if (pre == -1) {

input[count++] = Input.substring(0, n);

} else {

input[count++] = Input.substring(pre, n);

}

}

//----------------------------------------------

Stack<String> S = new Stack<String>();

int j = 0;

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

if (example.indexOf(input[i]) == -1) {

Post[j++] = input[i];

} else {

if (input[i].equals("(")) {

S.push(input[i]);

} else if (input[i].equals(")")) {

while (!S.peek().equals("(")) {

Post[j++] = S.pop();

}

S.pop();

} else {

if (S.empty()) {

S.push(input[i]);

} else {

while (!S.empty() && com(S.peek()) >= com(input[i]) && (!S.peek().equals("("))) {

Post[j++] = S.pop();

}

S.push(input[i]);

}

}

}

}

while (!S.empty()) {

Post[j++] = S.pop();

}

S.clear();

int m = j;

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

if (Post[i].equals("+")) {

double a = Double.parseDouble(S.pop());

S.push(String.valueOf(Double.parseDouble(S.pop()) + a));

} else if (Post[i].equals("-")) {

double a = Double.parseDouble(S.pop());

S.push(String.valueOf(Double.parseDouble(S.pop()) - a));

} else if (Post[i].equals("\*")) {

double a = Double.parseDouble(S.pop());

S.push(String.valueOf(Double.parseDouble(S.pop()) \* a));

} else if (Post[i].equals("/")) {

double a = Double.parseDouble(S.pop());

S.push(String.valueOf(Double.parseDouble(S.pop()) / a));

} else

{

S.push(Post[i]);

}

}

String temp=S.pop();

if (!S.isEmpty())

temp=S.pop();

double result = Double.parseDouble(temp);

return result;

}

public static void main(String[] args)

{

CalculateClass obj=new CalculateClass();

System.out.println("(3.5+6)\*(45/38-12)="+obj.calculate("(3.5+6)\*(45/38-12)"));

}

}