**面向对象程序设计实验**

**实验项目名称: 综合实验**

**学生姓名: 范少帅 学号： 20181101047 班级： 18级计科3班**

**提交时间: 2019年12月16日 指导教师: 司建辉**

格式说明: 正文 1.25倍行距 小四 宋体

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**一、实验目的：**

1、掌握Java图形用户界面的编程设计方法；

2、掌握异常处理的编程方法。

3、掌握组件的监听和动作响应处理方法

**二、实验内容**

实现一个计算器。

**三、实验原理**、

1、参照Windows计算器功能，实现加、减、乘、除运算；

2、声明为按钮数组；

3、对0~9数字操作，采用相同的算法；

4、解决除数为零的问题；

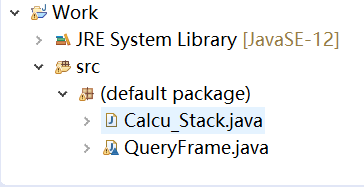
5、具有清除一个数字、全部清除、开平方、取百分号功能。

6、增加菜单栏的选项，使用户可选择不同的风格。

7、界面缩放时，所有组件同步缩放，保持界面整体风格不变。

**四、实验步骤**

1．在MyEclipse环境中，新建工作区和类；



2．编辑并编译源程序；

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.util.Scanner;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**import** javax.swing.\*;

**class** QueryFrame **extends** JFrame **implements** ActionListener

{

JPanel p = **new** JPanel(**new** GridLayout(5,4,5,4)); //定义面板，并设置为网格布局，5行4列，组件水平、垂直间距均为3

JTextField t = **new** JTextField(); //定义文本框

JCheckBox checkb\_bold;

JCheckBox checkb\_italic;

String str[]={"sqrt","%","CE","C","7","8","9","/","4","5","6","\*","1","2","3","-","0",".","=","+"}; //定义字符串数组，为按钮的显示文本赋值

JButton btn[]=**new** JButton[str.length]; //声明并创建按钮数组

**double** n1,n2,flag\_bold = 0 , flag\_italic = 0;

**private** JMenu[] menus;

**private** String colorstr[] = {"黑色","绿色","蓝色","红色"};

**private** Color[] colors = {Color.***black***,Color.***GREEN***,Color.***blue***,Color.***red***};

**private** Calcu\_Stack exp; //实例化Calcu\_Stack对象进行运算

**public** QueryFrame()

{

**super**("Calculator");

t.setSize(40,40);

t.setHorizontalAlignment(JTextField.***RIGHT***);//设置文字水平对齐方式为右对齐

t.setFont(**new** java.awt.Font("黑体",0, 25));

t.setEditable(**false**);//文本框不许编辑

setLayout(**new** BorderLayout()); //定义窗体布局为边界布局

**for**(**int** i=0;i<str.length;i++) //循环定义按钮，并添加到面板中

{

btn[i]=**new** JButton(str[i]);

p.add(btn[i]);

btn[i].addActionListener(**this**);

**if**(isNumber(btn[i].getActionCommand()))

{

btn[i].setFont(**new** java.awt.Font("华文琥珀", 1, 18));

}

**else**

{

btn[i].setFont(**new** java.awt.Font("行楷", 2, 15));

}

}

**this**.add(p);

**this**.getContentPane().add(t,BorderLayout.***NORTH***); //将文本框放置在窗体NORTH位置

**this**.getContentPane().add(p,BorderLayout.***CENTER***); //将面板放置在窗体CENTER位置

**this**.setSize(400,350);

**this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***); //实现关闭窗口功能

**this**.setLocationRelativeTo(**null**); //让窗体居中显示

**this**.addMenu();

**this**.setVisible(**true**);

}

**public** **void** actionPerformed(ActionEvent ev)

{

Font font = t.getFont(); //获得文本区的当前字体对象

**int** style = font.getStyle(); //获得字形

//如果输入加减乘除以及数字直接添加

**if**(isNumber(ev.getActionCommand()) || ev.getActionCommand() == "+" || ev.getActionCommand() == "-" || ev.getActionCommand() == "\*" || ev.getActionCommand() == "/")//如果按下数字键

{

**if**(isNumber(ev.getActionCommand()) && t.getText().equals("0"))

{

t.setText(ev.getActionCommand());

}

**else**

{

t.setText(t.getText()+ ev.getActionCommand());

}

}

**if**( ev.getActionCommand() == ".")//如果按下点

{

**if**(t.getText().indexOf(".") != -1)

{

Exception e1 = **new** Exception();

e1.printStackTrace();

JOptionPane.*showMessageDialog*(**null**, "操作错误!╮(╯▽╰)╭", "错误",JOptionPane.***WARNING\_MESSAGE***);

}

**else**

{

t.setText(t.getText()+ev.getActionCommand());

}

}

**if**( ev.getActionCommand() == "=")//如果按下=

{

exp = **new** Calcu\_Stack();

exp.setExp(t.getText());

t.setText(resultFormat(exp.calcu()));

}

**if**(ev.getActionCommand() == "%" )//如果按下+/-

{

**double** num = Double.*parseDouble*(t.getText()); //保存原来的值

**if**(num != 0)

{

**double** d = num/100;

t.setText(resultFormat(String.*valueOf*(d)));

}

}

**if**(ev.getActionCommand() == "sqrt" )//如果按下sqrt

{

**if**((t.getText().indexOf("+") != -1) || (t.getText().indexOf("\*") != -1) || (t.getText().indexOf("/") != -1)) // /后面不能输入0 否则弹出错误信息

{

Exception e1 = **new** Exception();

e1.printStackTrace();

JOptionPane.*showMessageDialog*(**null**, "开方系数含有特殊字符！╮(╯▽╰)╭", "错误",JOptionPane.***WARNING\_MESSAGE***);

}

**else** **if**((t.getText().indexOf("-") != -1))

{

JOptionPane.*showMessageDialog*(**null**, "被开方系数不能为负数！╮(╯▽╰)╭", "错误",JOptionPane.***WARNING\_MESSAGE***);

}

**else**

{

**double** num = Math.*sqrt*(Double.*valueOf*(t.getText()));

t.setText(resultFormat(String.*valueOf*(num)));

}

}

**if**(ev.getActionCommand() == "CE" )//如果按下CE

{

**if**(t.getText().length() == 1)

{

t.setText("0");

}

**else**

{

String s = t.getText().substring(0,t.getText().length()-1);

t.setText(s);

}

}

**if**(ev.getActionCommand() == "C" )//如果按下C

{

t.setText("0");

}

**if**(ev.getSource() **instanceof** JMenuItem) //如果选择菜单栏

{

**if** (ev.getActionCommand()=="退出")

{

**if** (JOptionPane.*showConfirmDialog*(**this**, "退出计算器？")==0)

{

System.*exit*(0); //单击确认对话框中的“是”按钮，结束程序运行

}

}

**if** (ev.getActionCommand()=="复制")

t.copy(); //将选中文本复制送系统剪贴板

**if** (ev.getActionCommand()=="黑色")

{

Color color = **new** Color(0,0,0);

t.setForeground(color); //设置文本区颜色

}

**if** (ev.getActionCommand()=="绿色")

{

Color color = **new** Color(0,255,0);

t.setForeground(color); //设置文本区颜色

}

**if** (ev.getActionCommand()=="蓝色")

{

Color color = **new** Color(0,0,255);

t.setForeground(color); //设置文本区颜色

}

**if** (ev.getActionCommand()=="红色")

{

Color color = **new** Color(255,0,0);

t.setForeground(color); //设置文本区颜色

}

}

**if** (ev.getSource()==checkb\_bold) //粗体

{

style = style ^ 1; //整数的位运算，异或^

t.setFont(**new** Font("宋体", style, 25)); //设置文本区字体

}

**if** (ev.getSource()==checkb\_italic) //斜体

{

style = style ^ 2; //整数的位运算，异或^

t.setFont(**new** Font("宋体", style, 25)); //设置文本区字体

}

**if**(ev.getActionCommand().equals("关于软件"))

{

JOptionPane.*showMessageDialog*(**null**, "实验6-31 计算器 V 1.2 \n ----18计科3班 范少帅", "关于软件",JOptionPane.***INFORMATION\_MESSAGE***);

}

}

**public** **boolean** isNumber(String str) //判断是不是数字

{

Pattern pattern = Pattern.*compile*("[0-9]\*");

Matcher isNum = pattern.matcher(str);

**if**(!isNum.matches())

{

**return** **false**;

}

**return** **true**;

}

**public** String resultFormat(String str) //约束数值格式

{

**double** d = Double.*valueOf*(str);

**if**((**int**)d != d)

{

**return** String.*format*("%.14f",Double.*valueOf*(str)+1e-15);

}

**else**

{

**return** String.*format*("%.0f",Double.*valueOf*(str));

}

}

**private** **void** addMenu()

{

JMenuBar menubar = **new** JMenuBar(); //菜单栏

**this**.setJMenuBar(menubar); //添加菜单栏

String menustr[] = {"菜单","编辑","格式","关于"}; //一级菜单

String menuitemstr[][] = {{"|","退出"},{"复制"},{},{"关于软件"}};//二级菜单

**this**.menus = **new** JMenu[menustr.length];//一级菜单数组

JMenuItem menuitems[][] = **new** JMenuItem[menuitemstr.length][];//二级菜单数组

**for**(**int** i=0;i<menuitemstr.length;i++)//添加名称

{

**this**.menus[i] = **new** JMenu(menustr[i]);

menubar.add(**this**.menus[i]);

menuitems[i] = **new** JMenuItem[menuitemstr[i].length];

**for**(**int** j=0;j<menuitemstr[i].length;j++)

{

**if**(menuitemstr[i][j].equals("|"))

{

**this**.menus[i].addSeparator();

}

**else**

{

menuitems[i][j] = **new** JMenuItem(menuitemstr[i][j]);

**this**.menus[i].add(menuitems[i][j]);

menuitems[i][j].addActionListener(**this**);

}

}

}

JMenu menu\_style = **new** JMenu("字形");

menus[2].add(menu\_style);

checkb\_bold = **new** JCheckBox("粗体"); //字形复选框

menu\_style.add(checkb\_bold);

checkb\_bold.addActionListener(**this**); //复选框注册单击事件监听器

checkb\_italic = **new** JCheckBox("斜体");

menu\_style.add(checkb\_italic);

checkb\_italic.addActionListener(**this**);

JMenu menu\_color = **new** JMenu("颜色");

menus[2].add(menu\_color);

ButtonGroup buttongroup = **new** ButtonGroup();

**for**(**int** i=0;i<**this**.colorstr.length;i++)

{

JRadioButtonMenuItem rbmi = **new** JRadioButtonMenuItem(**this**.colorstr[i]);

buttongroup.add(rbmi);

menu\_color.add(rbmi);

rbmi.setForeground(**this**.colors [i]);

rbmi.addActionListener(**this**);

}

}

**public** **static** **void** main(String[] args)//主函数

{

**new** QueryFrame();

}

}

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.Stack;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.swing.JOptionPane;

public class Calcu\_Stack

{

private String exp ;

public Calcu\_Stack()

{

setExp("0");

}

public void setExp(String str)

{

exp = str;

}

public String getExp()

{

return exp;

}

public String calcu()

{

double res = 0;

try

{

List<Object> list = trans(exp);

Stack<Double> result=new Stack<Double>();

res=f(list,result);

}

catch (Exception e)

{

JOptionPane.showMessageDialog(null, "表达式不合法！", "错误",JOptionPane.WARNING\_MESSAGE);

}

return String.valueOf(res);

}

private double f(List<Object> list, Stack<Double> result)

{

Iterator<Object> it=list.iterator();

while(it.hasNext()){

String m=it.next().toString();

if (m.equals("+")||m.equals("-")||m.equals("\*")||m.equals("/")) {

double b=result.pop();

double a=result.pop();

double v=g(a,b,m);

result.push(v);

}else {

result.push(Double.valueOf(m));

}

}

return(result.pop());

}

private double g(double a, double b, String m)

{

double v=0;

switch (m)

{

case "+":

v=a+b;

break;

case "-":

v=a-b;

break;

case "\*":

v=a\*b;

break;

case "/":

if(b==0)

{

JOptionPane.showMessageDialog(null, "除数不能为0！", "错误",JOptionPane.WARNING\_MESSAGE);

return 0;

}

else

{

v=a/b;

}

break;

}

return v;

}

private List<Object> trans(String s)

{

Stack<Character> op=new Stack<Character>();

ArrayList<Object> list=new ArrayList<Object>();

Pattern P=Pattern.compile("[0-9]+(\\.[0-9]+)?"); //正则表达式来处理带小数点的数字

int i=0;

while(i<s.length()){

char c=s.charAt(i);

if (c>='0'&&c<='9') {

String s1=s.substring(i);

Matcher m =P.matcher(s1);

if (m.find()) { //取匹配到的第一个数字

s1=m.group();

list.add(s1);

}

i=i+s1.length();

continue;

}else if (c=='(') {

op.push(c);

}else if (c==')') {

char p=op.pop();

while(p!='('){

list.add(p);

p=op.pop();

}

}else if (c=='+'||c=='-') {

while(!op.isEmpty()&&(op.peek()=='+'||op.peek()=='-'||

op.peek()=='\*'||op.peek()=='/')){

list.add(op.pop());

}

op.push(c);

}else if (c=='\*'||c=='/') {

while(!op.isEmpty()&&(op.peek()=='\*'||op.peek()=='/')){

list.add(op.pop());

}

op.push(c);

}

i++;

}

while(!op.isEmpty()){

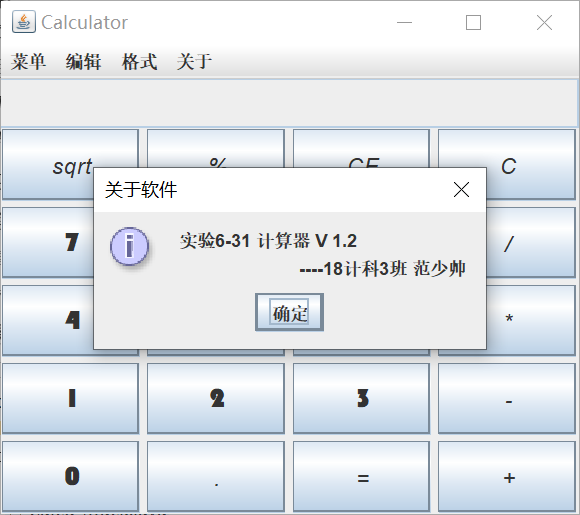
list.add(op.pop());

}

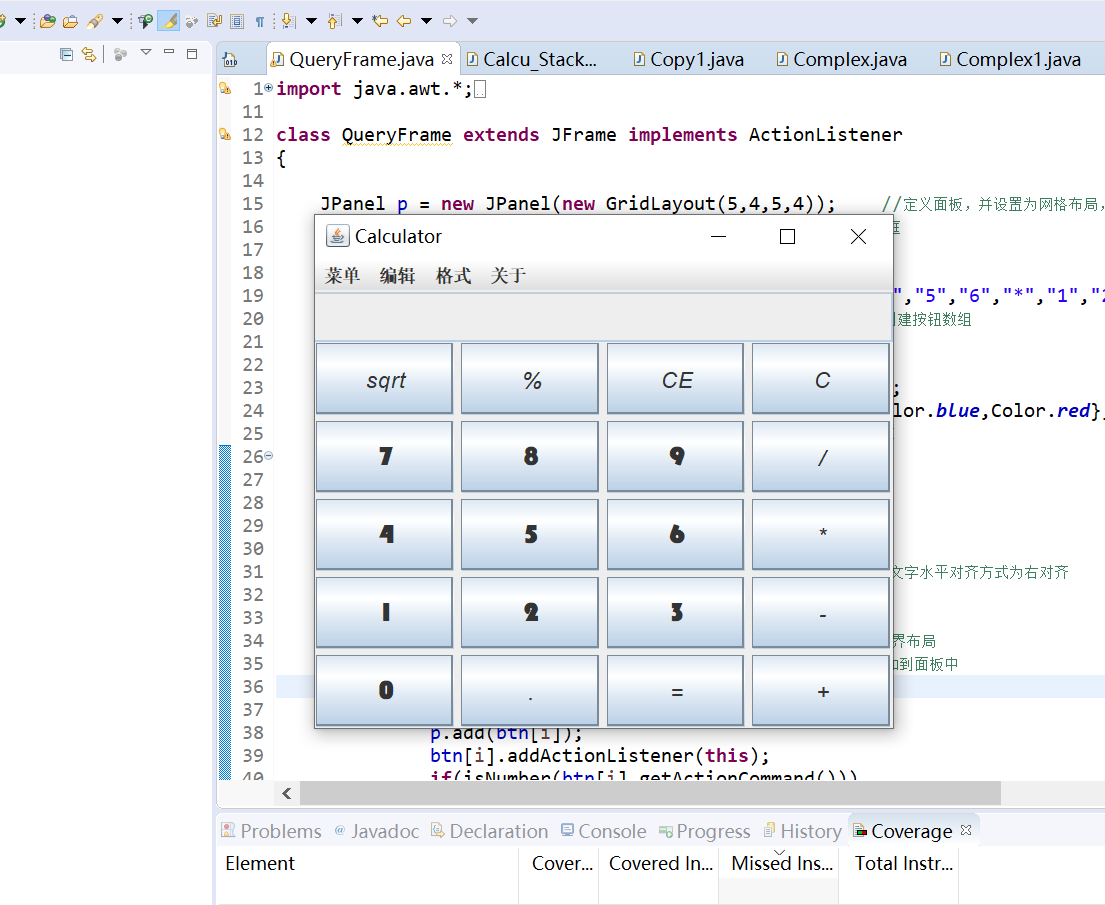
return list;

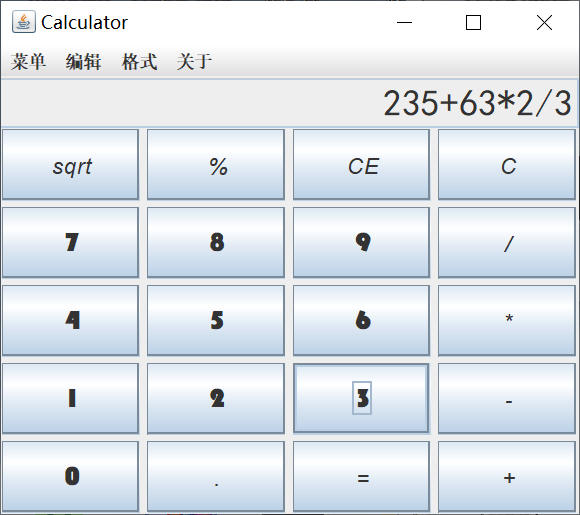
}

}

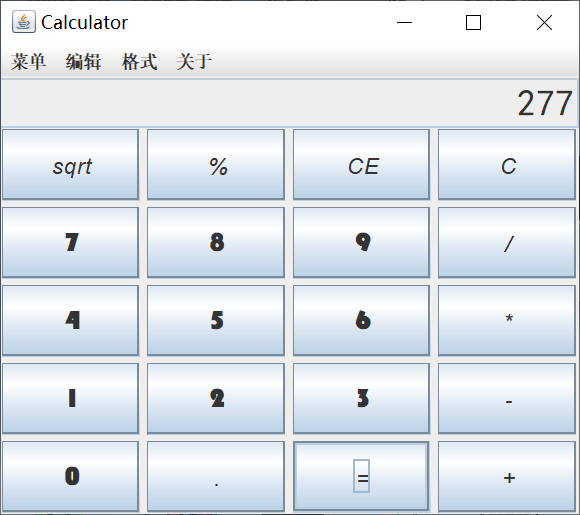


3、运行。

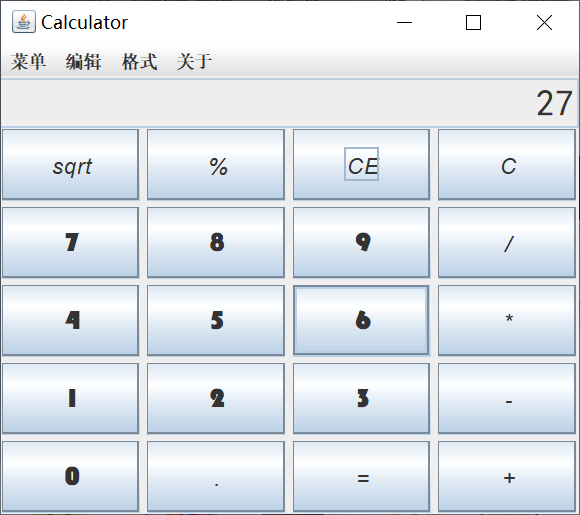




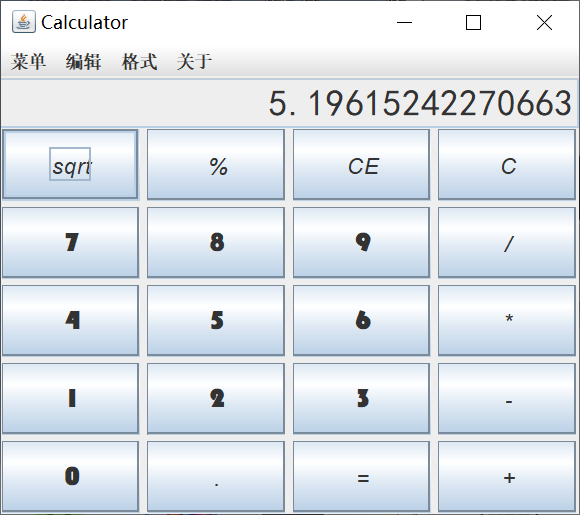
加减乘除



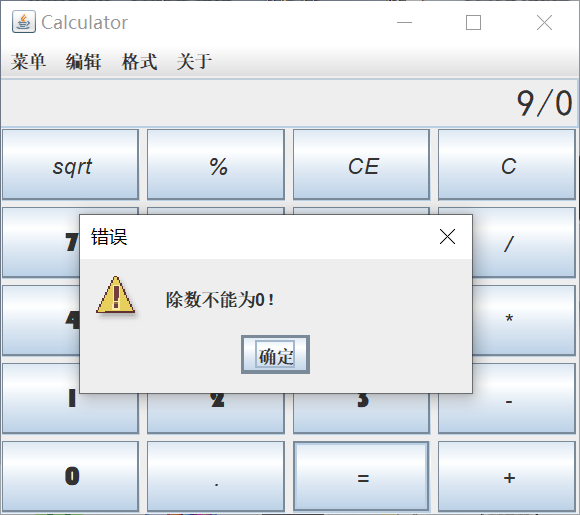
等于



清除一个



开方



报错

**五、实验心得体会**

（1）在创建选项框时，要将所有的选项放在一个“容器”里，并把这个容器添加到程序中。

（2）由于设置了按钮响应功能，所以要设置按键动作和响应，这里导入了包java.awt.event.ActionEvent和java.awt.event.ActionListener

（3）因为文本框中输入读取到的是字符串，所以要进行计算时，要先将其转为整形，在文本框输出时，同理要将整形转换为字符串

（4）注意：当光标移动到文本框上面时，获得焦点事件，当光标移走时，便失去焦点事件，所以要注意两个函数的作用。