莆田学院新工科产业学院

课 程 设 计

课程名称：JAVA程序设计

题 目：简易计算器

学生姓名：王斌

学 号：202012900127

专业班级：软工201

指导教师：刘少宇

**2022** 年 4 月 3 日

课程设计任务书

|  |  |
| --- | --- |
| **设计题目** | **简易计算器** |
| **课**  **程**  **设**  **计**  **主**  **要**  **内**  **容**  **及**  **要**  **求** | 利用 Java 开发一个可以进行简单的四则运算的图形化计算器（以下为参考界面，同学们自行根据功能设计界面）。     1. 基本功能要求如下：   1、要求计算器至少能够实现基本的加减乘除运算和清空功能。  2、C按钮用于实现清空操作，直接将显示置零。  以上为基本功能，想要更好的成绩，需要有扩展功能，扩展功能鼓励同学发挥自己的想象力，以下仅为参考。   1. 扩展功能：   a.实际上按资料做完的计算器存在一定问题，比如连续的加减乘除并没有实现。例如输入7\*8然后不输入=号，而是输入+号，这个时候不能得到结果。  b.能否有更多的计算功能，比如开方等  c.除了显示结果，还有显示输入。比如：当你输入7+8+9的时候，屏幕有两行，第一显示7+8+9，第2行显示结果24  d.在c基础上能否实现括号功能  其他：更多功能(进制转换)、更漂亮界面、更方便的操作、支持对大数的加减法、支持对运算结果的保存与读取。  等等等。 |
| **日**  **程**  **安**  **排** | 2022.02.28-2022.03.01：完成分组、任务的调研、需求的分析以及组内任务的分工。  2022.03.02-2022.03.04：软件的设计、代码的编写与调试。  2022.03.05-2022.03.06：软件的测试和设计报告的撰写。  本学期第7周周末前，需完成答辩、报告以及代码的提交、资料整理等工作。 |
| **指导老师签名： 202 年 月 日** | |

**课程设计成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| 课程设计题目 | **简易计算器** | | |
| 学生姓名 | **王斌** | 总成绩 |  |
| 平时成绩 |  | 设计报告成绩 |  |
| 指导教师评语（建议：从学生的工作态度、工作量、设计任务的完成情况及设计报告书面表达能力等方面给出评价。）  **指导教师（签名）：**  **年 月 日** | | | |

目录

1. 概述

通过java的JFrame窗口功能制作一个简易的计算器，拥有计算器加减乘除的功能。

1. 方案设计

package cn.lidan.start;  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Carculator extends JFrame implements ActionListener{  
 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*北面按键\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  
 private JPanel jp\_north = new JPanel();  
 private JTextField input\_text = new JTextField();  
 private JButton c\_Btn = new JButton("c");  
  
 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*中间按键\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  
 private JPanel jp\_center = new JPanel();  
  
 public Carculator() throws HeadlessException {  
 this.init();  
 this.addNorthComponent();  
 this.addCenterButton();  
 }  
 //初始化  
 public void init(){  
 this.setTitle("计算器");  
 this.setSize(400,400);  
 this.setResizable(false);  
 this.setLayout(new BorderLayout());  
 this.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 this.setLocationRelativeTo(null);  
  
 }  
 //北面按键  
 public void addNorthComponent() {  
 this.input\_text.setPreferredSize(new Dimension(330, 30));  
 jp\_north.add(input\_text);  
 this.c\_Btn.setForeground(Color.*RED*);  
 jp\_north.add(c\_Btn);  
  
 c\_Btn.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 input\_text.setText("");  
 }  
 });  
  
 this.add(jp\_north, BorderLayout.*NORTH*);  
 }  
 //中间的按钮  
 public void addCenterButton(){  
 String btn\_text = "123+456-789\*0.=/";  
 String regex = "[\\+\\-\*/=.]";  
 this.jp\_center.setLayout(new GridLayout(4,4));  
 for(int i = 0;i<16;i++){  
 String temp = btn\_text.substring(i,i+1);  
 JButton btn = new JButton();  
 btn.setText(temp);  
 if(temp.matches(regex)){  
 btn.setFont(new Font("粗体",Font.*BOLD*,20));  
 btn.setForeground(Color.*RED*);  
 }  
 /\* if(temp.equals("+")||temp.equals("-")||temp.equals("\*")||temp.equals("/")||temp.equals("=")){  
 btn.setFont(new Font("粗体",Font.BOLD,20));  
 btn.setForeground(Color.RED);  
 }\*/  
 btn.addActionListener(this);  
 jp\_center.add(btn);  
 }  
 this.add(jp\_center,BorderLayout.*CENTER*);  
 }  
  
 public static void main(String[] args) {  
 Carculator carculator = new Carculator();  
 carculator.setVisible(true);  
 }  
  
 private String fristInput = null;  
 private String operator = null;  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String clickStr = e.getActionCommand();  
 if(".0123456789".indexOf(clickStr) != -1){  
 this.input\_text.setText( input\_text.getText() + clickStr);  
 this.input\_text.setHorizontalAlignment(JTextField.*RIGHT*);  
 // JOptionPane.showMessageDialog(this,clickStr);  
 }  
 else if (clickStr.matches("[\\+\\-\*/]{1}")){  
 operator = clickStr;  
 fristInput = this.input\_text.getText();  
 this.input\_text.setText("");  
 }else if(clickStr.equals("=")){  
 Double a = Double.*valueOf*(fristInput);  
 Double b = Double.*valueOf*(this.input\_text.getText());  
 Double result = null;  
 switch (operator){  
 case "+":  
 result = a+b;  
 break;  
 case "-":  
 result = a-b;  
 break;  
 case "\*":  
 result = a\*b;  
 break;  
 case "/":  
 if (b!=0){  
 result = a/b;  
 }  
 break;  
  
 }  
 this.input\_text.setText(result.toString());  
  
 }  
  
 }  
}

1. 功能描述与实现

**窗口的建立**：

this.setTitle("计算器");  
this.setSize(400,400);  
this.setResizable(false);  
this.setLayout(new BorderLayout());  
this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
this.setLocationRelativeTo(null);

首先给建立的窗口命名为计算器，然后设置高度和长度，设置关闭窗口时退出程序，令窗口大小固定，随后使窗口打开时位于屏幕中间。

**输入框和清空按钮：**

private JPanel jp\_north = new JPanel();  
private JTextField input\_text = new JTextField();  
private JButton c\_Btn = new JButton("c");

创立一个输入框和一个按钮，给按钮命名为c。

this.input\_text.setPreferredSize(new Dimension(330, 30));  
jp\_north.add(input\_text);  
this.c\_Btn.setForeground(Color.RED);  
jp\_north.add(c\_Btn);

给输入框设置大小，并且把按钮颜色变成红色。

c\_Btn.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 input\_text.setText("");  
 }  
});

如果监听到清空键，则使屏幕上的值变为空。

数字键与运算符：

private JPanel jp\_center = new JPanel();

先建立一个面板。

String btn\_text = "123+456-789\*0.=/";  
String regex = "[\\+\\-\*/=.]";  
this.jp\_center.setLayout(new GridLayout(4,4));

输入数字和运算符，把面板平均分4x4的网格布局。

for(int i = 0;i<16;i++){  
 String temp = btn\_text.substring(i,i+1);  
 JButton btn = new JButton();  
 btn.setText(temp);

通过循环给每个小按钮分配值。

if(temp.matches(regex)){  
 btn.setFont(new Font("粗体",Font.BOLD,20));  
 btn.setForeground(Color.RED);

判断是否是运算符，如果是运算符则按要求把字体调大并且字体变成红色。

this.add(jp\_center,BorderLayout.CENTER);

使这个4x4的窗口加入到初始化的面板中间。

监听与运算：

btn.addActionListener(this);  
jp\_center.add(btn);

监听被点击的按钮。

if(".0123456789".indexOf(clickStr) != -1){  
 this.input\_text.setText( input\_text.getText() + clickStr);  
 this.input\_text.setHorizontalAlignment(JTextField.RIGHT);}

判断监听到的内容是否为非运算符，得到值之后使监听到的值现实在屏幕右边。

else if (clickStr.matches("[\\+\\-\*/]{1}")){  
 operator = clickStr;  
 fristInput = this.input\_text.getText();  
 this.input\_text.setText("");

判断运算符，如果监听到的是运算符，则清空屏幕中显示的内容。

else if(clickStr.equals("=")){  
 Double a = Double.valueOf(fristInput);  
 Double b = Double.valueOf(this.input\_text.getText());  
 Double result = null;  
 switch (operator){  
 case "+":  
 result = a+b;  
 break;  
 case "-":  
 result = a-b;  
 break;  
 case "\*":  
 result = a\*b;  
 break;  
 case "/":  
 if (b!=0){  
 result = a/b;  
 }  
 break;  
  
 }

this.input\_text.setText(result.toString());

如果监听到的是=，则对之前监听到的功能进行判断，并进行相应的运算，然后把结果显示到屏幕上。

1. 结论分析

制作了一个简易的计算器，实现了创建窗口功能，创建按键，显示器等功能，并且给按键赋值、监听按键，然后通过监听实现计算器的运算功能。计算器还有很多的不足，比如不能进行连续的运算，除了加减乘除没有更多的功能。学到了很多的东西。

1. 参考文献

[1] java码农先生,java计算器程序 ,bilibili ,2021