# 实验二 图形用户界面设计

学号：14111204075 姓名：章文

## 实验目的

1. 掌握常用GUI构件及其布局管理的使用方法。

2. 掌握事件处理的方法。

3．了解Swing控件的概念。

## 实验内容

1. 完成下列程序的填空。这是一个标题为“计算”的窗口，窗口的布局为FlowLayout布局。窗口中添加两个文本区，当在第一个文本区中输入若干个数时，另一个文本区同时对输入的数进行求和与求平均值运算。

编程分析：填空题……照着填就好了，分析就在程序注释里

源代码：

**Compute.java**

package exp2\_1;

public class Compute {

int count;

double[] num;

double sum;

double ave;

Compute(){

count = 0;

num = null;

ave = 0;

sum = 0;

}

void setnum(String s){

String[] a = s.split("[^0123456789.]+");

int n = a.length;

num = new double[n]; **//分配内存空间**

count = 0; //初始化

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

try{

num[i] = Integer.valueOf(a[i]); **//把字符数组的内容转换成数字**

count++; **//计数器+1**

}catch(Exception e){

num[i] = 0;

count++; **//无法转换的字符串算作0，而且计数器也要+1**

}

}

void add\_aver(){

sum = 0;

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

sum+=num[i]; **//总数加上数组里面的有效数字**

}

ave = sum / count; **//计算平均数**

}

}

**ComputeWindows.java**

package exp2\_1;

import java.awt.FlowLayout;

import javax.swing.\*;

public class ComputeWindow extends JFrame{

JTextArea text\_input,text\_result;

TextListener changetxtlistener;

Compute cp;

void init(){

cp = new Compute(); **//新建Compute对象**

text\_input = new JTextArea(6,20); **//新建JTextArea对象**

text\_result = new JTextArea(6,20);

this.setLayout(new FlowLayout());

this.add(new JScrollPane(text\_input));

this.add(new JScrollPane(text\_result)); **//让text\_result带滚动条**

text\_result.setEditable(false);

changetxtlistener = new TextListener(); **//新建TextListener**

changetxtlistener.setInputText(text\_input); **//把text\_input传递到changetxtlistener里**

changetxtlistener.setShowText(text\_result)**;//同上**

changetxtlistener.setCompute(cp);**//同上**

(text\_input.getDocument()).addDocumentListener(changetxtlistener);

}**//给text\_input添加监听器**

ComputeWindow(){

init();

this.setTitle("计算");

this.setBounds(100,100,400,300);

this.setVisible(true);

this.validate();

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

}

}

**Exp2\_1.java**

package exp2\_1;

public class Exp2\_1 {

public static void main(String[] args) {

// TODO code application logic here

ComputeWindow win = new ComputeWindow();**//新建窗体对象**

}

}

**TextListener.java**

package exp2\_1;

import javax.swing.JTextArea;

import javax.swing.event.DocumentEvent;

import javax.swing.event.DocumentListener;

public class TextListener implements DocumentListener{

JTextArea inputText,showText;

Compute cmp;

public void setInputText(JTextArea text){

inputText = text; **//inputText指向text对应的对象**

}

public void setShowText(JTextArea text){

showText = text; **//ishowText指向text对应的对象**

}

public void setCompute(Compute cp){

cmp = cp; **//cmp指向cp对应的对象**

}

public void changedUpdate(DocumentEvent e){

String text = inputText.getText(); **//获取inputText的内容**

cmp.setnum(text); **//获取数值**

cmp.add\_aver(); **//计算和、平均值**

showText.setText(null);

showText.append("\n和："+cmp.sum); **//显示和**

showText.append("\n平均值："+cmp.ave); **//显示平均值**

}

public void removeUpdate(DocumentEvent e){

changedUpdate(e); **//当字符被删除时调用changedUpdate函数**

}

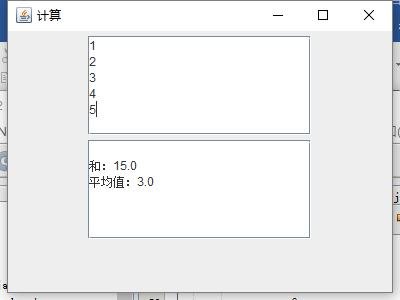
public void insertUpdate(DocumentEvent e){

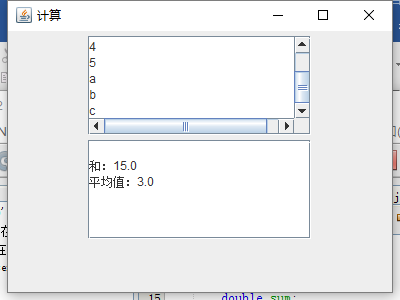
changedUpdate(e); **//当字符被插入的时候调用changedUpdate函数**

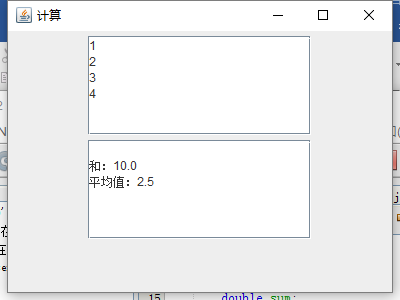
}

}

运行结果：





（图片按照时间顺序排列）

2. 验证性实验。请回答该程序的功能是什么，并将运行情况截图。

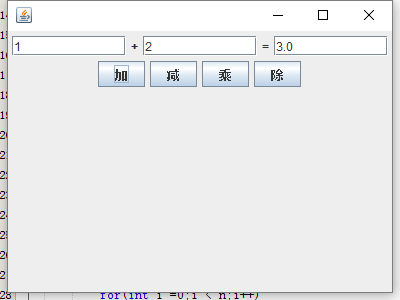
（题目略）

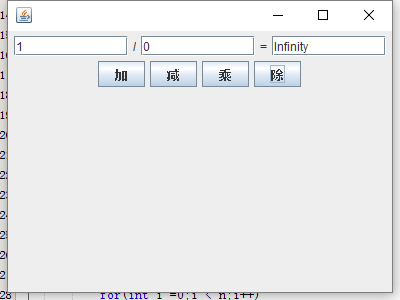
编程分析：对前两个框的数字进行加/减/乘/除运算，如果输入不符合要求会在最后一个textfeild进行提示。

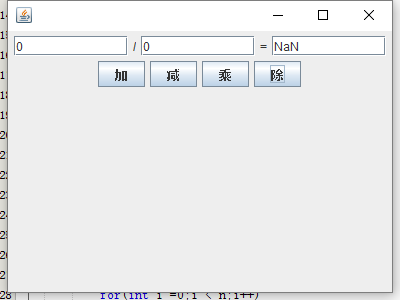
源代码：略（反正都是照抄）

运行结果：







（Not a Number）

3. 请在第2题的基础上修改，该程序要求实现计算机自动出题，并统计显示总题数和答对的题数。该窗口中有两个按钮Buttonproblem和Buttonok，单击Buttonproblem按钮，则计算机自动出题，单击Buttonok按钮，则表示本题计算结果已输入；有三个文本框text1、text2和text3，禁止在窗口的文本框text1和text2中输入数值，文本框text1和text2只是用来显示系统随机生成的两个100以内的整数，text3中输入计算结果；标签labelop所显示的算术运算符由系统随机生成；标签labeltj显示总题数和答对的题数。

编程分析:

jTextNum1和jTextNum2用来显示两个操作数，并在构造函数中将他们的Editable属性设置为false(不能修改内容);

jLabelOp显示运算符,jLabelEqual显示等于号

jButtonProblem出题，jButtonOk计算答案是否正确,jLabelState显示总体木薯和正确题目数。

自定义Generator类生成随机运算符和操作数，在生成除法的操作数的时候，先生成两个10以内的不为0的整数，然后相乘得到的答案作为第一个操作数，然后任取剩下的数作为第二个操作数。

Mainframe类中有两个变量，total和correct来计数。

如果在没有开始生成题目的情况下单击OK按钮将不会有任何反应（判断gen，Generator类对象是否被初始化）

按下jButtonProblem以后开始初始化gen对象，然后开始生成题目，按下OK键以后会判断答案是否正确并计数，更新jLabelState的内容。

源代码：

**MainFrame.java**

package exp2\_3;

public class MainFrame extends javax.swing.JFrame {

Generator gen;

int total;

int correct;

public MainFrame() {

initComponents();

jTextNum1.setEditable(false);

jTextNum2.setEditable(false);

}

private void initComponents() {

jTextNum1 = new javax.swing.JTextField();

jLabelOp = new javax.swing.JLabel();

jTextNum2 = new javax.swing.JTextField();

jLabelEqual = new javax.swing.JLabel();

jTextResult = new javax.swing.JTextField();

jButtonProblem = new javax.swing.JButton();

jButtonOK = new javax.swing.JButton();

jLabelState = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jTextNum1.setMinimumSize(new java.awt.Dimension(100, 25));

jTextNum1.setPreferredSize(new java.awt.Dimension(100, 25));

jTextNum1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextNum1ActionPerformed(evt);

}

});

jLabelOp.setText("+");

jLabelOp.setMaximumSize(new java.awt.Dimension(8, 15));

jLabelOp.setMinimumSize(new java.awt.Dimension(8, 15));

jLabelOp.setPreferredSize(new java.awt.Dimension(8, 15));

jTextNum2.setMinimumSize(new java.awt.Dimension(100, 25));

jTextNum2.setPreferredSize(new java.awt.Dimension(100, 25));

jLabelEqual.setText("=");

jLabelEqual.setMaximumSize(new java.awt.Dimension(8, 15));

jLabelEqual.setMinimumSize(new java.awt.Dimension(8, 15));

jLabelEqual.setPreferredSize(new java.awt.Dimension(8, 15));

jTextResult.setMinimumSize(new java.awt.Dimension(100, 25));

jTextResult.setPreferredSize(new java.awt.Dimension(100, 25));

jButtonProblem.setText("出题");

jButtonProblem.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonProblemActionPerformed(evt);

}

});

jButtonOK.setText("确认");

jButtonOK.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButtonOKActionPerformed(evt);

}

});

jLabelState.setText("<html>\n总题目数：<br>\n正确题目数：\n</html>");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(12, 12, 12)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jButtonProblem)

.addGroup(layout.createSequentialGroup()

.addComponent(jTextNum1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabelOp, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jTextNum2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabelEqual, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jLabelState, javax.swing.GroupLayout.PREFERRED\_SIZE, 94, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButtonOK)

.addComponent(jTextResult, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(16, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(46, 46, 46)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabelOp, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabelEqual, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextResult, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextNum2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextNum1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 60, Short.MAX\_VALUE)

.addComponent(jLabelState, javax.swing.GroupLayout.PREFERRED\_SIZE, 39, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(49, 49, 49)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButtonProblem)

.addComponent(jButtonOK))

.addGap(58, 58, 58))

);

pack();

}// </editor-fold>

private void jTextNum1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButtonProblemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(gen == null)

gen = new Generator();

gen\_Problem();

total = 0;

correct = 0;

set\_Labels();

}

private void gen\_Problem()

{

gen.Generate();

jTextNum1.setText(Integer.toString(gen.getNum1()));

jTextNum2.setText(Integer.toString(gen.getNum2()));

jLabelOp.setText(gen.getOp());

}

private void jButtonOKActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(gen == null)

return;

else

{

if(!jTextResult.getText().isEmpty() && Integer.valueOf(jTextResult.getText()) == gen.getResult())

correct++;

}

total++;

set\_Labels();

gen\_Problem();

}

private void set\_Labels()

{

jLabelState.setText("<html> 总题目数：" + total + "<br> 正确题目数：" + correct + "</html>");

jTextResult.setText("");

}

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new MainFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButtonOK;

private javax.swing.JButton jButtonProblem;

private javax.swing.JLabel jLabelEqual;

private javax.swing.JLabel jLabelOp;

private javax.swing.JLabel jLabelState;

private javax.swing.JTextField jTextNum1;

private javax.swing.JTextField jTextNum2;

private javax.swing.JTextField jTextResult;

// End of variables declaration

}

**Generator.java**

package exp2\_3;

public class Generator {

int num1,num2;

int result;

String operator;

Random rand;

public Generator(){

rand = new Random();

}

void Generate(){

num1=0;

num2=0;

int index;

index=rand.nextInt(4);

switch(index)

{

case 0:

operator = "+";

num1=rand.nextInt(100);

num2=rand.nextInt(100);

result=num1+num2;

break;

case 1:

operator = "-";

num1=rand.nextInt(100);

num2=rand.nextInt(100);

result=num1-num2;

break;

case 2:

operator = "\*";

num1=rand.nextInt(100);

num2=rand.nextInt(100);

result=num1\*num2;

break;

case 3:

operator = "/";

while(num1==0)

num1=rand.nextInt(10);

while(num2==0)

num2=rand.nextInt(10);

result=num1\*num2;

int temp=num1;

num1=result;

result=temp;

break;

}

}

int getNum1(){

return num1;

}

int getNum2(){

return num2;

}

String getOp(){

return operator;

}

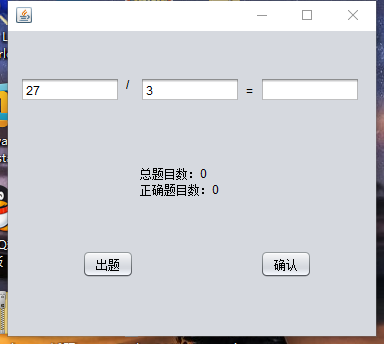
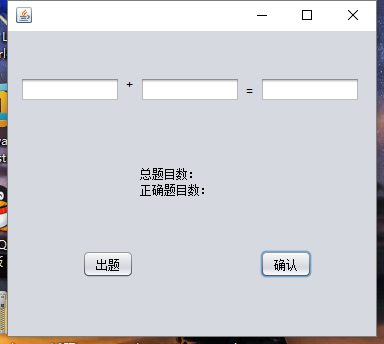
int getResult(){

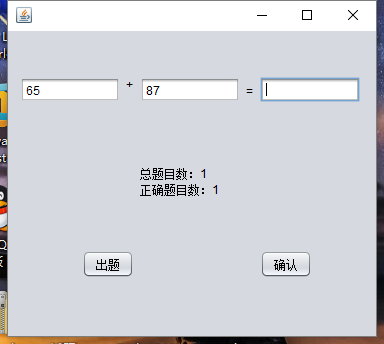
return result;

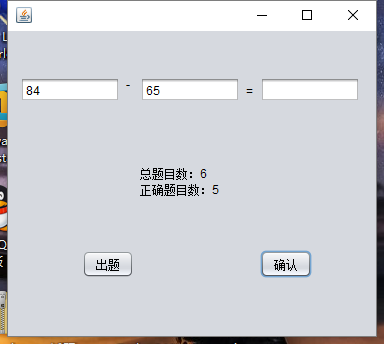
}

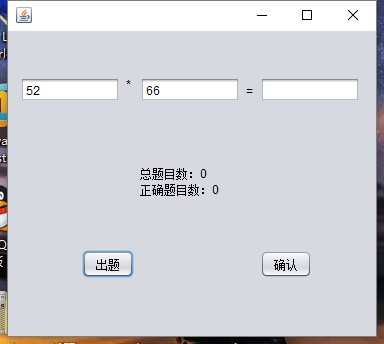
}

运行结果：





//上一题回答错误或者没有回答

//又按了出题按钮

调试分析：Integer.valueOf()函数不接受空字符串（会抛出异常）所以当没有输入数值的时候程序虽然会继续运行但是会在控制台看到程序抛出Exception，而且不会继续出题，所以修改了判断逻辑。

4. 通过以下添加学生信息的案例，掌握Java的事件处理机制以及对话框技术的应用。操作步骤如下：

（步骤太长，略）

编程分析：问题的话。。。

没有验证输入有效性（身份证号有效性和姓名是否有效）

用户可以直接修改TextArea的值

还有一个不知道算不算（其实ActionPerformed响应右键单击）

源代码：(只列出来修改的代码)

在StudentDialog.java里面的buttonOK的监听器中添加正则表达式验证身份证号，姓名的话只判断是否输入。

String regEx="^[1-9]\\d{5}[1-9]\\d{3}((0\\d)|(1[0-2]))(([0|1|2]\\d)|3[0-1])\\d{4}$";

Pattern pattern = Pattern.compile(regEx);

Matcher matcher = pattern.matcher(textFieldidenId.getText());

matcher.matches();

if(!matcher.matches())

{

JOptionPane.showMessageDialog(null, "身份证号输入有误", "提示", JOptionPane.ERROR\_MESSAGE);

return;

}

if(textFieldname.getText().isEmpty())

{

JOptionPane.showMessageDialog(null, "姓名不能为空", "提示", JOptionPane.ERROR\_MESSAGE);

return;

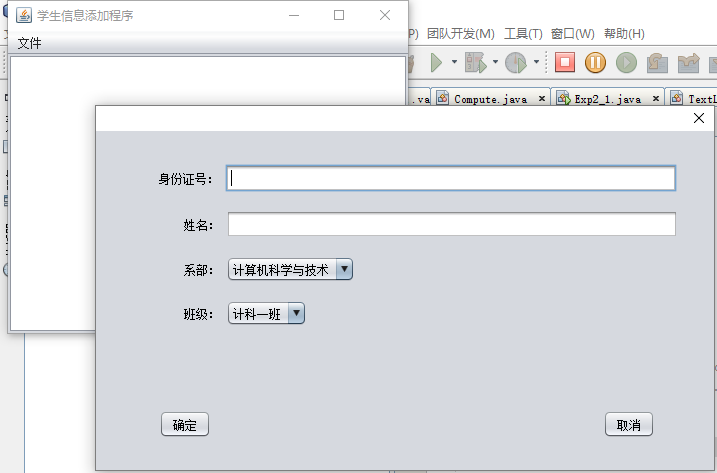
}

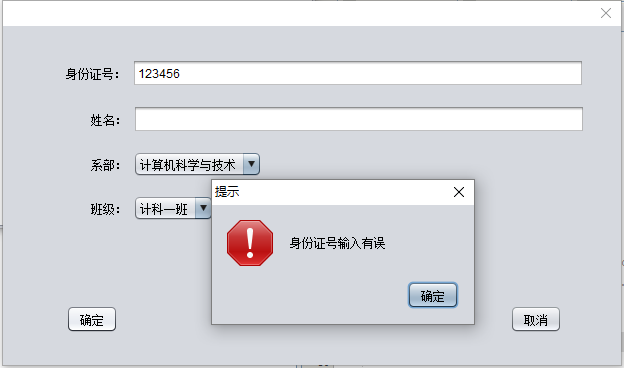
ok = true;

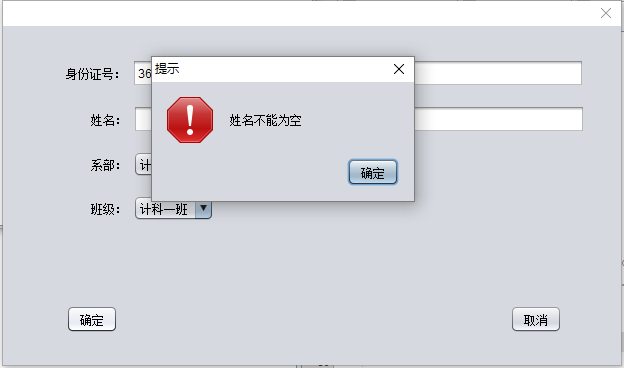
this.setVisible(false);

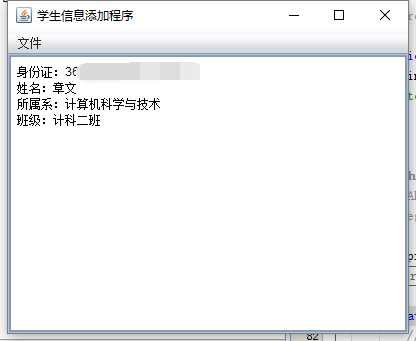
另外在StudentFrame的构造函数中把textArea的Editable设置为false。

截图：









调试分析：偶尔会出现程序窗口混叠的情况（有可能是我的显卡驱动有问题，因为VS在调试的过程中偶尔也会这样）

还有Netbeans似乎会自动对转义字符前面加\，结果我粘贴进去的正则表达式已经加好了\又给我加了一遍……程序总是显示身份证有错我还以为哪里有问题……

还有写报告的时候才想起来其实可以添加一个清屏的功能来着……

5.思考题

(1) Java的事件处理机制是怎样的？

如果用户用鼠标单击了按钮对象button，则该按钮button就是事件源，而java运行时系统会生成ActionEvent类的对象actionE，该对象中描述了该单击事件发生时的一些信息，然后，事件处理者对象将接收由java运行时系统传递过来的事件对象actionE并进行相应的处理。

　　由于同一个事件源上可能发生多种事件，因此java采取了授权处理机制(Delegation Model)，事件源可以把在其自身所有可能发生的事件分别授权给不同的事件处理者来处理。比如在Canvas对象上既可能发生鼠标事件，也可能发生键盘事件，该Canvas对象就可以授权给事件处理者一来处理鼠标事件，同时授权给事件处理者二来处理键盘事件。有时也将事件处理者称为监听器，主要原因也在于监听器时刻监听着事件源上所有发生的事件类型，一旦该事件类型与自己所负责处理的事件类型一致，就马上进行处理。授权模型把事件的处理委托给外部的处理实体进行处理，实现了将事件源和监听器分开的机制。事件处理者（监听器）通常是一个类，该类如果要能够处理某种类型的事件，就必须实现与该事件类型相对的接口。例如例5.9中类ButtonHandler之所以能够处理ActionEvent事件，原因在于它实现了与ActionEvent事件对应的接口ActionListener。每个事件类都有一个与之相对应的接口。

(2) 事件监听器和事件适配器的关系是什么？

事件适配器是事件监听器的实现。

事件监听器:

一个事件监听器对象负责处理一类事件

一类事件的每一种发生情况,分别由事件监听器对象中的一个方法来具体处理.

在事件源和事件监听器对象中进行约定的接口类,被成为事件监听器接口.

事件监听器接口类的名称与事件类的名称是相对应的,例如MouseEvent事件类的监听器接口名为MouseListener.

处理发生在某个GUI组件上的XXXEvent事件的某种情况,其事件处理的通用编写流程:编写一个实现了XXXListener接口的事件监听器类.

XXXListener类中的用于处理该事件情况的方法中,编写处理代码.

调用组建的addXXXListener方法,将类XXXListener创建的实例对象注册到GUI组件上.

JDK中也提供了大多数事件监听器接口的最简单的实现类,称之为事件适配器(Adapter)类.

用事件适配器来处理事件,可以简化事件监听器编写