**简易计算器设计文档**

1. **总体设计**

本应用程序继承自框架类(Frame), 此程序使用Frame布局管理器BorderLayout,将单行文本框加入到“North”区域，包含各种按钮的面板Pane1p加入到”Center”区域。包含各种按钮的面板Panel p采用4行4列的网格布局，然后将数字按钮和运算符按钮以及控制按钮添加到面板中。事件处理分以下几种情况:数字按钮事件(”0”，”1”，”2”..”8”””、运算符按钮事件(”+””“，”\*”，””，”%”、正负号按钮事件("+/-“)、小数点按钮事件(”.”)、等号按钮事件(”=”)、求倒按钮事件(”1/x")、退格按钮事件(” 退格”、清楚按钮事件(“C”)。

触发按钮事件时，要进一步分析，是重新开始计算时触发的按钮事件还是计算中间触发的按钮事件。计算器完成的是一个数学表达式， 如3+2,所以可以采用一一个链表(LinkedList类)来存储数字或字符，如3+2分别存储在链表中，最后运算时，可以一一取出来进行运算。

计算器设计外形及结构如下：



1. 计算器功能模块设计

以下为标准计算器的基本功能：

加法

两个数据进行加操作，可以为小数。

(2)减法

两个数据进行减操作，可以为负数相减。

(3)乘法

两个数相乘操作。

(4)除法

两个数相除操作。

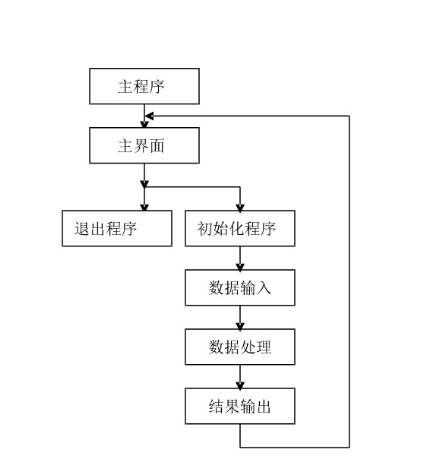
(5)清零

清除显示框的数据。

1. 计算器算法流程设计

计算器的程序设计，必然有数据的输入，当中就包含运算符和数字，都要通过计算器上的button按钮选项按键输入进去，里面录入的为字符型的数据，若要进行运算，要将输入的字符转换成double型，以防运算出错。

流程图设计：



1. 设计软件工具及代码

设计工具：Tower、java（jdk1.8）

代码如下：

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.StringTokenizer;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class JiSuan extends JFrame{

JPanel panel;

JTextField text;

JButton jb0,jb1,jb2,jb3,jb4,jb5,jb6,jb7,jb8,jb9; //0~9四个数字（可写成数组）

JButton bot,add,sub,mul,div,equ; //加减乘除以及小数点五个符号

public JiSuan(){

this.setBounds(300, 200, 400, 300); //窗体大小

this.setTitle("计算器"); //窗体名称

this.setLayout(new BorderLayout()); //设置计算机窗体的布局方式

text = new JTextField(15); //计算区域

//新建数字按钮

jb0 = new JButton("0");

jb1 = new JButton("1");

jb2 = new JButton("2");

jb3 = new JButton("3");

jb4 = new JButton("4");

jb5 = new JButton("5");

jb6 = new JButton("6");

jb7 = new JButton("7");

jb8 = new JButton("8");

jb9 = new JButton("9");

//新建符号按钮

bot = new JButton(".");

add = new JButton("+");

sub = new JButton("-");

mul = new JButton("\*");

div = new JButton("/");

equ = new JButton("=");

//新建计算器数字及符号所在区域面板

panel = new JPanel();

panel.setLayout(new GridLayout(4,4)); //设置面板的布局方式

//将各按钮组件放入面板（panel）中

//第一行

panel.add(jb7);

panel.add(jb8);

panel.add(jb9);

panel.add(div);

//第二行

panel.add(jb4);

panel.add(jb5);

panel.add(jb6);

panel.add(mul);

//第三行

panel.add(jb1);

panel.add(jb2);

panel.add(jb3);

panel.add(sub);

//第四行

panel.add(bot);

panel.add(jb0);

panel.add(equ);

panel.add(add);

this.add(text,BorderLayout.NORTH);

this.add(panel);

MyEvent(); //事件处理

this.setVisible(true); //设置窗体显示

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); //设置窗体可关闭

}

public void MyEvent(){

/\*将以下内容封装到Text方法中

\* jb0.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

// text.setText("0");

String str = jb0.getText();

System.out.println(str);

}

});\*/

//为各按钮添加事件处理

Text(jb0);

Text(jb1);

Text(jb2);

Text(jb3);

Text(jb4);

Text(jb5);

Text(jb6);

Text(jb7);

Text(jb8);

Text(jb9);

Text(bot);

Text(add);

Text(sub);

Text(mul);

Text(div);

//“=”按钮的事件处理

//这些代码的意思是 从你输入的东西里面 获取 含有哪些运算符 然后对应着进行运算

equ.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

text.setText(text.getText()+equ.getText());

if(text.getText().contains("+")){//做加法运算

StringTokenizer st1 = new StringTokenizer(text.getText(),"+");

double d1,d2 = 0;

d1 = Double.parseDouble(st1.nextToken());

while(st1.hasMoreTokens()){

StringTokenizer st2 = new StringTokenizer(st1.nextToken(),"=");

d2 = Double.parseDouble(st2.nextToken());

}

text.setText(text.getText()+" "+(d1+d2));

}else if(text.getText().contains("-")){//做减法运算

StringTokenizer st1 = new StringTokenizer(text.getText(),"-");

double d1,d2 = 0;

d1 = Double.parseDouble(st1.nextToken());

while(st1.hasMoreTokens()){

StringTokenizer st2 = new StringTokenizer(st1.nextToken(),"=");

d2 = Double.parseDouble(st2.nextToken());

}

text.setText(text.getText()+" "+(d1-d2));

}else if(text.getText().contains("\*")){//做乘法运算

StringTokenizer st1 = new StringTokenizer(text.getText(),"\*");

double d1,d2 = 0;

d1 = Double.parseDouble(st1.nextToken());

while(st1.hasMoreTokens()){

StringTokenizer st2 = new StringTokenizer(st1.nextToken(),"=");

d2 = Double.parseDouble(st2.nextToken());

}

text.setText(text.getText()+" "+(d1\*d2));

}else if(text.getText().contains("/")){//做除法运算

StringTokenizer st1 = new StringTokenizer(text.getText(),"/");

double d1,d2 = 0;

d1 = Double.parseDouble(st1.nextToken());

while(st1.hasMoreTokens()){

StringTokenizer st2 = new StringTokenizer(st1.nextToken(),"=");

d2 = Double.parseDouble(st2.nextToken());

}

text.setText(text.getText()+" "+(d1/d2)); //将运算结果放到text区域

}

}

});

}

//各按钮的事件处理方法

public void Text(JButton button){

button.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

// text.setText("0");

String str = button.getText();

// System.out.println(str);

text.setText(text.getText()+str);

}

});

// return button.getText();

}

//主函数

public static void main(String[] args) {

// TODO Auto-generated method stub

new JiSuan();

}

}

1. **设计结果**



**六、设计体会**

通过本次项目我们学到了好多、也发现了自己的好多不足!  
 很显然最明显的收获是理解和较熟练的掌握了Java图形界面的基本方法，尤其在Java图形界面的布局方面学到颇多，以前布局很乱并且很多布局都是无效的。在此次课设中通过查阅很多资料和同学的互相帮助，充分发挥了Java界面布局的优越性。另外按钮功能的实现也是本次项目的一大难点，怎样实现那些功能是关键。通过这次项目我们又学会了好多Java中的简单便捷的函数。  
 另外我们这次课设部分程序是通过外部资料查阅到的的，这使得我认识到查阅资料和开外多看书籍的重要性。但不主张照搬，因为那不是我们自己成果，那是不光彩的。

这次项目给我们的感受就是:只要你有心，不怕你完成不了项目。