

面向对象Java—实验报告

****

**实验五： GUI** .

**姓 名：**  於俊涛

**学 号： 202007020625**

**专业班级： 计算机203**

**系 别： 计算机系**

**学 院： 电子信息与人工智能学院**

**实验五 GUI**

1. **实验目的**

通过图形化界面设计相关类、接口等，实现用户图形化应用程序的开发；进一步巩固JDBC连接数据库以及文件读写操作。

1. **实验目标**

（一）能够掌握常用GUI控制组件的使用方法，通过java.awt包以及Javax.swing包中的类和接口实现用户图形界面的开发；

（二）能够运用Java的事件处理机制，通过JDBC操作数据库，实现用户登录功能；

（三）能够掌握利用I/O流对文件进行操作。

1. **实验内容**

**3.1 实验环境**

Eclipse IDE for Java Developers

**3.2 具体实验内容**

1.利用GUI设计实现一个计算器程序（注：至少要实现基本的加减乘除运算四则运算）。

2.设计一个关于文件操作的图形化应用程序，至少实现以下功能：

a)包含一个文本框以及添加按钮，在文本框中输入文字后，点击添加按钮可以在文件中写入文本框中的文字；

b)包含一个读取按钮，点击该按钮后，可以读取文件内容，并显示到文本框中。

1. **实验过程分析**

**4.1 实验步骤**

1.利用GUI设计实现一个计算器程序（注：至少要实现基本的加减乘除运算四则运算）。

实验代码：

package 1;

//计算器，可加减乘除混合运算

import java.awt.BorderLayout;

import java.awt.Container;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Stack;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.Border;

public class MyCalc {

public static void main(String[] args) {

new myFrame("Calculator");

}

}

class myFrame extends JFrame {

JTextField text = new JTextField();

JButton[] bt1 = new JButton[16];

JButton[] bt2 = new JButton[3];

String key2[] = new String[]{ "BackSpace","Clear", "C"};

String key1[] = new String[] { "1", "2", "3", "+", "4", "5", "6", "-", "7", "8", "9", "\*",

"0", ".", "=", "/" };

JPanel key1Panel = new JPanel();

JPanel key2Panel = new JPanel();

boolean finish = false;

double ans=0;

public myFrame(String title) {

super(title);

init();

}

public void init() {

text.setHorizontalAlignment(JTextField.RIGHT);

text.setPreferredSize(new Dimension(0, 40));

text.setFont(new Font("微软雅黑", Font.BOLD, 20));

for (int i = 0; i < bt1.length; i++) {

bt1[i] = new JButton(key1[i]);

}

for (int i = 0; i < bt2.length; i++) {

bt2[i] = new JButton(key2[i]);

}

Container pane = this.getContentPane();

this.add(text, BorderLayout.NORTH);

key1Panel.setLayout(new GridLayout(5, 4, 3, 3));

this.add(key2Panel, BorderLayout.CENTER);

this.add(key1Panel, BorderLayout.SOUTH);

for (int i = 0; i < bt1.length; i++) {

key1Panel.add(bt1[i]);

bt1[i].addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (finish == true) {

text.setText("");

finish = false;

}

String command = e.getActionCommand();

String s = text.getText();

if ("0123456789.+-\*/".indexOf(command) >= 0)

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

else if (command.equals("Clear"))

text.setText(s.substring(0, s.length() - 1));

else if (command.equals("C") || command.equals("BackSpace"))

text.setText("");

else if (command.equals("=")) {

ans=new calc(s).result;

text.setText(text.getText() + "="+ans);

finish = true;

}

}

});

}

for (int i = 0; i < bt2.length; i++) {

key2Panel.add(bt2[i]);

bt2[i].addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (finish == true) {

text.setText("");

finish = false;

}

String command = e.getActionCommand();

String s = text.getText();

if ("0123456789.+-\*/".indexOf(command) >= 0)

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

else if (command.equals("Clear"))

text.setText(s.substring(0, s.length() - 1));

else if (command.equals("C") || command.equals("BackSpace"))

text.setText("");

else if (command.equals("=")) {

ans=new calc(s).result;

text.setText(text.getText() + "="+ans);

finish = true;

}

}

});

}

this.setVisible(true);

this.setLocation(500, 200);

this.setSize(500, 400);

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

}

}

class calc {

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

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

String exp;

double result;

public calc(String s) {

exp = s;

calctor();

}

public void calctor() {

String s[] = tranform(exp).split(" ");

for (String string : s) {

if (string.matches("\\d+") || string.matches("\\d+.\\d+")) {

double n = Double.parseDouble(string);

num.push(n);

} else if (string.equals("+") || string.equals("-") || string.equals("/") || string.equals("\*")) {

double op1 = num.pop(), op2 = num.pop();

switch (string) {

case "+":

num.push(op1 + op2);

break;

case "-":

num.push(op2 - op1);

break;

case "\*":

num.push(op2 \* op1);

break;

case "/":

num.push(op2 / op1);

break;

default:

break;

}

} else {

}

}

result = num.peek();

// System.out.println("ans= " + result);

}

public String tranform(String s) {

s = s.replace("+", " + ");

s = s.replace("-", " - ");

s = s.replace("\*", " \* ");

s = s.replace("/", " / ");

String str[] = s.split(" ");

String s1 = "";

for (String string : str) {

if (string.matches("\\d+") || string.matches("\\d+.\\d+")) {

s1 = s1 + " " + string;

} else {

if (ops.isEmpty()) {

ops.push(string);

continue;// 跳过下面

}

String temp = ops.peek();

while (priority(temp) >= priority(string)) {// 栈顶元素优先级高于或等于进栈元素,不断弹出栈顶元素

s1 = s1 + " " + temp;

ops.pop();

if (ops.isEmpty())

break;

temp = ops.peek();

}

ops.push(string);

}

}

while (!ops.isEmpty())

s1 = s1 + " " + ops.pop();

// System.out.println(s1);

return s1;

}

public int priority(String s) {

switch (s) {

case "+":

case "-":

return 1;

case "\*":

case "/":

return 2;

default:

break;

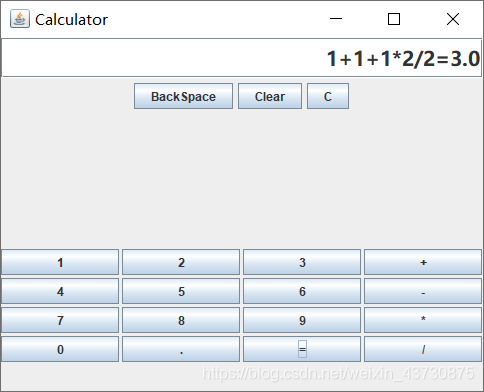
}

return 0;

}

}

结果输出：



2. 设计一个关于文件操作的图形化应用程序，至少实现以下功能：

（1）包含一个文本框以及添加按钮，在文本框中输入文字后，点击添加按钮可以在文件中写入文本框中的文字；

（2）包含一个读取按钮，点击该按钮后，可以读取文件内容，并显示到文本框中。

实验代码

import lombok.SneakyThrows;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.\*;

import java.nio.charset.StandardCharsets;

class FileSaveOpen extends JFrame {

private final JFrame thisFrame = this;

private final JPanel rootPanel = new JPanel();

private final JLabel titleLabel = new JLabel("文 本");

private final JButton saveButton = new JButton("保存");

private final JButton openButton = new JButton("打开");

private final JTextArea inputTextArea = new JTextArea();

private final JTextArea outputTextArea = new JTextArea();

private final JLabel TipsLable = new JLabel();

public static void main(String[] args) throws InterruptedException {

new FileSaveOpen("");}

FileSaveOpen(String name) {

super(name);

this.setBounds(new Rectangle(600, 300, 400, 300));

this.setDefaultCloseOperation(WindowConstants.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.dataInitialize();

this.componentInitial();

this.setParameter();

this.dealLogic();}

private void dataInitialize() { }

private void componentInitial() {

this.setComponent();

this.setComponentsLayout();

this.setComponentsPreferredSize();

this.setComponentsHorizontalAlignment();}

private void setComponent() {

rootPanel.add(titleLabel);

rootPanel.add(saveButton);

rootPanel.add(openButton);

rootPanel.add(inputTextArea);

rootPanel.add(TipsLable);

rootPanel.add(outputTextArea);}

private void setComponentsLayout() {

this.setContentPane(rootPanel);

rootPanel.setLayout(null);

titleLabel.setBounds(new Rectangle(30, 0, 30, 25));

saveButton.setBounds(new Rectangle(240, 0, 60, 25));

openButton.setBounds(new Rectangle(320, 0, 60, 25));

inputTextArea.setBounds(new Rectangle(5, 30, 375, 50));

TipsLable.setBounds(new Rectangle(100,85,375,25));

outputTextArea.setBounds(new Rectangle(5,120,375,150));}

private void setComponentsPreferredSize() {

titleLabel.setPreferredSize(new Dimension(100, 100));}

private void setComponentsHorizontalAlignment() {

titleLabel.setHorizontalAlignment(SwingConstants.CENTER);

saveButton.setHorizontalAlignment(SwingConstants.CENTER);

openButton.setHorizontalAlignment(SwingConstants.CENTER);}

private void setParameter() {}

private void dealLogic() {

saveButton.addActionListener(new saveButtonActionListener());

openButton.addActionListener(new openButtonActionListener());}

class saveButtonActionListener implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

TipsLable.setFont(new Font("黑体",Font.PLAIN,15));

String contents = inputTextArea.getText();

File file = new File("../data.txt");

if (!file.exists()) {

try {

if (!file.createNewFile()) {

TipsLable.setText("文件创建失败");

TipsLable.setForeground(Color.red);

return;

}

} catch (IOException ex) {

ex.printStackTrace();}}

FileOutputStream fileOutputStream = null;

try {

fileOutputStream = new FileOutputStream(file);

fileOutputStream.write(contents.getBytes(StandardCharsets.UTF\_8));

fileOutputStream.flush();

fileOutputStream.close();

TipsLable.setText("文件写入成功");

TipsLable.setForeground(Color.BLUE);

} catch (IOException ex) {

ex.printStackTrace();

TipsLable.setText("文件写入出现问题");

TipsLable.setForeground(Color.red);}}}

class openButtonActionListener implements ActionListener {

@SneakyThrows

@Override

public void actionPerformed(ActionEvent e) {

int length = 10000000;

byte[] buffer = new byte[length];

File file = new File("../data.txt");

try {

InputStream inputStream = new FileInputStream(file);

int n = inputStream.read(buffer, 0, length);

inputStream.close();

String str = new String(buffer, 0, n, StandardCharsets.UTF\_8);

outputTextArea.setText(str);

} catch (FileNotFoundException c) {

c.printStackTrace();}}}}

结果输出：



**4.2 错误分析**

1.问题复现：出现Could not create connection to database server.报错，检查自己的url语句并没有错误，数据库的用户名和密码也都正确。

问题分析：经排查是，MySQL8.0版本需要更换驱动为com.mysql.cj.jdbc.Driver，之前的com.mysql.jdbc.Driver已经不能在MySQL 8.0版本使用了.

解决方案：使用select version() from dual;,查询获得我的MySQL版本为8.0.27.然后从Maven Repository: mysql » mysql-connector-java » 8.0.27 (mvnrepository.com)下载驱动，并将JDBC\_DRVER的驱动从com.mysql.jdbc.Driver改为com.mysql.cj.jdbc.Driver.

1. **实验总结**

通过本次实验学习并掌握了java GUI的相关操作,并通过接入数据库以及用户登陆操作体验了图形化界面应用程序的开发。并在数据库连接的过程中解决了相关问题，加深了对于jadbc使用的相关操作的熟练程度。这对于我们今后的程序开发设计很有帮助，通过本次实验也尝试了GUI与文件操作的搭配使用进行文件可视化读写操作。通过本次实验不仅学习了GUI相关知识，也进一步巩固JDBC连接数据库以及文件读写操作。