目 录

[第一次实验 6](#_Toc343628504)

[题目一：转换字符大小写 6](#_Toc343628505)

[**1、题目** 6](#_Toc343628506)

[**2、源程序** 6](#_Toc343628507)

[**3、实验总结** 6](#_Toc343628508)

[**4、实验截图** 6](#_Toc343628509)

[题目二：求阶乘值 7](#_Toc343628510)

[**1、题目** 7](#_Toc343628511)

[**2、源程序** 7](#_Toc343628512)

[**3、实验总结** 7](#_Toc343628513)

[**4、实验截图** 8](#_Toc343628514)

[第二次实验 8](#_Toc343628515)

[题目一：在main( )方法里完成相应功能 8](#_Toc343628516)

[**1、题目** 8](#_Toc343628517)

[**2、源程序** 8](#_Toc343628518)

[**3、实验总结** 8](#_Toc343628519)

[**4、实验截图** 8](#_Toc343628520)

[题目二：编写Java代码实现一个计数器类Counter 9](#_Toc343628521)

[**1、题目** 9](#_Toc343628522)

[**2、源程序** 9](#_Toc343628523)

[**3、实验总结** 9](#_Toc343628524)

[**4、实验截图** 9](#_Toc343628525)

[题目三：实现矩形类，包括计算周长和面积的方法 10](#_Toc343628526)

[**1、题目** 10](#_Toc343628527)

[**2、源程序** 10](#_Toc343628528)

[**3、实验总结** 10](#_Toc343628529)

[**4、实验截图** 10](#_Toc343628530)

[题目四：使用矩形类计算并显示土地的相关信息 10](#_Toc343628531)

[**1、题目** 10](#_Toc343628532)

[**2、源程序** 10](#_Toc343628533)

[**3、实验总结** 11](#_Toc343628534)

[**4、实验截图** 12](#_Toc343628535)

[第三次实验 12](#_Toc343628536)

[题目一：派生正方形类，定义新方法 12](#_Toc343628537)

[**1、题目** 12](#_Toc343628538)

[**2、源程序** 12](#_Toc343628539)

[**3、实验总结** 13](#_Toc343628540)

[**4、实验截图** 13](#_Toc343628541)

[题目二：定义接口 13](#_Toc343628542)

[**1、题目** 13](#_Toc343628543)

[**2、源程序** 13](#_Toc343628544)

[题目三：改写矩形类使之实现Printable接口 13](#_Toc343628545)

[**1、题目** 13](#_Toc343628546)

[**2、源程序** 13](#_Toc343628547)

[**3、实验总结** 14](#_Toc343628548)

[**4、实验截图** 14](#_Toc343628549)

[题目四：改写正方形类，重载printItMyWay（）方法 14](#_Toc343628550)

[**1、题目** 14](#_Toc343628551)

[**2、源程序** 14](#_Toc343628552)

[**3、实验总结** 15](#_Toc343628553)

[**4、实验截图** 15](#_Toc343628554)

[题目五、六：增加Printable接口的printItMyWay（char）方法并组成一个包 15](#_Toc343628555)

[**1、题目** 15](#_Toc343628556)

[**2、源程序** 16](#_Toc343628557)

[**3、实验总结** 17](#_Toc343628558)

[第四次实验 18](#_Toc343628559)

[题目一：接收正浮点数，生成随机数 18](#_Toc343628560)

[**1、题目** 18](#_Toc343628561)

[**2、源程序** 18](#_Toc343628562)

[**3、实验总结** 19](#_Toc343628563)

[**4、实验截图** 19](#_Toc343628564)

[题目二：使用数组打印10到20的平方表 19](#_Toc343628565)

[**1、题目** 19](#_Toc343628566)

[**2、源程序** 19](#_Toc343628567)

[**3、实验总结** 19](#_Toc343628568)

[**4、实验截图** 19](#_Toc343628569)

[题目三：记录正整数数据并按要求统计 20](#_Toc343628570)

[**1、题目** 20](#_Toc343628571)

[**2、源程序** 20](#_Toc343628572)

[**3、实验总结** 20](#_Toc343628573)

[**4、实验截图** 20](#_Toc343628574)

[题目四：编写图形界面程序，删除字符串中的指定字符 21](#_Toc343628575)

[**1、题目** 21](#_Toc343628576)

[**2、源程序** 21](#_Toc343628577)

[**3、实验总结** 22](#_Toc343628578)

[**4、实验截图** 22](#_Toc343628579)

[题目五：编写Java Application，打印用户输入的命令行 22](#_Toc343628580)

[**1、题目** 22](#_Toc343628581)

[**2、源程序** 22](#_Toc343628582)

[**3、实验总结** 23](#_Toc343628583)

[**4、实验截图** 23](#_Toc343628584)

[第五次实验 23](#_Toc343628585)

[题目一：按要求编写计算器图形界面 23](#_Toc343628586)

[**1、题目** 23](#_Toc343628587)

[**2、源程序** 23](#_Toc343628588)

[**3、实验总结** 24](#_Toc343628589)

[**4、实验截图** 24](#_Toc343628590)

[题目二：模拟手机界面 24](#_Toc343628591)

[**1、题目** 24](#_Toc343628592)

[**2、源程序** 24](#_Toc343628593)

[**3、实验总结** 28](#_Toc343628594)

[**4、实验截图** 28](#_Toc343628595)

[题目三：为上一实例增加事件响应功能 29](#_Toc343628596)

[**1、题目** 29](#_Toc343628597)

[**2、源程序** 29](#_Toc343628598)

[**3、实验总结** 33](#_Toc343628599)

[**4、实验截图** 33](#_Toc343628600)

[题目四：编程实现学生信息管理系统的图形界面 33](#_Toc343628601)

[**1、题目** 33](#_Toc343628602)

[**2、源程序** 33](#_Toc343628603)

[**3、实验总结** 36](#_Toc343628604)

[**4、实验截图** 37](#_Toc343628605)

[题目五：为上例增加事件相应功能 37](#_Toc343628606)

[**1、题目** 37](#_Toc343628607)

[**2、源程序** 37](#_Toc343628608)

[**3、实验总结** 40](#_Toc343628609)

[**4、实验截图** 41](#_Toc343628610)

[题目六：按要求编写一个图形界面的Java Application 41](#_Toc343628611)

[**1、题目** 41](#_Toc343628612)

[**2、源程序** 41](#_Toc343628613)

[**3、实验总结** 42](#_Toc343628614)

[**4、实验截图** 42](#_Toc343628615)

[第六次实验 42](#_Toc343628616)

[题目一至实验三：在图形界面中绘制二维坐标系 42](#_Toc343628617)

[**1、题目** 42](#_Toc343628618)

[**2、源程序** 43](#_Toc343628619)

[**3、实验总结** 46](#_Toc343628620)

[**4、实验截图** 47](#_Toc343628621)

[题目四：跟踪用户的鼠标并显示时刻位置 47](#_Toc343628622)

[**1、题目** 47](#_Toc343628623)

[**2、源程序** 47](#_Toc343628624)

[**3、实验总结** 48](#_Toc343628625)

[**4、实验截图** 48](#_Toc343628626)

[题目五：编写图形界面响应鼠标单击和键盘事件 48](#_Toc343628627)

[**1、题目** 48](#_Toc343628628)

[**2、源程序** 48](#_Toc343628629)

[**3、实验总结** 51](#_Toc343628630)

[**4、实验截图** 51](#_Toc343628631)

[第七次实验 51](#_Toc343628632)

[题目一、二：创建、抛出并修改异常类的事例 51](#_Toc343628633)

[**1、题目** 51](#_Toc343628634)

[**2、源程序** 51](#_Toc343628635)

[**3、实验总结** 52](#_Toc343628636)

[**4、实验截图** 52](#_Toc343628637)

[题目三：使用Swing图形用户界面实现计数器 52](#_Toc343628638)

[**1、题目** 52](#_Toc343628639)

[**2、源程序** 52](#_Toc343628640)

[**3、实验总结** 53](#_Toc343628641)

[**4、实验截图** 53](#_Toc343628642)

[题目四、五：使用JApplet实现某一门课程的分数录入系统 54](#_Toc343628643)

[**1、题目** 54](#_Toc343628644)

[**2、源程序** 54](#_Toc343628645)

[**3、实验总结** 59](#_Toc343628646)

[**4、实验截图** 60](#_Toc343628647)

[第八次实验 60](#_Toc343628648)

[题目一：接受用户输入的文件名并检查文件信息 60](#_Toc343628649)

[**1、题目** 60](#_Toc343628650)

[**2、源程序** 60](#_Toc343628651)

[**3、实验总结** 60](#_Toc343628652)

[**4、实验截图** 60](#_Toc343628653)

[题目二：以二进制保存用户输入的数据 60](#_Toc343628654)

[**1、题目** 61](#_Toc343628655)

[**2、源程序** 61](#_Toc343628656)

[**3、实验总结** 63](#_Toc343628657)

[题目三：在用户输入文件名的文档中保存用户输入的字符串 63](#_Toc343628658)

[**1、题目** 63](#_Toc343628659)

[**2、源程序** 64](#_Toc343628660)

[**3、实验总结** 64](#_Toc343628661)

[题目四：编写将两个文件中的内容合并成一个新文件的程序 64](#_Toc343628662)

[**1、题目** 64](#_Toc343628663)

[**2、源程序** 64](#_Toc343628664)

[**3、实验总结** 65](#_Toc343628665)

[第九次实验 65](#_Toc343628666)

[题目一：接受用户输入的主机名并打印IP地址 65](#_Toc343628667)

[**1、题目** 65](#_Toc343628668)

[**2、源程序** 65](#_Toc343628669)

[**3、实验总结** 65](#_Toc343628670)

[**4、实验截图** 65](#_Toc343628671)

[题目二：接受用户输入的网页名并打印电子邮件地址 65](#_Toc343628672)

[**1、题目** 65](#_Toc343628673)

[**2、源程序** 66](#_Toc343628675)

[**3、实验总结** 66](#_Toc343628676)

[题目三：利用流式Socket通信机制模拟手机的短信息服务 67](#_Toc343628677)

[**1、题目** 67](#_Toc343628678)

[**2、源程序** 67](#_Toc343628680)

[**3、实验总结** 68](#_Toc343628681)

第一次实验

**题目一：转换字符大小写**

**1、题目**

编写字符界面的java application，将用户输入的小写字符转化成大写字符打印在屏幕上，若用户输入的不是小写字符则打印信息说明无法完成的操作。

**2、源程序**

**import** java.util.Scanner;

**public** **class** test11 {

**public** **static** **void** main(String[] args) {

System.*out*.println("请输入一个小写字母：");

Scanner in=**new** Scanner(System.*in*);//Scanner对象in接收输入字符

**char** c = in.next().charAt(0);

// if(c>='A'&&c<='Z'){

// c+=32;

// }//将大写字母转换成小写字母并输出

**If** (c >= 'a' && c <= 'z') {

c -= 32;

System.*out*.println("小写字母:"+(**char**)(c+32)+"转换为大写字母:"+c);

}// 将小写字母转换成大写字母并输出

**else** {

System.*out*.println("输入错误，请重新输入");

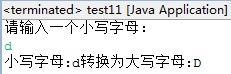
}

}

}

**3、实验总结**

此程序为顺序结构，程序用到了java.util.Scanner包中的类Scanner，此程序没有遇到错误。

 **4、实验截图**

**题目二：求阶乘值**

**1、题目**

编写一个Applet程序，从键盘输入一个整数，求出其阶乘值。

**2、源程序**

**import** java.applet.\*;

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

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

**public** **class** test12 **extends** Applet **implements** ActionListener {

Label prompt;//定义一个标签对象

TextField input;//定义一个文本框对象

**double** n = 0;

**double** temp ;

**public** **void** init() {

prompt = **new** Label("请输入一个整数");//用标签提示输入信息

input = **new** TextField(3);//设置文本框大小

add(prompt);

add(input);

input.addActionListener(**this**);//把文本对象注册给动作事件监听者

}

**double** factorial(**double** n) {

**if** (0 == n || 1 == n)

**return** 1;

**else**

**return** factorial(n - 1) \* (n);

}//用递归实现n的阶乘

**public** **void** paint(Graphics g) {

g.drawString("你输入了数据：" + n, 50,50);//确认用户输入数据

g.drawString("其阶乘是:" + temp, 50,75);//显示用户欲计算的阶乘

}

**public** **void** actionPerformed(ActionEvent e) {

n = Double.*parseDouble*(input.getText());//取得用户输入数据并将其转化为双精度数据再将其赋值给n

temp = factorial(n);

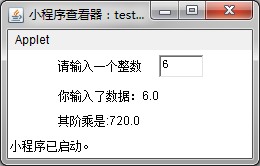
repaint();

}

}

**3、实验总结**

此程序为Applet程序采用顺序结构，此程序用到采用图形界面的方法引用用了java.applet.\*，java.awt.\*以及java.awt.\*包中的所有类，类test12继承了Applet类并实现了其init()接口和actionPerformed()接口并且用到了paint()函数和repaint()函数。

 **4、实验截图**

第二次实验

**题目一：在main( )方法里完成相应功能**

**1、题目**

阅读程序，在main()方法里添加语句完成如下的功能：

（1）创建一个MyValue类的对象myValue

（2）为myValue对象中的value域赋值10

（3）使用getValue()方法获得myValue对象中的数据并将它们打印在屏幕上

**2、源程序**

**public** **class** test21 {

**public** **static** **void** main(String args[]) {

MyValue myvalue=**new** MyValue();

myvalue.setValue(10);

System.*out*.println("myvalued的值为" +myvalue. getValue());

}

}

**class** MyValue {

**private** **int** value;

**public** **void** setValue(**int** x) {

value = x;

}

**public** **int** getValue() {

**return** value;

}

}

**3、实验总结**

此程序为顺序结构，程序创建了一个MyValue类的对象myValue并且为myValue对象中的value域赋值10。在主函数中使用getValue()方法获得myValue对象中的数据并将它们打印在屏幕上。此程序没有遇到错误。

**** **4、实验截图**

**题目二：编写Java代码实现一个计数器类Counter**

**1、题目**

编写Java代码实现一个计数器类Counter, 其中包含：

域counterValue 用来保存计数器的当前数值；

方法increment(), 计数器加一；

方法decrement(), 计数器减一；

方法reset(), 计数器清零。

**2、源程序**

**public** **class** test22 {

**private** **int** counterValue =6;//设计数器初始值为6

**public** **static** **void** main(String []args){

test22 counter=**new** test22();//创建Counter对象 counter

System.*out*.println(counter.increment());//调用increment()函数 System.*out*.println(counter.decrement());//调用decrement()函数 System.*out*.println(counter.reset());

}//调用reset()函数实现计数器清零

**public** **int** increment(){

counterValue++;

**return** counterValue;

}

**public** **int** decrement(){

counterValue--;

**return** counterValue;

}

**public** **int** reset(){

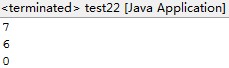
**return** counterValue=0;

}

}

**3、实验总结**

此程序采用顺序结构， 其中包含了域counterValue 用来保存计数器的当前数值，方法increment()、decrement()、reset()。在主函数中分别调用方法increment(), 实现方法decrement()、方法reset()。

 **4、实验截图**

**题目三：实现矩形类，包括计算周长和面积的方法**

**1、题目**

编程实现矩形类，其中包括计算矩形周长和面积的方法。

**2、源程序**

**public** **class** test23 {

**private double** length=0.0;

**private** **double** width =0.0;

test23 (**double** x,**double** y){

length=x;

width=y;

}

**public** **static** **void** main(String []args){

test23 rectange=**new** test23(3,5); //创建矩形对象test23并赋值为3、5.

System.*out*.println(rectangle .CZhouchang());//矩形对象test23调用求周长的方法，并将结果打印出来

System.*out*.println(rectange .Area());//矩形对象test23调用求面积的方法，并将结果打印出来

}

**public** **double** Girth(){ //求矩形周长

**return** 2\* (length+width );

}

**public** **double** Area(){ //求矩形面积

**return** length\*width ;

}

}

**3、实验总结**

此程序采用顺序结构，其中包含了域length和length初值均为0，方法CZhouchang ()用来计算矩形周长，方法Area(),用来计算矩形面积。在主函数中分别调用方法CZhouchang(), 计算矩形周长并打印出，方法Area(),计算矩形面积并打印出。

 **4、实验截图**

**题目四：使用矩形类计算并显示土地的相关信息**

**1、题目**

编程实现矩形类，其中包括计算矩形周长和面积的方法。

**2、源程序**

**import** java.applet.\*;

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

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

**public** **class** test24 **extends** Applet **implements** ActionListener {

**private** **double** length;//土地长度

**private** **double** width;//土地宽度

**double** temp ;

Label prompt ;//建立两个标签对象用来实现信息的输入

Label prompt1;

TextField input;

TextField input1;

**public** **void** init() {

prompt = **new** Label("请输入土地的长度");

prompt1 = **new** Label("请输入土地的宽度");

input = **new** TextField(2);

input1 = **new** TextField(2);

add(prompt);

add(input);

add(prompt1);

add(input1);

input.addActionListener(**this**);

input1.addActionListener(**this**);

}

**public** **void** paint(Graphics g) {

g.drawString("长：" + length, 25, 80); //输出用户输入的长度信息

g.drawString("宽：" + width, 25, 110);//输出用户输入的宽度信息

g.drawString("面积：" + temp, 25, 140);//输出土地面积信息

}

**public** **void** actionPerformed(ActionEvent e) {

length = Double.*parseDouble*(input.getText());//获取用户输入的长度信息

width = Double.*parseDouble*(input1.getText());//获取用户输入的宽度信息

temp = length \* width;

repaint();//重新调用paint（）方法。

}

}

**3、实验总结**

此程序为Applet程序应用了图形界面的方法，采用顺序结构，此程序引用了java.applet.\*，java.awt.\*以及java.awt.event.\*中所有类。此程序用到了init()方法用来实现信息的输入，和actionPerformed(ActionEvent e)方法来获取用户输入信息，以及paint(Graphics g)方法用来实现信息的输出。

 **4、实验截图**

第三次实验

**题目一：派生正方形类，定义新方法**

**1、题目**

为上一实验中的矩形类派生一个子类：正方形类。若正方形类的操作同样是求周长和面积，则这个字类除了从父类继承来的方法之外，还需要定义哪些方法。列出正方形类的所有域和方法。编程检查、运行所编写的正方形类。

**2、源程序**

**class** Rectange {

**private** **double** length=0.0;

**private** **double** width =0.0;

Rectange(){}

Rectange (**double** x,**double** y){

length=x;

width=y;

}

**public** **static** **void** main(String []args){

Rectange rectange=**new** Rectange(66,88);

test31 square=**new** test31(6,8);

System.*out*.println(rectange.Girth());

System.*out*.println(rectange.Area());

System.*out*.println(square.Girth());

System.*out*.println(square.Area());

}

**public** **double** Girth(){//求矩形周长

**return** 2\* (length+width );

}

**public** **double** Area(){//求矩形面积

**return** length\*width ;

}

}

**class** test31 **extends** Rectange{//正方形类

**private** **double** length=0.0;

test31(**double** x,**double** y){//super(x,y);

length=x;

}

**public** **double** Girth(){//求正方形周长

**return** 4\* length;

}

**public** **double** Area(){//求正方形面积

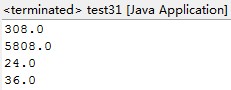
**return** length\*length ;

}

}

**3、实验总结**

此程序为顺序结构，程序创建了一个Rectange类对象 rectange和子类Square 对象test31并且分别赋值为(66,88)和(6,8)在主函数中使用矩形对象rectange和正方形对象 test31分别调用自己的周长和面积的方法并打印出其结果。

**** **4、实验截图**

**题目二：定义接口**

**1、题目**

定义接口Printable, 其中包括方法printItMyWay()且没有形参，返回值为空。

**2、源程序**

**interface** Printable {

**public** **abstract** **void** printItMyWay();

}

**题目三：改写矩形类使之实现Printable接口**

**1、题目**

改写矩形类，使之实现Printable接口，用printItMyWay()方法将矩形的相关信息（长，宽，周长，面积）打印在屏幕上。

**2、源程序**

**class** Rectange **implements** Printable {

**private** **double** length = 0.0;

**private** **double** width = 0.0;

**public** **static** **void** main(String[] args) {

Rectange rectange = **new** Rectange(3, 5);

rectange.printItMyWay();

}

Rectange() {}

Rectange(**double** x, **double** y) {

length= x;

width = y;

}

**public** **void** printItMyWay() {

System.out.println("长:" + length);

System.out.println("宽:" + width);

System.out.println("周长:" + 2 \* (length + width));

System.out.println("面积:" + length \* width);

}

}

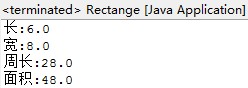
**interface** Printable {

**public** **abstract** **void** printItMyWay();

}

**3、实验总结**

此程序采用顺序结构，改写了矩形类，使之实现了Printable接口，在主函数中定义了Rectange对象 rectange并赋值为（6，8），然后用Rectange对象 rectange调用printItMyWay()方法将矩形的相关信息（长，宽，周长，面积）打印在了屏幕上。

 **4、实验截图**

**题目四：改写正方形类，重载printItMyWay（）方法**

**1、题目**

改写正方形类，重载printItMyWay()方法，将正方形的边长、周长，面积打印。

**2、源程序**

**class** Rectange **implements** Printable {

**private** **double** length=0.0;

**private** **double** width =0.0;

Rectange(){}

Rectange (**double** x,**double** y){

length=x;

width=y;

}

**public** **static** **void** main(String []args){

Rectange rectange=**new** Rectange(6,8);

Square square=**new** Square(3,6);

square.printItMyWay();

}

**public** **void** printItMyWay() {

System.*out*.println("长方形长:" + length);

System.*out*.println("长方形宽:" + width);

System.*out*.println("长方形周长:" + 2 \* (length + width));

System.*out*.println("长方形面积:" + length \* width);

}

}

**class** Square **extends** Rectange{ //正方形类

**private** **double** length=0.0;

Square(**double** x,**double** y){

length=x;

}

**public** **void** printItMyWay() {

System.*out*.println("正方形边长:" + length);

System.*out*.println("正方形周长:" + 4\* length);

System.*out*.println("正方形面积:" + length \* length);

}

}

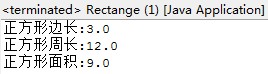
**interface** Printable {

**public** **abstract** **void** printItMyWay();

}

**3、实验总结**

此程序采用顺序结构，改写了正方形类，重载了printItMyWay()方法，在主函数中定义Square类对象并赋值为(3,6)， Square类对象 square调用printItMyWay()方法将正方形的边长、周长，面积打印在了屏幕上。

 **4、实验截图**

**题目五、六：增加Printable接口的printItMyWay（char）方法并组成一个包**

**1、题目**

在Printable接口中增加一个新的printItMyWay(char)方法，这个方法有一个字符类型的形参，返回值为空。其功能是利用给出的字符打印，例如若给定的字符为#，一个长为5，宽为3的矩形的屏幕打印结果为：

#####

#####

#####

在矩形类和正方形类中实现该方法，将矩形和正方形组成一个包MyFigures。

**2、源程序**

**package** MyFigures;

**import** java.util.Scanner;

**class** Rectange **implements** Printable {

**private** **double** length = 0.0;

**private** **double** width = 0.0;

**public** **static** **void** main(String[] args) {

Rectange rectange = **new** Rectange(4, 5);

Square square = **new** Square(5, 5);

rectange.printItMyWay();

square.printItMyWay();

rectange.printItMyway('c');

square.printItMyway('c');

}

Rectange() {}

Rectange(**double** x, **double** y) {

length = x;

width = y;

}

**public** **void** printItMyWay() {

System.*out*.println("矩形的长为" + length );

System.*out*.println("矩形的宽为" + width);

System.*out*.println("矩形的周长为" + 2 \* (length + width));

System.*out*.println("矩形面积为" + length \* width);

}

**public** **void** printItMyway(**char** c) {

System.*out*.println("请输入一个字符");

Scanner in = **new** Scanner(System.*in*);// Scanner对象in接收输入的字符

c = in.next().charAt(0);

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

**for** (**int** j = 0; j < width; j++) {

System.*out*.print(c);

}

System.*out*.println();

}

}

**public** **double** Girth() {// 求矩形周长

**return** 2 \* (length + width);

}

**public** **double** Area() {// 求矩形面积

**return** length \* width;

}

}

**interface** Printable {

**public** **abstract** **void** printItMyWay();

**public** **abstract** **void** printItMyway(**char** c);

}

**package** MyFigures;

**import** java.util.Scanner;

**class** Square **extends** Rectange {// 正方形类

**private** **double** length = 0.0;

Square(**double** x, **double** y) {

// super(x,y);

length = x;

}

**public** **double** Girth() {// 求矩形周长

**return** 4 \* length ;

}

**public** **double** Area() {// 求矩形面积

**return** length \* length;

}

**public** **void** printItMyWay() {

System.*out*.println("矩形的边长为" + length );

System.*out*.println("正方形的周长为" + 4\* length );

System.*out*.println("正方形的面积为" + length \* length);

}

**public** **void** printItMyway(**char** c) {

System.*out*.println("请输入一个字符");

Scanner in = **new** Scanner(System.*in*);// Scanner对象in接收输入的字符

c = in.next().charAt(0);

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

**for** (**int** j = 0; j < length; j++) {

System.*out*.print(c);

}

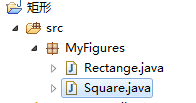
System.*out*.println();

}

}

}

**3、实验总结**

此程序为顺序结构，首先将矩形类和正方形类组成一个包MyFigures.如图所示

然后在Printable接口中增加了一个新的printItMyWay(char)方法，这个方法有一个字符类型的形参，返回值为空。其功能是利用给出的字符打印，例如若给定的字符为#，一个长为8，宽为6的矩形的屏幕打印结果为：

########

########

########

########

########

########

第四次实验

**题目一：接收正浮点数，生成随机数**

**1、题目**

接收用户输入的一个正浮点数，随机生成一个不大于该数的正整数，在屏幕打印输出。

**2、源程序**

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**public** **class** test41 {

**public** **static** **void** main(String[] args) {

String s;

**float** f = 0;

**int** r;

System.*out*.println("请输入正浮点数:");

**try** {

BufferedReader y = **new** BufferedReader(**new** InputStreamReader(

System.*in*));

s = y.readLine();//将用户输入的字符串赋给s。

f = Float.*parseFloat*(s);//将字符串s转换成浮点型数据赋值给f。

**while** (f <= 0) {

System.*out*.println("输入错误，请输入一个至少一位小数的正浮点数：");

BufferedReader x = **new** BufferedReader(**new** InputStreamReader(

System.*in*));

s = x.readLine();

f = Float.*parseFloat*(s);

}

} **catch** (IOException e) {

}

r = (**int**) (Math.*random*() \* f);

System.*out*.println("产生的随机数为" + r);

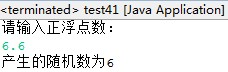
}

}

**3、实验总结**

此程序为顺序结构，程序用到了io包中的BufferedReader, IOExcep InputStreamReader类,此程序没有遇到错误。

**4、实验截图**

****

**题目二：使用数组打印10到20的平方表**

**1、题目**

使用数组实现一个10到20的平方表，打印输出这个平方表。

**2、源程序**

**public** **class** test42 {

**public** **static** **void** main(String[] args) {

**int** A[] = **new** **int**[20];//新建一个整形数组A，数组大小为20

**int** m = 0;

**for** (**int** i = 11; i <= 20; i++) {

A[m] = i \* i;

System.*out*.println(A[m++]);//每打印1个数组元素换行

**if** (m % 5 == 0)

System.*out*.println();//每打印5个平方和换行

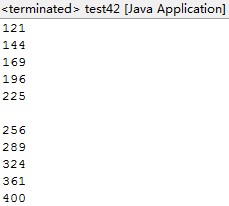
}

}

}

**3、实验总结**

此程序采用顺序结构，此程序没有遇到错误。

 **4、实验截图**

**题目三：记录正整数数据并按要求统计**

**1、题目**

编程记录用户输入的一组正整数数据，-1表示输入结束，求这些数据的平均值，并统计大于平均值的数据个数。

**2、源程序**

**import** java.util.Scanner;

**public** **class** test43 {

**public** **static** **void** main(String[] args) {

**int**[] a = **new** **int**[20];

**int** i = 0, t, sum = 0;

**int** avg;

**int** count = 0;

System.*out*.println("请输入一组正整数数据，以-1结束：");

Scanner s = **new** Scanner(System.*in*);

**while**(**true**)

{

t = s.nextInt();

**if** (t == -1) **break**;

a[i++] = t;

sum += t;

}

avg = sum / i;

**for** (**int** j = 0; j < i; j++)

**if** (a[j] >avg)

count++;

System.*out*.println("平均值:" + avg);

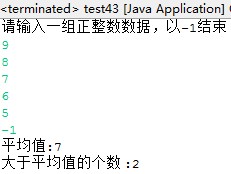
System.*out*.println("大于平均值的个数 :" + count);

}

}

**3、实验总结**

此程序采用顺序结构，此程序没有遇到错误。

 **4、实验截图**

**题目四：编写图形界面程序，删除字符串中的指定字符**

**1、题目**

编写图形界面的程序，接收用户输入的一个字符串和一个字符，将字符串中的所有该字符删除，打印输出新生成的结果字符串。

**2、源程序**

**import** java.applet.\*;

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

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

**public** **class** test44 **extends** Applet **implements** ActionListener {

String originalString,modifieldString;

**int** delChar;

Label orgStringLb1=**new** Label("原始字符串");

Label delCharLb1=**new** Label("需去除字符");

TextField orgStringTfd=**new** TextField(20);

TextField delCharTfd=**new** TextField(1);

Button modifyBtn =**new** Button("删除字符");

**public** **void** init(){

add(orgStringLb1);

add(orgStringTfd);

add(delCharLb1);

add(delCharTfd);

add(modifyBtn);

orgStringTfd.setText("");

delCharTfd.setText("");

originalString="";

modifieldString="";

modifyBtn.addActionListener(**this**);

}

**public** **void** paint(Graphics g){

g.drawString(modifieldString,10,100);

}

**public** **void** actionPerformed(ActionEvent e){

**if**(e.getSource()==modifyBtn){

originalString=orgStringTfd.getText();

delChar=(**int**)(delCharTfd.getText().charAt(0));

modifieldString= "";

**int** i=0,j=0;

**while** ((j=originalString.indexOf(delChar,i))!=-1){

System.*out*.println(i +","+j+","+(j-1));

System.*out*.println(originalString);

modifieldString=modifieldString+originalString.substring(i,j);

System.*out*.println(modifieldString);

i=j+1;

}

modifieldString=modifieldString+originalString.substring(i,originalString.length());

repaint();

}

**else**{

showStatus("事件"+e.toString()+"为定义处理操作");

orgStringTfd.setText("");

delCharTfd.setText("");

originalString="";

modifieldString="";

}

}

}

**3、实验总结**

此程序采用顺序结构，此程序没有遇到错误。

 **4、实验截图**

**题目五：编写Java Application，打印用户输入的命令行**

**1、题目**

编写Java Application，将用户输入的命令行参数一个一行打印出来。

**2、源程序**

**public** **class** test45 {

**public** **static** **void** main(String[] args) {

**if** (args.length == 0) {

System.*out*.println("没有参数");

}

**else** {

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

System.*out*.println("参数" + (i + 1) + "为：" + args[i]);

}

}

}

}

**3、实验总结**

此程序采用顺序结构，此程序没有遇到错误。

 **4、实验截图**

第五次实验

**题目一：按要求编写计算器图形界面**

**1、题目**

为第2次实验中编写的计数器类Counter编写图形界面，其中包括三个按钮、一个标签。按钮分别是计数器加一、计数器减一和计数器清零，标签显示当前的计数器数值。

**2、源程序**

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

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

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

**public** **class** test51 **extends** JFrame **implements** ActionListener {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**int** counterValue = 10;

JButton jb = **new** JButton("初始值为10");

JButton jb0 = **new** JButton(" ");

JButton jb1 = **new** JButton("加一");

JButton jb2 = **new** JButton("减一");

JButton jb3 = **new** JButton("清零");

**public** **void** init() {

BorderLayout b = **new** BorderLayout(10,10);

setLayout(b);

add(jb,BorderLayout.*NORTH*);

add(jb0,BorderLayout.*CENTER*);

add(jb1,BorderLayout.*WEST*);

add(jb2,BorderLayout.*EAST*);

add(jb3,BorderLayout.*SOUTH*);

jb1.addActionListener(**this**);

jb2.addActionListener(**this**);

jb3.addActionListener(**this**);

setVisible(**true**);

setSize(100,60);

setLocation(20,20);

}

**public** **static** **void** main(String args[]){

**new** test51();

}

**public** test51 (){

init();

}

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == jb1) {

counterValue++;

jb0.setText(" " + counterValue);// 显示计数器加一后结果

} **else** **if** (e.getSource() == jb2) {

counterValue--;

jb0.setText(" " + counterValue);// 显示计数器减一后结果

} **else** {

counterValue = 0;

jb0.setText(" " + counterValue);// 显示计数器清零后结果

}

}

}

**3、实验总结**

此程序为顺序结构，为JApplet图形界面程序，此程序用到了import java.awt.\*包中的所有类和java.awt.event.ActionEvent类以及javax.swing.\*包中所有类。

**** **4、实验截图**

**题目二：模拟手机界面**

**1、题目**

编程实现一个模拟手机的图形用户界面，包括一个显示区和一个键盘区，显示区用来显示数字和信息，键盘区包括0—9十个数字键和“发送”“清除”“关机”三个命令键。考虑应该使用什么样的图形用户界面布局策略。

**2、源程序**

**import** java.awt.BorderLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;

**public** **class** test52 **extends** JFrame {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**private** JTextField txtView;

**public** **static** **void** main(String args[]) {

**try** {

test52 moble = **new** test52();

moble.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

**public** test52() {

setTitle("Moble");

setBounds(100, 100, 200, 300);

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

**final** JPanel panel = **new** JPanel();

getContentPane().add(panel, BorderLayout.*CENTER*);

txtView = **new** JTextField();

panel.setLayout(**null**);

panel.add(txtView);

**final** JButton btn1 = **new** JButton();

btn1.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("1"));

}

}

);

btn1.setText("1");

btn1.setBounds(0, 73, 59, 23);

panel.add(btn1);

**final** JButton btn2 = **new** JButton();

btn2.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("2"));

}

}

);

btn2.setText("2");

btn2.setBounds(68, 73, 59, 23);

panel.add(btn2);

**final** JButton btn3 = **new** JButton();

btn3.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("3"));

}

}

);

btn3.setText("3");

btn3.setBounds(133, 73, 59, 23);

panel.add(btn3);

**final** JButton btn4 = **new** JButton();

btn4.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("4"));

}

}

);

btn4.setText("4");

btn4.setBounds(0, 102, 59, 23);

panel.add(btn4);

**final** JButton btn5 = **new** JButton();

btn5.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("5"));

}

}

);

btn5.setText("5");

btn5.setBounds(65, 102, 59, 23);

panel.add(btn5);

**final** JButton btn6 = **new** JButton();

btn6.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("6"));

}

}

);

btn6.setText("6");

btn6.setBounds(133, 102, 59, 23);

panel.add(btn6);

**final** JButton btn7 = **new** JButton();

btn7.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("7"));

}

}

);

btn7.setText("7");

btn7.setBounds(0, 131, 59, 23);

panel.add(btn7);

**final** JButton btn8 = **new** JButton();

btn8.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("8"));

}

}

);

btn8.setText("8");

btn8.setBounds(65, 131, 59, 23);

panel.add(btn8);

**final** JButton btn9 = **new** JButton();

btn9.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("9"));

}

}

);

btn9.setText("9");

btn9.setBounds(133, 131, 59, 23);

panel.add(btn9);

**final** JButton btn0 = **new** JButton();

btn0.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText(txtView.getText().concat("0"));

}

}

);

btn0.setText("0");

btn0.setBounds(68, 160, 57, 23);

panel.add(btn0);

txtView = **new** JTextField();

txtView.setBounds(59, 24, 90, 21);

panel.add(txtView);

**final** JButton btnSend = **new** JButton();

btnSend.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

**if**(txtView.getText().indexOf("calling...")==-1){

txtView.setText(txtView.getText().concat("calling..."));

}

**else**{

**return**;

}

}

}

);

btnSend.setText("Send");

btnSend.setBounds(40, 189, 127, 23);

panel.add(btnSend);

**final** JButton btnclear = **new** JButton();

btnclear.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

txtView.setText("");

}

}

);

btnclear.setText("Clear");

btnclear.setBounds(40, 212, 127, 23);

panel.add(btnclear);

**final** JButton btnOff = **new** JButton();

btnOff.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

dispose();

}

}

);

btnOff.setText("Off");

btnOff.setBounds(40, 232, 127, 23);

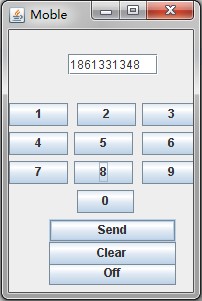
panel.add(btnOff);

}

}

**3、实验总结**

此程序采用顺序结构，程序中用到了java.awt.BorderLayout，java.awt.event.ActionEvent，java.awt.event.ActionListener、javax.swing.JButton、javax.swing.JFrame，javax.swing.JPanel、javax.swing.JTextField这些类。

 **4、实验截图**

**题目三：为上一实例增加事件响应功能**

**1、题目**

为上题所编写的界面增加事件响应功能：当用户按数字键时，相应的数字回显在显示区上，当用户按清除键时，显示区的数字清空，当用户按发送键时，显示区出现当前的号码和“拨号中……”的信息，当用户按关机键时结束程序。

**2、源程序**

**import** java.applet.\*;

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

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

**public** **class** test53 **extends** Applet **implements** ActionListener {

**int** flag = 0;

StringBuffer Str = **new** StringBuffer();

TextField show = **new** TextField("中国移动通信");

Panel p1 = **new** Panel();

Button b0 = **new** Button("发送");

Button b1 = **new** Button("清除");

Button b2 = **new** Button("关机");

Panel p2 = **new** Panel();

Button b3 = **new** Button("0");

Button b4 = **new** Button("1");

Button b5 = **new** Button("2");

Button b6 = **new** Button("3");

Button b7 = **new** Button("4");

Button b8 = **new** Button("5");

Button b9 = **new** Button("6");

Button b10 = **new** Button("7");

Button b11 = **new** Button("8");

Button b12 = **new** Button("9");

Button b13 = **new** Button("\*");

Button b14 = **new** Button("#");

**public** **void** init() {

show.setEditable(**false**);

setLayout(**new** BorderLayout());

add(show, BorderLayout.*NORTH*);

show.setBackground(Color.*GRAY*);

add(p1, BorderLayout.*CENTER*);

add(p2, BorderLayout.*SOUTH*);

p1.setLayout(**new** GridLayout(1, 3));

p1.add(b0);

p1.add(b1);

p1.add(b2);

p2.setLayout(**new** GridLayout(3, 4));

p2.add(b4);

p2.add(b5);

p2.add(b6);

p2.add(b14);

p2.add(b7);

p2.add(b8);

p2.add(b9);

p2.add(b13);

p2.add(b10);

p2.add(b11);

p2.add(b12);

p2.add(b3);

b0.addActionListener(**this**);

b1.addActionListener(**this**);

b2.addActionListener(**this**);

b3.addActionListener(**this**);

b4.addActionListener(**this**);

b5.addActionListener(**this**);

b6.addActionListener(**this**);

b7.addActionListener(**this**);

b8.addActionListener(**this**);

b9.addActionListener(**this**);

b10.addActionListener(**this**);

b11.addActionListener(**this**);

b12.addActionListener(**this**);

b13.addActionListener(**this**);

b14.addActionListener(**this**);

}

**public** **void** actionPerformed(**final** ActionEvent e) {

**if** (e.getSource() == b0) {

**if** (Str.toString().equals("10086")) {

flag = 0;

show.setText("智能服务拨号中……");

}

**else** {

flag = 0;

show.setText("拨号中……");

Str.delete(0, Str.length());

}

}

**else** **if** (e.getSource() == b1) {

flag = 0;

show.setText("中国移动");

Str.delete(0, Str.length());

}

**else** **if**(e.getSource() == b2) {

flag = 0;

show.setText("再见");

Str.delete(0, Str.length());

**try**{

Thread.*sleep*(1000);//程序休眠1000ms后再结束该程序

}**catch**(InterruptedException e1){

e1.printStackTrace();

}

System.*exit*(0);

}

**else** **if**(e.getSource() == b3) {

**if**(flag==1){

flag=0;

show.setText("人工服务");

Str.delete(0, Str.length());

}

**else**{

Str.append(0);

show.setText(Str.toString());

}

}

**else** **if**(e.getSource() == b4){

**if**(flag==1){

show.setText("消费查询");

Str.delete(0, Str.length());

flag=0;

}

**else**{

Str.append(b4.getActionCommand());

show.setText(Str.toString());

}

}

**else** **if**(e.getSource() == b5){

**if**(flag==1){

show.setText("GPRS流量查询");

Str.delete(0, Str.length());

flag=0;

}

**else**{

Str.append(b5.getActionCommand());

show.setText(Str.toString());

}

}

**else** **if**(e.getSource() == b6){

**if**(flag==1){

show.setText("套餐查询");

Str.delete(0, Str.length());

flag=0;

}

**else**{

Str.append(b6.getActionCommand());

show.setText(Str.toString());

}

}

**else** **if**(e.getSource() == b7){

flag=0;

Str.append(b7.getActionCommand());

show.setText(Str.toString());

}

**else** **if**(e.getSource() == b8){

flag=0;

Str.append(b8.getActionCommand());

show.setText(Str.toString());

}

**else** **if**(e.getSource() == b9){

flag=0;

Str.append(b9.getActionCommand());

show.setText(Str.toString());

}

**else** **if**(e.getSource() == b10){

flag=0;

Str.append(b10.getActionCommand());

show.setText(Str.toString());

}

**else** **if**(e.getSource() == b11){

flag=0;

Str.append(b11.getActionCommand());

show.setText(Str.toString());

}

**else** **if**(e.getSource() == b12){

flag=0;

Str.append(b12.getActionCommand());

show.setText(Str.toString());

}

**else** **if**(e.getSource() == b13){

flag=0;

Str.append(b13.getActionCommand());

show.setText(Str.toString());

}

**else** **if**(e.getSource() == b14){

flag=0;

Str.append(b14.getActionCommand());

show.setText(Str.toString());

}

}

}

**3、实验总结**

此程序采用顺序结构，为上题所编写的界面增加了事件响应功能：当用户按数字键时，相应的数字回显在显示区上，当用户按清除键时，显示区的数字清空，当用户按发送键时，显示区出现当前的号码和“拨号中……”的信息，当用户按关机键时出现再见，经过1000ms后结束程序。

**4、实验截图**

|  |  |
| --- | --- |
| 拨号界面1 | 拨号界面2 |

**题目四：编程实现学生信息管理系统的图形界面**

**1、题目**

编程实现一个学生信息管理系统的图形界面。需要录入的信息包括学生学号、姓名、性别、出生年、月、日、年级以及是否联合培养。根据这些信息的性质选择使用不同的图形组件来设计界面。

**2、源程序**

**import** java.awt.BorderLayout;

**import** java.awt.Checkbox;

**import** java.awt.CheckboxGroup;

**import** java.awt.Container;

**import** java.awt.Font;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ItemEvent;

**import** java.awt.event.ItemListener;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;//录入的信息包括学号、姓名、性别、出生日期、是否联合培养等

**public** **class** test54 **extends** JFrame **implements** ActionListener {

**public** test54() {//界面显示问题

**super**("学生信息图形界面");

setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);

setSize(320, 300);

setVisible(**true**);

Container content = getContentPane();

JPanel jp1 = **new** JPanel(); //创建一个面板

JLabel label = **new** JLabel("欢迎使用学生信息管理系统");

label.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 20));

jp1.add(label); //将标签添加到面板jp1中

JPanel jp2 = **new** JPanel();//创建一个面板，用于摆放提示各种功能的标签

JLabel label1 = **new** JLabel("请输入学生的详细资料：");//输入提示输入的语言：

label1.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

JLabel label2 = **new** JLabel("学 生 学 号："); //提示各种输入内容的标签

label2.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

JTextField field1=**new** JTextField(10); //10为学号长度，可自行设定

JLabel label3 = **new** JLabel("学 生 姓 名："); //提示输入学生姓名的标签

label3.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18)); //设置字体

JTextField field2=**new** JTextField(10); //10为姓名长度，可自行设定

JLabel label4 = **new** JLabel("学 生 性 别： "); //提示输入学生性别标签

label4.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置显示字体

CheckboxGroup group1 = **new** CheckboxGroup();

//创建一个复选框用于选择某一功能

Checkbox check11 = **new** Checkbox("男",**true**,group1); //添加选项复选框中

check11.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

check11.addItemListener(**new** CheckHandler1(1));

//给相应复选框添加事件监听器

Checkbox check12 = **new** Checkbox("女",**false**,group1); //添加选项复选框中

check12.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

check12.addItemListener(**new** CheckHandler1(2));

//给相应复选框添加事件监听器

JLabel label5 = **new** JLabel("出 生 日 期："); //提示输入出生日期的标签

label5.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

JTextField field3=**new** JTextField("2000",3); //3为姓名长度，可自行设定

JLabel label51 = **new** JLabel("年");

label51.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

JTextField field4=**new** JTextField("01",2); //2为姓名长度，可自行设定

JLabel label52 = **new** JLabel("月");

label52.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

JTextField field5=**new** JTextField("01",2); //2为姓名长度，可自行设定

JLabel label53 = **new** JLabel("日");

label53.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

JLabel label6 = **new** JLabel("联合培养：");//提示选择联合培养与否的标签

label6.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18)); //设置字体

CheckboxGroup group2 = **new** CheckboxGroup();

//创建一个复选框用于选择某一功能

Checkbox check21 = **new** Checkbox("是",**true**,group2);//添加选项到复选框中

check21.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18)); //设置字体

check21.addItemListener(**new** CheckHandler2(1));

//给相应复选框添加事件监听器

Checkbox check22 = **new** Checkbox("否",**false**,group2);//添加选项复选框中

check22.setFont(**new** Font("楷体\_GB2312", Font.*BOLD*, 18));//设置字体

check22.addItemListener(**new** CheckHandler2(2));

jp2.add(label1);

jp2.add(label2);

jp2.add(field1);

jp2.add(label3);

jp2.add(field2);

jp2.add(label4);

jp2.add(check11);

jp2.add(check12);

jp2.add(label5);

jp2.add(field3);

jp2.add(label51);

jp2.add(field4);

jp2.add(label52);

jp2.add(field5);

jp2.add(label53);

jp2.add(label6);

jp2.add(check21);

jp2.add(check22);

content.add(jp1, BorderLayout.*NORTH*); //分别添加面板到容器中

content.add(jp2, BorderLayout.*CENTER*); //分别添加面板到容器中

}

**public** **void** actionPerformed(ActionEvent e) {}

**class** CheckHandler1 **implements** ItemListener //复选框键的事件响应类

{

**private** **int** sel;

CheckHandler1(**int** select)

{

sel = select;

}

**public** **void** itemStateChanged(ItemEvent e)

{

**switch**(sel)

{

**case** 1:method1();**break**; //调用选择的相应功能的方法

**case** 2:method2();**break**; //调用选择的相应功能的方法

}

}

**private** **void** method1() {}

**private** **void** method2() {}

}

**class** CheckHandler2 **implements** ItemListener //复选框键的事件响应类

{

**private** **int** sel;

CheckHandler2(**int** select)

{

sel = select;

}

**public** **void** itemStateChanged(ItemEvent e)

{

**switch**(sel)

{

**case** 1:method1();**break**; //调用选择的相应功能的方法

**case** 2:method2();**break**; //调用选择的相应功能的方法

}

}

**private** **void** method1() {}

**private** **void** method2() {}

}

**public** **static** **void** main(String[] args) {

test54 SIG = **new** test54();

SIG.setVisible(**true**);

}

}

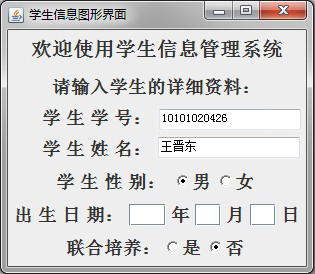
**3、实验总结**

此程序采用顺序结构，程序中用到了java.awt.event.ActionListener，

java.awt.event.ItemEvent，java.awt.event.ItemListener

javax.swing.JButton，javax.swing.JFrame，javax.swing.JLabel

javax.swing.JPanel，javax.swing.JTextField这些包，实现了一个学生信息管理系统的图形界面。可录入的信息包括学生学号、姓名、性别、出生年、月、日、年级以及是否联合培养。

 **4、实验截图**

**题目五：为上例增加事件相应功能**

**1、题目**

为上一题所编写的图形界面增加事件响应功能。当用户选定了一个出生月份，例如3月份时，系统自动把日期的选择范围限制在1~31之间，当用户单击“录入”按钮“时，系统把所有的学生信息打印在屏幕上。

**2、源程序**

**import** java.applet.\*;

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

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

**public** **class** test55 **extends** Applet **implements** ActionListener, ItemListener {

**private** **static** **final** **long** *serialVersionUID* = 1L;

Panel p1 = **new** Panel();

Panel p2 = **new** Panel();

Choice year, month, day;

**int** y, m;

Button next = **new** Button("添加");

Button prev = **new** Button("修改");

Button first = **new** Button("删除");

Button last = **new** Button("退出");

TextField field1 = **new** TextField(10);

TextArea area1 = **new** TextArea(5,10);

TextField field2 = **new** TextField(10);

**public** **void** init() {

**int** i;

setLayout(**new** BorderLayout());

Label label = **new** Label("欢迎使用学生信息管理系统");

label.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 17));

p1.add(label);// 将标签添加到面板jp1中

Label label1 = **new** Label("请输入学生详细资料：");

label1.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

p2.add(label1);// 将标签添加到面板jp2中

Label label2 = **new** Label("学生学号：");

label2.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

Label label3 = **new** Label("学生姓名：");

label3.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

p2.add(label2);

p2.add(field1);

p2.add(label3);

p2.add(field2);

Label label4 = **new** Label("学生性别：");

label4.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

CheckboxGroup group1 = **new** CheckboxGroup();

Checkbox checkl1 = **new** Checkbox("男", **true**, group1);

checkl1.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

checkl1.addItemListener(**this**);

Checkbox checkl2 = **new** Checkbox("女", **false**, group1);

checkl2.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

checkl2.addItemListener(**this**);

p2.add(label4);

p2.add(checkl1);

p2.add(checkl2);

Label label5 = **new** Label("出生日期：");

label5.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

Label label0 = **new** Label();

year = **new** Choice();

month = **new** Choice();

day = **new** Choice();

**for** (i = 1980; i < 2088; i++) {

year.add(Integer.*toString*(i));

}

**for** (i = 1; i < 13; i++) {

month.add(Integer.*toString*(i));

}

**for** (i = 1; i < 32; i++) {

day.add(Integer.*toString*(i));

}

year.addItemListener(**this**);

month.addItemListener(**this**);

day.addItemListener(**this**);

Label label51 = **new** Label("年：");

label51.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

Label label52 = **new** Label("月：");

label52.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

Label label53 = **new** Label("日：");

label53.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

p2.add(label5);

p2.add(label0);

p2.add(year);

p2.add(label51);

p2.add(month);

p2.add(label52);

p2.add(day);

p2.add(label53);

Label label6 = **new** Label("联合培养：");

label6.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

CheckboxGroup group2 = **new** CheckboxGroup();

Checkbox check21 = **new** Checkbox("是", **true**, group2);

check21.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

check21.addItemListener(**this**);

Checkbox check22 = **new** Checkbox("否", **false**, group2);

check22.setFont(**new** Font("楷体\_GB2312", Font.*PLAIN*, 15));

check22.addItemListener(**this**);

p2.add(label6);

p2.add(check21);

p2.add(check22);

p2.add(next);

p2.add(prev);

p2.add(first);

p2.add(last);

p2.add(area1);

next.addActionListener(**this**);

prev.addActionListener(**this**);

first.addActionListener(**this**);

last.addActionListener(**this**);

add(p1, BorderLayout.*NORTH*);

add(p2, BorderLayout.*CENTER*);

}

**public** **void** actionPerformed(ActionEvent e) {

**if**(e.getSource()==last){

System.*exit*(0);

}

**else** **if**(e.getSource()==next){

area1.append(field1.getText()+"\n");

area1.append(field2.getText()+"\n");

}

**else** **if**(e.getSource()==prev){

}

**else**{}

}

**public** **void** itemStateChanged(ItemEvent e) {

y = Integer.*parseInt*(year.getSelectedItem());

m = Integer.*parseInt*(month.getSelectedItem());

**if** (e.getItemSelectable() **instanceof** Choice) {

**if** (y % 400 == 0 || y % 100 == 0 || y % 4 != 0) {

**switch** (m) {

**case** 4:

**case** 6:

**case** 9:

**case** 11:

**if** (day.getSelectedItem() == "31") {

}

**break**;

**case** 2:

**if** (day.getSelectedItem() == "31"

|| day.getSelectedItem() == "30") {

}

**break**;

}

} **else** {

**switch** (m) {

**case** 4:

**case** 6:

**case** 9:

**case** 11:

**if** (day.getSelectedItem() == "31") {

}

**break**;

**case** 2:

**if** (day.getSelectedItem() == "31"

|| day.getSelectedItem() == "30"

|| day.getSelectedItem() == "29") {

}

**break**;

}

}

}

}

}

**3、实验总结**

此程序采用顺序结构，此程序没有遇到错误。

 **4、实验截图**

**题目六：按要求编写一个图形界面的Java Application**

**1、题目**

编写一个图形界面的Java Application，为用户提供三种关闭窗口的方法：使用按钮，适用菜单项，使用窗口关闭图标。

**2、源程序**

**import** java.awt.BorderLayout;

**import** java.awt.FlowLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JPanel;

**public** **class** test56 **extends** JFrame **implements** ActionListener {

JMenuBar menu;

JMenu file;

JMenuItem closeMenu;

**public** test56() {

menu = **new** JMenuBar();

file = **new** JMenu("文件");

closeMenu = **new** JMenuItem("关闭");

closeMenu.addActionListener(**this**);

JButton closeButton = **new** JButton(" 关闭 ");

closeButton.addActionListener(**this**);

JPanel closePanel = **new** JPanel();

closePanel.setLayout(**new** FlowLayout());

closePanel.add(closeButton);

**this**.add(closePanel, BorderLayout.*CENTER*);

**this**.add(menu, BorderLayout.*NORTH*);

menu.add(file);

file.add(closeMenu);

**this**.setBounds(200, 100, 200, 120);

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

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

}

**public** **void** actionPerformed(ActionEvent e) {

System.*exit*(0);

}

**public** **static** **void** main(String[] args) {

**new** test56();

}

}

**3、实验总结**

此实验采用顺序结构， 实验中用到了java.awt.BorderLayout，java.awt.FlowLayout，java.awt.event.ActionEvent，java.awt.event.ActionListener，javax.swing.JButton，javax.swing.JFrame，javax.swing.JMenu，javax.swing.JMenuBar，javax.swing.JMenuItem，javax.swing.JPanel这些包。此程序为图形界面的Java Application¸为用户提供三种关闭窗口的方法：使用按钮，适用菜单项，使用窗口关闭图标。

**4、实验截图**

第六次实验

**题目一至实验三：在图形界面中绘制二维坐标系**

**1、题目**

在图形界面中绘制二维坐标系，其中x轴位于绘图区正中，y轴位于绘图区的最左边。

绘制数据点，每个数据点用一个小叉号表示，并在叉号边以（x，y）的格式标示出数据点的坐标数值，最后用折线连接相邻的数据点。

设计界面接收用户输入的实验数据，绘制数据点和实验曲线。

|  |  |  |
| --- | --- | --- |
| 数据点序号 | x坐标 | y坐标 |
| 1 | 0 | 20 |
| 2 | 50 | -30 |
| 3 | 100 | 16 |
| 4 | 150 | 126 |
| 5 | 200 | 164 |

**2、源程序**

**import** java.awt.Graphics;

**import** java.awt.Graphics2D;

**import** java.awt.RenderingHints;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.util.Vector;

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

**public** **class** test61 **extends** JFrame{

JTextField t,t1,t2;

JPanel pan;

Vector v=**new** Vector();

**boolean** bo=**false**;

**public** test61(){

t=**new** JTextField();

t1=**new** JTextField();

t2=**new** JTextField();

JLabel l1=**new** JLabel("录入数据：");

JLabel l2=**new** JLabel("说明：");

JLabel l3=**new** JLabel("X：");

JLabel l4=**new** JLabel("Y：");

JButton b1=**new** JButton("确定");

JButton b2=**new** JButton("完成");

b1.addActionListener(**new** listen());

b2.addActionListener(**new** listen());

b1.setBounds(510,100,60,25);

b2.setBounds(580,100,60,25);

l1.setBounds(500,10,120,20);

l2.setBounds(500,40,40,20);

l3.setBounds(510,70,20,20);

l4.setBounds(580,70,20,20);

t.setLocation(540,40);

t1.setLocation(530,70);

t2.setLocation(600,70);

t.setSize(100,20);

t1.setSize(40,20);

t2.setSize(40,20);

setSize(680,460);

setLayout(**null**);

getContentPane().add(getpan());

getContentPane().add(t);

getContentPane().add(t1);

getContentPane().add(t2);

getContentPane().add(l1);

getContentPane().add(l2);

getContentPane().add(l3);

getContentPane().add(l4);

getContentPane().add(b1);

getContentPane().add(b2);

setLocationRelativeTo(**null**);

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

**try**{

UIManager.*setLookAndFeel*("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

SwingUtilities.*updateComponentTreeUI*(**this**);

}

**catch**(Exception e){}

setVisible(**true**);

}

**class** listen **implements** ActionListener{

**public** **void** actionPerformed(ActionEvent e) {

**if**(e.getActionCommand().equals("确定")){

dian d=**new** dian(Integer.*valueOf*(t1.getText()),

Integer.*valueOf*(t2.getText()),t.getText());

v.add(d);System.*out*.println(t.getText());

}

**else**{

bo=**true**;

repaint();

}

repaint();

}

}

**public** JPanel getpan(){

**if**(pan==**null**)

pan=**new** JPanel(){

**public** **void** paint(Graphics g){

Graphics2D gg=(Graphics2D)g;

gg.setRenderingHint(RenderingHints.*KEY\_ANTIALIASING*,

RenderingHints.*VALUE\_ANTIALIAS\_ON*);

gg.clearRect(0,0,481,402);

gg.drawLine(0,200,480,200);

gg.drawLine(470,196,480,200);

gg.drawLine(470,204,480,200);

gg.drawLine(0,0,0,481);

gg.drawLine(0,0,4,10);

**for**(**int** i=0;i<v.size();i++){

dian d=(dian)v.elementAt(i);

gg.fillOval(d.getx()-5,195-d.gety(),10,10);

**if**(d.getbo()){

gg.drawString(d.getn(),d.getx()+5,203-d.gety());

}

}

**if**(bo&&v.size()>1){

dian d=(dian)v.elementAt(0);

**for**(**int** i=1;i<v.size();i++){

dian a=(dian)v.elementAt(i);

gg.drawLine(d.getx(),200-d.gety(),a.getx(),200-a.gety());

d=a;

}

}

}

};

pan.setLayout(**null**);

pan.addMouseMotionListener(**new** MouseAdapter(){

**public** **void** mouseMoved(MouseEvent e) {

setTitle("X："+e.getX()+" Y："+(200-e.getY()));

}

});

pan.addMouseListener(**new** MouseAdapter(){

**public** **void** mouseClicked(MouseEvent e) {

**if**(e.getClickCount()==2){

v.clear();

repaint();

}

**else**{

**for**(**int** i=0;i<v.size();i++){

**final** dian d=(dian)v.elementAt(i);

**if**((e.getX()-d.getx())>-4

&&(e.getX()-d.getx())<4

&&(200-e.getY()-d.gety())>-4

&&(200-e.getY()-d.gety())<4){

d.setbo(**true**);

v.add(i,d);

v.remove(i+1);

repaint();

}

}

}

}

});

pan.setSize(481,402);

pan.setLocation(10,10);

**return** pan;

}

**public** **static** **void** main(String[] args){

**new** test61();

}

**class** dian{

**int** x,y;

String s;

**boolean** bo=**false**;

dian(**int** a,**int** b,String c){

x=a;

y=b;

s=c;

}

**public** **int** getx(){

**return** x;

}

**public** **int** gety(){

**return** y;

}

**public** String getn(){

**return** s;

}

**public** **boolean** getbo(){

**return** bo;

}

**public** **void** setbo(**boolean** b){

bo=b;

}

}

}

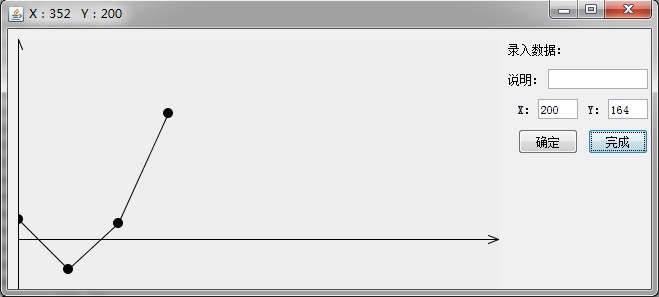
**3、实验总结**

此程序采用顺序结构，程序中用到了java.awt.Graphics，java.awt.Graphics2D，

java.awt.RenderingHints，java.awt.event.ActionEvent

java.awt.event.ActionListener，java.awt.event.MouseAdapter，

java.awt.event.MouseEvent，java.util.Vector，这些类以及 javax.swing.\*包中所有类。此程序实现的功能为：在图形界面中绘制二维的坐标系统。

 **4、实验截图**

**题目四：跟踪用户的鼠标并显示时刻位置**

**1、题目**

编写程序跟踪用户的鼠标，并时刻把鼠标的位置显示出来。

**2、源程序**

**import** java.applet.\*;

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

**public** **class** test64 **extends** Applet {

**int** pointx, pointy;

**public** **void** init() {

pointx = -1;

pointy = -1;

resize(800, 800); // 显示区大小。

}

**public** **void** paint(Graphics g) {

**if** (pointx != -1)

g.drawString("I am here.(X:"+pointx+" Y:"+pointy+")", pointx, pointy);

}

@Override

**public** **boolean** mouseMove(Event evt, **int** x, **int** y) {

pointx = x;

pointy = y; // 在这里捕捉到鼠标的x和y的坐标。

repaint();

**return** **super**.~~mouseMove~~(evt, x, y);

}

}

**3、实验总结**

此程序采用顺序结构，程序中用到了java.applet.\*,import java.awt.\*包中的所有类这些包，实现了程序跟踪用户的鼠标位置，并时刻把鼠标的位置显示出来。

 **4、实验截图**

**题目五：编写图形界面响应鼠标单击和键盘事件**

**1、题目**

编写图形界面的Java程序响应鼠标单击和键盘事件，从鼠标单击的位置开始横向显示用户键入的字符，若鼠标双击则清空屏幕。

**2、源程序**

**import** java.awt.Graphics;

**import** java.util.Date;

**public** **class** test65 **extends** javax.swing.JFrame {

**int** x, y;

**long** nowtime,lasttime;

**int** i = 0;

**public** test65() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

**private** **void** initComponents() {

jPanel1 = **new** javax.swing.JPanel();

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

addKeyListener(**new** java.awt.event.KeyAdapter() {

**public** **void** keyTyped(java.awt.event.KeyEvent evt) {

formKeyTyped(evt);

}

});

jPanel1.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** **void** mouseClicked(java.awt.event.MouseEvent evt) {

jPanel1MouseClicked(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = **new** javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.*LEADING*)

.addGap(0, 400, Short.*MAX\_VALUE*)

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.*LEADING*)

.addGap(0, 300, Short.*MAX\_VALUE*)

);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addComponent(jPanel1, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

);

layout.setVerticalGroup(

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

.addComponent(jPanel1, javax.swing.GroupLayout.Alignment.*TRAILING*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

);

pack();

}

**private** **void** jPanel1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jPanel1MouseClicked

**if**(i == 0){ //判断是否为第一次点击

lasttime = (**new** Date().getTime());

i++;

}

**else**{

nowtime = (**new** Date().getTime());

i++;

}

**if**(i > 1){

i = 0;

}

**if**( Math.*abs*(lasttime - nowtime) < 300){ //若两次间隔小于300MS则视为双击

jPanel1.removeAll(); //重置面板

jPanel1.repaint();

}

x = evt.getX();

y = evt.getY();

}//GEN-LAST:event\_jPanel1MouseClicked

**private** **void** formKeyTyped(java.awt.event.KeyEvent evt) {// 在鼠标点击处显示键盘输入字符

Graphics g = jPanel1.getGraphics();

g.drawString(("" + evt.getKeyChar()), x, y);

x+=6;

}//GEN-LAST:event\_formKeyTyped

**public** **static** **void** main(String args[]) {

**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*(test65.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (InstantiationException ex)

java.util.logging.Logger.*getLogger*(test65.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.*getLogger*(test65.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

}

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

**public** **void** run() {

**new** test65().setVisible(**true**);

}

});

}

**private** javax.swing.JPanel jPanel1;

}

**3、实验总结**

此程序采用顺序结构，程序中用到了java.awt.Graphics，java.util.Date。这些类此程序实现的功能为：图形界面的Java程序响应鼠标单击和键盘事件，从鼠标单击的位置开始横向显示用户键入的字符，若鼠标双击则清空屏幕。

 **4、实验截图**

第七次实验

**题目一、二：创建、抛出并修改异常类的事例**

**1、题目**

选择一个比较了解Java系统异常，例如ArrayIndexOutofBoundsException，编程创建并抛出这个异常类的事例。运行该程序并观察执行结果。修改程序增加异常处理语句，再次运行程序并观察执行结果。

**2、源程序**

**public** **class** test71 {

**public** **static** **void** main(String []args){

TestArray a =**new** TestArray();

**try**{

a.ml();

}

**catch**(ArrayIndexOutOfBoundsException e1){//处理数组下标越界异常

System.*out*.println("数组下标越界");

}

**catch**(ArithmeticException e2){//处理分母为0的运算错误

System.*out*.println("运算错误，分母为0");

e2.printStackTrace();

}

}

}

**class** TestArray{

**private** **int** i;

**private** **int**[]array={1,2,3,4,5};

**void** ml(){

**while**(**true**){

i=(**int**)(Math.*random*()\*10);//产生0~10之间的随机数

System.*out*.println("以随机数为分母的除法结果是：" +100/i);

System.*out*.println("数组 ayyay["+ i +"]的值为" +array[i]);

}

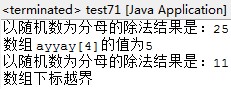
}

}

**3、实验总结**

此程序采用顺序结构，程序中没有用到系统包，程序正确。

**4、实验截图**

****

**题目三：使用Swing图形用户界面实现计数器**

**1、题目**

使用Swing图形用户界面实现计数器。界面包括计算器数值的显示区，计算器加一按扭、计数器减一按扭、计数器清零按扭。运行程序并观察执行结果。

**2、源程序**

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

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

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

**public** **class** test73 **extends** JFrame **implements** ActionListener {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**int** counterValue = 10;

JButton jb = **new** JButton("计数器初始值为10");

JButton jb0 = **new** JButton(" ");

JButton jb1 = **new** JButton("加一");

JButton jb2 = **new** JButton("减一");

JButton jb3 = **new** JButton("清零");

**public** **void** init() {

BorderLayout b = **new** BorderLayout(10,10);

setLayout(b);

add(jb,BorderLayout.*NORTH*);

add(jb0,BorderLayout.*CENTER*);

//add(t1,b.CENTER);

add(jb1,BorderLayout.*WEST*);

add(jb2,BorderLayout.*EAST*);

add(jb3,BorderLayout.*SOUTH*);

jb1.addActionListener(**this**);// 将有事件响应的按钮添加给事件监听者

jb2.addActionListener(**this**);

jb3.addActionListener(**this**);

setVisible(**true**);

setSize(500,300);

setLocation(200,200);

}

**public** **static** **void** main(String args[]){

**new** test73();

}

**public** test73(){

init();

}

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == jb1) {

counterValue++;

jb0.setText(" " + counterValue);// 显示加一后结果

}

**else**

**if** (e.getSource() == jb2) {

counterValue--;

jb0.setText(" " + counterValue);// 显示减一后结果

}

**else** {

counterValue = 0;

jb0.setText(" " + counterValue);// 显示清零结果

}

}

}

**3、实验总结**

此程序采用顺序结构，程序中用到了java.awt.\*，javax.swing.\*包中所有类以及java.awt.event.ActionEvent类和java.awt.event.ActionListener类程序正确。

 **4、实验截图**

**题目四、五：使用JApplet实现某一门课程的分数录入系统**

**1、题目**

使用JApplet实现某一门课程的分数录入系统，界面包括学号、姓名、班级、班号、成绩的输入域和一个录入按扭。当用户单击录入按扭时，程序检查各输入域中是否有非空的数据，若有空域则抛出一个Exception对象；同时程序检查成绩输入域，如果不是数值信息，或是负的数值信息，也抛出相应的异常。在抛出异常之前应提供详细的信息说明，从而使使用该程序的用户了解所出现的问题，同时考虑上述异常出现时程序应该如何处理。编写一个Thread类的子类，在该类内部保存若干条喜欢的习语、警句。它的功能是：每隔固定的时间，例如3秒钟，就随机选择一条格言显示在JApplet录入界面的状态条里。运行程序并观察执行结果。

**2、源程序**

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**import** javax.swing.JOptionPane;

**public** **class** test74 **extends** javax.swing.JApplet {

**public** **void** init() {

**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*(test74.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (InstantiationException ex) {java.util.logging.Logger.*getLogger*(test74.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (IllegalAccessException ex) {java.util.logging.Logger.*getLogger*(test74.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (javax.swing.UnsupportedLookAndFeelException ex) {java.util.logging.Logger.*getLogger*(test74.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

}

**try** {

java.awt.EventQueue.*invokeAndWait*(**new** Runnable() {

**public** **void** run() {

initComponents();

}

});

} **catch** (Exception ex) {

ex.printStackTrace();

}

}

**private** **void** initComponents() {

jToggleButton1 = **new** javax.swing.JToggleButton();

jLabel1 = **new** javax.swing.JLabel();

jLabel2 = **new** javax.swing.JLabel();

jLabel3 = **new** javax.swing.JLabel();

jLabel4 = **new** javax.swing.JLabel();

jLabel5 = **new** javax.swing.JLabel();

jTextField1 = **new** javax.swing.JTextField();

jTextField2 = **new** javax.swing.JTextField();

jTextField3 = **new** javax.swing.JTextField();

jTextField4 = **new** javax.swing.JTextField();

jTextField5 = **new** javax.swing.JTextField();

jPanel1 = **new** javax.swing.JPanel();

jToggleButton1.setText("录入");

jToggleButton1.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** **void** mouseClicked(java.awt.event.MouseEvent evt) {

jToggleButton1MouseClicked(evt);

}

});

jLabel1.setText("学号");

jLabel2.setText("姓名");

jLabel3.setText("班级");

jLabel4.setText("班号");

jLabel5.setText("成绩");

javax.swing.GroupLayout jPanel1Layout = **new** javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.*LEADING*)

.addGap(0, 393, Short.*MAX\_VALUE*)

);

jPanel1Layout.setVerticalGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.*LEADING*)

.addGap(0, 25, Short.*MAX\_VALUE*)

);

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(96, 96, 96)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.*TRAILING*, **false**)

.addComponent(jLabel5, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

.addComponent(jLabel4, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

.addComponent(jLabel2, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

.addComponent(jLabel3, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

.addComponent(jLabel1, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 55, Short.*MAX\_VALUE*))

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

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

.addGroup(layout.createSequentialGroup()

.addComponent(jToggleButton1)

.addContainerGap())

.addGroup(layout.createSequentialGroup()

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

.addComponent(jTextField2, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 185, Short.*MAX\_VALUE*)

.addComponent(jTextField3, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 185, Short.*MAX\_VALUE*)

.addComponent(jTextField4, javax.swing.GroupLayout.*DEFAULT\_SIZE*, 185, Short.*MAX\_VALUE*)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.*LEADING*, **false**)

.addComponent(jTextField1)

.addComponent(jTextField5, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 66, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

.addGap(53, 53, 53))))

.addComponent(jPanel1, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

);

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

.addGroup(javax.swing.GroupLayout.Alignment.*TRAILING*, layout.createSequentialGroup()

.addContainerGap(68, Short.*MAX\_VALUE*)

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

.addComponent(jLabel1)

.addComponent(jTextField1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

.addGap(18, 18, 18)

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

.addComponent(jLabel2)

.addComponent(jTextField2, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

.addGap(16, 16, 16)

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

.addComponent(jLabel3)

.addComponent(jTextField3, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

.addGap(28, 28, 28)

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

.addComponent(jLabel4)

.addComponent(jTextField4, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

.addGap(18, 18, 18)

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

.addComponent(jTextField5, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

.addComponent(jLabel5))

.addGap(14, 14, 14)

.addComponent(jToggleButton1)

.addGap(18, 18, 18)

.addComponent(jPanel1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

);

}

**private** **void** jToggleButton1MouseClicked(java.awt.event.MouseEvent evt) {

String name = jTextField2.getText();

String num = jTextField1.getText();

String sclass = jTextField3.getText();

String sclassnum = jTextField4.getText();

String grade = jTextField4.getText();

**int** nisdigi = 0;

**try** {

**if**( "".equals(num) || "".equals(name) || "".equals(sclass) || "".equals(sclassnum) || "".equals(grade) )

{

JOptionPane.*showMessageDialog*(**this**, "数据不能为空", "错误", 0);

**throw** **new** Exception( "数据为空" ) ;

}

}

**catch** (Exception ex) {

Logger.*getLogger*(test74.**class**.getName()).log(Level.*SEVERE*, **null**, ex);

}

**try**{

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

**if** (grade.charAt(i) <48 || grade.charAt(i)>57){

nisdigi = 1;

**break**;

}

}

**if**( nisdigi == 1){

JOptionPane.*showMessageDialog*(**this**, "成绩不是数字", "错误", 0);

**throw** **new** Exception( "成绩不是数字" ) ;

}

**else**{

**int** c = Integer.*parseInt*(grade);

**if** (c < 0 ){

JOptionPane.*showMessageDialog*(**this**, "成绩不能小于0", "错误", 0);

**throw** **new** Exception( "成绩不能小于0" ) ;

}

}

}

**catch** (Exception ex){

Logger.*getLogger*(test74.**class**.getName()).log(Level.*SEVERE*, **null**, ex);

}

}

**private** javax.swing.JLabel jLabel1;

**private** javax.swing.JLabel jLabel2;

**private** javax.swing.JLabel jLabel3;

**private** javax.swing.JLabel jLabel4;

**private** javax.swing.JLabel jLabel5;

**private** javax.swing.JPanel jPanel1;

**private** javax.swing.JTextField jTextField1;

**private** javax.swing.JTextField jTextField2;

**private** javax.swing.JTextField jTextField3;

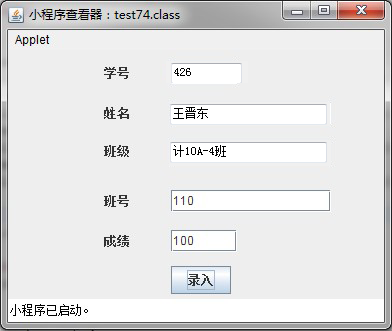
**private** javax.swing.JTextField jTextField4;

**private** javax.swing.JTextField jTextField5;

**private** javax.swing.JToggleButton jToggleButton1;

}

**3、实验总结**

****此程序采用顺序结构，程序中用到了java.util.logging.Level，java.util.logging.Logger，javax.swing.JOptionPane这些类此程序的功能为JApplet实现某一门课程的分数录入系统，界面包括学号、姓名、班级、班号、成绩的输入域和一个录入按扭。当用户单击录入按扭时，程序检查各输入域中是否有非空的数据，若有空域则抛出一个Exception对象；同时程序检查成绩输入域，如果不是数值信息，或是负的数值信息，也抛出相应的异常。在抛出异常之前应提供详细的信息说明，从而使使用该程序的用户了解所出现的问题，同时考虑上述异常出现时程序应该如何处理。 编写一个Thread类的子类，在该类内部保存若干条喜欢的习语、警句。它的功能是：每隔固定的时间，例如3秒钟，就随机选择一条格言显示在JApplet录入界面的状态条里。

**4、实验截图**

第八次实验

**题目一：接受用户输入的文件名并检查文件信息**

**1、题目**

编程接受用户输入的一个文件名（可以包括路径名），检查这个文件是否存在、是否可读、是否可写，并将结果在屏幕上输出。

**2、源程序**

**import** java.io.File;

**import** java.util.Scanner;

**public** **class** test81 {

**public** **static** **void** main(String[] args) {

String path;

**boolean** canread;

**boolean** canwrite;

**boolean** canexe;

Scanner input = **new** Scanner(System.*in*);

path = input.next(); //接受用户输入

File file = **new** File(path); //新建文件类

canexe = file.canExecute(); //取得执行权限信息

canwrite = file.canWrite(); //取得可写权限信息

canread = file.canRead(); //取得可读权限信息

System.*out*.println("可读:" + canread + " \n可写:" + canwrite + " \n可执行:" + canread); //输出信息

}

}

**3、实验总结**

此程序采用顺序结构，程序中用到了java.io.File类以及java.util.Scanner类，程序没有遇到错误。此程序的作用是接受用户输入的一个文件名（可以包括路径名），检查这个文件是否存在、是否可读、是否可写，并将结果在屏幕上输出。

**4、实验截图**

|  |  |
| --- | --- |
| 可读、可写、可执行 | 可读、不可写、可执行 |

**题目二：以二进制保存用户输入的数据**

**1、题目**

编写图形界面接受用户输入的10个整数并以二进制方式保存在指定的文件中。用DOS的type命令打开这个文件观察，是否能看到所输入的数字？为什么？编程打开文件，以二进值读取数据并显示在屏幕上，它们是否与输入的数字相符？为什么？

**2、源程序**

**import** java.io.DataOutputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**public** **class** test82 **extends** javax.swing.JFrame {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**public** **void** NewJFrame() {

initComponents();

}

**private** **void** initComponents() {

jTextField1 = **new** javax.swing.JTextField();

jButton1 = **new** javax.swing.JButton();

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

jButton1.setText("写入");

jButton1.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** **void** mouseClicked(java.awt.event.MouseEvent evt) {

jButton1MouseClicked(evt);

}

});

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addGap(38, 38, 38)

.addComponent(jTextField1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 323, javax.swing.GroupLayout.*PREFERRED\_SIZE*))

.addGroup(layout.createSequentialGroup()

.addGap(139, 139, 139)

.addComponent(jButton1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, 98, javax.swing.GroupLayout.*PREFERRED\_SIZE*)))

.addContainerGap(37, Short.*MAX\_VALUE*))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(50, 50, 50)

.addComponent(jTextField1, javax.swing.GroupLayout.*PREFERRED\_SIZE*, javax.swing.GroupLayout.*DEFAULT\_SIZE*, javax.swing.GroupLayout.*PREFERRED\_SIZE*)

.addGap(18, 18, 18)

.addComponent(jButton1)

.addContainerGap(23, Short.*MAX\_VALUE*))

);

pack();

}

**private** **void** jButton1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jButton1MouseClicked

String shuru = jTextField1.getText(); //取得输入路径

String path = "g:\\abc1.txt"; //保存在txt文件中

String[] got = shuru.split(" "); //以空格分开字符串

FileOutputStream fout;

**try** {

fout = **new** FileOutputStream(path);

DataOutputStream dout = **new** DataOutputStream(fout);

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

**try** {

dout.writeUTF(got[i]);

} **catch** (IOException ex) {

Logger.*getLogger*(test82.**class**.getName()).log(Level.*SEVERE*, **null**, ex);

}

}

} **catch** (FileNotFoundException ex) {

Logger.*getLogger*(test82.**class**.getName()).log(Level.*SEVERE*, **null**, ex);

}

}

**public** **static** **void** main(String args[]) {

**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*(test82.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (InstantiationException ex) {

java.util.logging.Logger.*getLogger*(test82.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (IllegalAccessException ex) {

java.util.logging.Logger.*getLogger*(test82.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

} **catch** (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.*getLogger*(test82.**class**.getName()).log(java.util.logging.Level.*SEVERE*, **null**, ex);

}

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

**public** **void** run() {

**new** test82().setVisible(**true**);

}

});

}

**private** javax.swing.JButton jButton1;

**private** javax.swing.JTextField jTextField1;

}

**3、实验总结**

此程序采用顺序结构，程序中用到了java.io.DataOutputStream;

java.io.FileNotFoundException，java.io.FileOutputStream，

java.io.IOException，java.util.logging.Level，java.util.logging.Logger2这些类。此程序的作用是接受用户输入的10个整数并一二进制方式保存在指定的文件中。

**题目三：在用户输入文件名的文档中保存用户输入的字符串**

**1、题目**

编程接受两个命令行参数，第一个是一个文件名，第二个是一个字符串。程序的功能是把第二个命令行参数以文本方式保存到第一个文件中。运行该程序，再使用DOS的type命令打开文件，观察文件所写人的信息是否与命令行的参数相符，解释这个结果。

**2、源程序**

**import** java.io.FileWriter;

**import** java.io.IOException;

**public** **class** test83 {

**public** **static** **void** main(String[] args) {

StringBuffer sb=**new** StringBuffer();

sb.append(args[1]); //在字符串BUFFER中添加第二个参数

**try** {

FileWriter fw=**new** FileWriter(args[0]);

//新建以第一个参数为文件名的文件写入类

fw.write(sb.toString()); //写入sb

fw.close(); //关闭文件

} **catch** (IOException e) {

}

}

}

**3、实验总结**

此程序采用顺序结构，程序中用到了java.io.FileWriter，java.io.IOException这些类，程序正确。此程序的功能为接受两个命令行参数，第一个是一个文件名，第二个是一个字符串。

**题目四：编写将两个文件中的内容合并成一个新文件的程序**

**1、题目**

编写将两个文件（文件1、文件2）中的内容合并成一个新文件（文件3）。合并方法是：从文件一读取一个字节放入文件3，在从文件2读取一个字节放入文件3，如此轮流直至某一个文件读完，再将较长文件中的剩余部分读取放入至文件3。可以使用图形界面或命令行参数输入三个文件名。

**2、源程序**

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**public** **class** test {

**public** **static** **void** main(String[] args) {

**int** ch1 = 0;

**int** ch2 = 0;

**try**{

FileReader fr1 = **new** FileReader(args[0]);

FileReader fr2 = **new** FileReader(args[1]);

FileWriter fw = **new** FileWriter(args[2]);

**while**((ch1=fr1.read())!=-1&&(ch2=fr2.read())!=-1){

fw.write(ch1);

fw.write(ch2);

}

**while**((ch1=fr1.read())!=-1){

fw.write(ch1);

}

**while**((ch2=fr2.read())!=-1){

fw.write(ch2);

}

fw.close();

}**catch**(IOException e){

}

}

}

**3、实验总结**

此程序采用顺序结构，程序中用到了j java.io.File，java.io.FileReader，

java.io.FileWriter，java.io.IOException这些类，程序正确。

第九次实验

**题目一：接受用户输入的主机名并打印IP地址**

**1、题目**

编写接收用户输入的一个主机名，通过网络找的其IP地址打印到屏幕上。

**2、源程序**

**import** java.net.\*;

**public** **class** exp9\_1 {

**public** **static** **void** main(String[] args) **throws** UnknownHostException

{

InetAddress address = InetAddress.*getByName*("www.ncut.edu.cn");

System.*out*.println(address.getHostName());

System.*out*.println(address.getHostAddress());

}

}

**3、实验总结**

此程序采用顺序结构，没有遇到问题。

 **4、实验截图**

|  |  |
| --- | --- |
|  |  |

**题目二：接受用户输入的网页名并打印电子邮件地址**

**1、题目**

编写Applet接收用户输入一个网页名，通过阅读该网页并寻找网页中所以的电子邮件地址，返回给用户，并显示在屏幕上。

**2、源程序**

**import** java.io.BufferedReader;

**import** java.io.InputStreamReader;

**import** java.net.URL;

**import** java.net.URLConnection;

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

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

**public** **class** exp9\_2 {

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

URL url = **new** URL("http://www.spsu.edu");

URLConnection conn = url.openConnection();

conn.setConnectTimeout(1000 \* 10);

BufferedReader bufr = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));

String line = **null**;

String regex = "[a-zA-Z0-9\_-]+@\\w+\\.[a-z]+(\\.[a-z]+)?";

Pattern p = Pattern.*compile*(regex);

**while**((line = bufr.readLine()) != **null**) {

Matcher m = p.matcher(line);

**while**(m.find()) {

System.*out*.println(m.group());

}

}

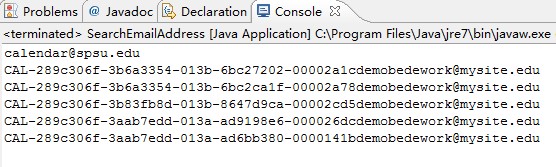
}

}

**3、实验总结**

此程序采用顺序结构，没有遇到问题。

**4、实验截图**



**题目三：利用流式Socket通信机制模拟手机的短信息服务**

**1、题目**

利用流式Socket通信机制模拟手机的短信息服务，编程时可以利用实验10中已经实现的手机界面。修改这个界面，增加一个“短信”按扭。设服务器中已事先保存了10条短信息，编号为0～9。当手机用户按“短信”按扭时，程序将是手机连接到服务器。服务器向手机用户提示，要求输入短信息的序号；手机用户按下0～9中的某个按扭后，服务器将相应的短信息传送给手机用户，在手机的显示区上显示。

**2、源程序**

**import** java.io.\*;

**import** java.net.\*;

**public** **class** exp9\_3 {

**static** **boolean** *connect*=**false**;

**static** **boolean** *start*=**true**;

**static** String[] *line*={

"Hello world 0",

"Hello world 1",

"Hello world 2",

"Hello world 3",

"Hello world 4",

"Hello world 5",

"Hello world 6",

"Hello world 7",

"Hello world 8",

"Hello world 9",

};

**static** **final** **int** *z*=-1;

**static** **int** *s*=-1;

**public** **static** **void** main(String[] args) {

**try**{

ServerSocket server=**null**;

**try**{

server=**new** ServerSocket(4700); //创建一个ServerSocket在端口4700监听客户请求

}**catch**(Exception e) {

System.*out*.println("1.can not listen to:"+e);

}

Socket socket=**null**;

**try**{

socket=server.accept(); //使用accept()阻塞等待客户请求，有客户请求到来则产生一个Socket对象，并继续执行

}**catch**(Exception e) {

System.*out*.println("2.Error."+e);

}

BufferedReader is=**new** BufferedReader(**new** InputStreamReader(socket.getInputStream()));

PrintWriter os=**new** PrintWriter(socket.getOutputStream());

**int** n=*s*;

**int** t=-1;

String str="";

**while**(*start*)

{

str=is.readLine();

System.*out*.println(str);

**if**(str.equals("connect"))

{

os.println("Please choose the text from 0-9, and end with #\n");

os.flush();

str="";

t=1;

}

**else** **if**(t>0)

{

n=Integer.*parseInt*(str,10);

**if**(*s*!=n) {

*connect*=**true**;*s*=n;

}

**while**(*connect*)

{

os.println(*line*[n]);

os.flush();

*s*=*z*;n=*z*;

*connect*=**false**;

}

}

**else** **if**(str==**null**)

{

}

}

os.close();

is.close();

socket.close();

server.close();

}**catch**(Exception e){

System.*out*.println("3.Error:"+e);

}

System.*out*.println("finished");

}

}

**3、实验总结**

此程序采用顺序结构，没有遇到问题。

**4、实验截图**