**3. 编写应用程序，求1！+2！+…+10！。**

答:

class Fact

{

public static void main(String args[])

{

int fact,sum=0;

for(int i=1;i<=10;i++)

{

fact=1;

for(int j=1;j<=i;j++)

fact\*=j;

sum+=fact;

}

System.out.println("1到10的阶乘之和是:"+sum);

}

}

**4. 编写一个应用程序，求100以内的全部素数。**

答:

class Primes

{ public static void main(String args[])

{ int w=1;

for(int i=2;i<=100;i++)

{ for(int j=2;j<i;j++)

{ w=i%j;

if(w==0)

break;

}

if(w!=0)

System.out.println(i+"是素数");

}

}

}

**5. 分别用do―while和for循环计算1+1/2！+1/3！+1/4！+…的前20项和。**

答: ①for循环

class Sum

{

public static void main(String args[])

{

int fact;

double sum=0;

for(int i=1;i<=20;i++)

{

fact=1;

for(int j=1;j<=i;j++)

fact\*=j;

sum+=1.0/fact;

}

System.out.println(sum);

}

}

②do―while循环

class Sum

{

public static void main(String args[])

{

int i=1;

int fact;

double sum=0;

do

{

fact=1;

int j=0;

while(++j<=i)

fact\*=j;

sum+=1.0/fact;

}

while(++i<=20);

System.out.println(sum);

}

}

**6. 一个数如果恰好等于它的因子之和，这个数就称为“完数”。编写应用程序，求1000之内的所有完数。**

答:

class Wanshu

{

public static void main(String args[])

{

int x,i,sum;

for(x=1;x<=1000;x++)

{

sum=0;

for(i=1;i<x;i++)

{

if(x%i==0)

sum+=i;

}

if(x==sum)

System.out.println(x+"是完数");

}

}

}

**7. 编写应用程序，分别使用while和for循环计算8+88+888+…前10项之和。**

答: ①for循环

class TheSum

{

public static void main(String args[])

{

long sum=0,data=8;

for(int i=1;i<=10;i++)

{

sum=sum+data;

data=data\*10+8;

}

System.out.println(sum);

}

}

②while循环

class TheSum

{

public static void main(String args[])

{

long sum=0,data=8,i=0;

while(++i<=10)

{

sum=sum+data;

data=data\*10+8;

}

System.out.println(sum);

}

}

**8. 编写应用程序，输出满足1+2+3+…+n<8888的最大正整数n。**

答:

class Maxn

{

public static void main(String args[])

{

int k=1,sum=0;

while((sum+=k)<8888)

k++;

k--;

System.out.println("最大能取到:"+k);

}

**15.模仿例子4.27，编写一个类实现两个接口的程序。**

答:

② interface 表面积

{

double allArea(double r);

}

interface 体积

{

double volu(double r);

}

class Sph implements 表面积,体积

{

double PI=3.14159;

public double allArea(double r)

{

return 4\*PI\*r\*r;

}

public double volu(double r)

{

return 4.0/3\*PI\*r\*r\*r;

}

}

public class Test

{

public static void main(String args[])

{

double r=5.0;

Sph a=new Sph();

System.out.println("半径为5的球的表面积是:"+a.allArea(r));

System.out.println("半径为5的球的体积是:"+a.volu(r));

}

}

**16.编写一个类A，该类创建的对象可以调用方法f输出英文字母表，然后再编写一个该类的子类B，要求子类B必须继承A类的方法f(不允许重写)，子类创建的对象不仅可以调用方法f输出英文字母表，而且调用子类新增的方法g输出希腊字母表。**

答: class A

{

int m;

void f()

{

for(m=65;m<91;m++)

System.out.print((char)m+" ");

for(m=97;m<123;m++)

System.out.print((char)m+" ");

System.out.println(" ");

}

}

class B extends A

{

int i;

void g()

{

for(i=913;i<930;i++)

System.out.print((char)i+" ");

for(i=931;i<938;i++)

System.out.print((char)i+" ");

for(i=945;i<962;i++)

System.out.print((char)i+" ");

for(i=963;i<970;i++)

System.out.print((char)i+" ");

System.out.println(" ");

}

}

public class Test

{

public static void main(String args[])

{

B b=new B();

System.out.println("我调用方法f输出英文字母表:");

b.f();

System.out.println("我调用方法g输出希腊字母表:");

b.g();

}

}

**17.编写一个异常类MyException，再编写一个类Student，该类有一个产生异常的方法public void speak(int m) throws MyException，要求参数m的值大于1000时，方法抛出一个MyException对象。最后编写主类，在主类的main方法中用Student创建一个对象，让该对象调用speak方法。**

答: class MyException extends Exception

{

String str1;

MyException(int m)

{

str1=m+"出现错误 可能造成的原因是取值大于1000";

}

public void showStr1()

{

System.out.println(str1);

}

}

class Student

{

public void speak(int m) throws MyException

{

if(m>1000)

{

MyException exception=new MyException(m);

throw exception;

}

else

System.out.println(m);

}

}

public class Test

{

public static void main(String agrs[])

{

int m;

Student stu1=new Student();

m=987;

try

{

stu1.speak(m);

m=1234;

stu1.speak(m);

}

catch(MyException e)

{

e.showStr1();

}

}

**18.编写一个类，该类有一个方法public int f(int a,int b)，该方法返回a和b的最大公约数。然后再编写一个该类的子类，要求子类重写方法f，而且重写的方法将返回a和b的最小公倍数。要求在重写的方法的方法体中首先调用被隐藏的方法返回a和b的最大公约数m，然后将乘积(a\*b)/m返回。要求在应用程序的主类中分别使用父类和子类创建对象，并分别调用方法f计算两个正整数的最大公约数和最小公倍数。**

答: class A

{

public int f(int a,int b)

{

if(a<b)

{

int temp=0;

temp=a;

a=b;

b=temp;

}

int r=a%b;

while(r!=0)

{

a=b;

b=r;

r=a%b;

}

return b;

}

}

class B extends A

{

public int f(int a,int b)

{

int m;

m=super.f(a,b);

return (a\*b)/m;

}

}

public class Test

{

public static void main(String args[])

{

A a=new A();

System.out.println("18和102的最大公约数是:"+a.f(18,102));

B b=new B();

System.out.println("18和102的最小公倍数是:"+b.f(18,102));

}

}

**1. 使用String类的public String toUpperCase()方法可以将一个字符串中的小写字母变成大写字母，使用public String toLowerCase()方法可以将一个字符串中的大写字母变成小写字母。编写一个程序，使用这两个方法实现大小写的转换。**

答: class Test

{

public static void main(String args[])

{

String str="I can use Java";

System.out.println("要转换的字符串是:"+str);

String s=str.toUpperCase();

System.out.println("转换成大写字符串是:"+s);

s=str.toLowerCase();

System.out.println("转换成小写字符串是:"+s);

}

}

**2. 使用String类的public String concat(String str)方法可以把调用该方法的字符串与参数指定的字符串连接，把str指定的串连接到当前串的尾部获得一个新的串。编写一个程序通过连接两个串得到一个新串，并输出这个新串。**

答: class Test

{

public static void main(String args[])

{

String str1="I can u";

String str2="se Java";

String s=str1.concat(str2);

System.out.println("将字符串"+str1+"与字符串"+str2+"连接后得到的新字符串是:");

System.out.println(s);

}

}

**1. 用Data类不带参数的构造方法创建日期，要求日期的输出格式是:星期 小时 分 秒。**

答: import java.util.\*;

import java.text.\*;

class Test

{

public static void main(String args[])

{

Date 时间=new Date();

SimpleDateFormat s=new SimpleDateFormat("E HH时 mm分 ss秒");

System.out.println(s.format(时间));

}

}

**3. 计算某年、某月、某日和某年、某月、某日之间的天数间隔。要求年、月、日通过main方法的参数传递到程序中。**

答: import java.util.\*;

class Test

{

public static void main(String args[])

{

Calendar c=Calendar.getInstance();

c.set(2000,0,1);

long time1=c.getTimeInMillis();

c.set(2008,7,8);

long time2=c.getTimeInMillis();

long dayCous=(time2-time1)/(1000\*60\*60\*24);

System.out.println("2008年8月8日和2000年1月1日相隔"+dayCous+"天");

}

}

**5. 使用BigInteger类计算1！+3！+5！+7！+…的前30项的和。**

答: import java.math.\*;

public class Test

{

public static void main(String args[])

{

BigInteger sum=new BigInteger("0"),

jieCheng=new BigInteger("1"),

ONE=new BigInteger("1"),

i=ONE;

int k=0;

while(++k<=30)

{

sum=sum.add(jieCheng);

i=i.add(ONE);

jieCheng=jieCheng.multiply(i);

i=i.add(ONE);

jieCheng=jieCheng.multiply(i);

}

System.out.println(sum);

}

}

**4. 编写应用程序，有一个标题为“计算的窗口”的窗口，窗口的布局为FlowLayout布局。窗口中添加两个文本区，当我们在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运算并求出平均值，也就是说随着输入的变化，另一个文本区不断地更新求和及平均值。**

答: import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

class Calculated extends Frame implements TextListener

{

TextArea text1,text2; //定义了2个文本区

Calculated(String s) //标题为“计算的窗口”的窗口

{

super(s);

setLayout(new FlowLayout()); //窗口布局为FlowLayout

text1=new TextArea(5,23);

text2=new TextArea(5,23);

add(text1);

add(text2);

text2.setEditable(false); //显示求和结果和平均值的文本区禁止编辑

text1.addTextListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

setBounds(100,100,400,160);

setVisible(true);

validate();

}

public void textValueChanged(TextEvent e)

{

String s=text1.getText();

StringTokenizer geshu=new StringTokenizer(s);

int n=geshu.countTokens();

double a[]=new double[n];

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

{

String temp=geshu.nextToken();

double date=Double.parseDouble(temp);

a[i]=date;

}

double sum=0,average;

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

{

sum=sum+a[i];

}

average=sum/n;

text2.setText(null);

text2.append("和:"+sum+"\n"+"平均数:"+average);

}

}

public class Test

{

public static void main(String args[])

{

Calculated calc=new Calculated("计算的窗口");

}

}

**5. 文本区可以使用getSelectedText()方法获取该文本区通过拖动鼠标选中的文件。编写应用程序，有一个标题为“挑单词”的窗口，窗口的布局为BorderLayout布局。窗口中添加两个文本去和一个按钮组件，要求文本区分别添加到窗口的东部区域和西部区域；按钮添加到窗口的南部区域，当单击按钮时，程序将东部区域的文本区中鼠标选中的内容尾加到西部区域的文本区中。**

答: import java.awt.\*;

import java.awt.event.\*;

class WindowSelectedText extends Frame implements ActionListener

{

TextArea text1,text2; //定义2个文本区

Button button; //定义一个按钮

WindowSelectedText(String s) //窗口名字为“挑单词”

{

super(s);

setLayout(new BorderLayout()); //窗口布局是BorderLayout布局

text1=new TextArea(6,15);

text2=new TextArea(6,15);

button=new Button("确定");

add(text1,BorderLayout.EAST);

add(text2,BorderLayout.WEST);

add(button,BorderLayout.SOUTH);

button.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

text2.setEditable(false);

setBounds(100,100,350,200);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==button)

{

String s=text1.getSelectedText()+"\n";

String str=text2.getText();

text2.setText(str+s);

}

}

}

public class Test

{

public static void main(String args[])

{

new WindowSelectedText("挑单词");

}

}

**7. 改进例子7.16，在程序中增加一个名称为“确定”的按钮和一个文本区。当单击按钮时，程序验证用户是否输入了合法的E-mail地址格式，如果合法就将用户输入的姓名、E-mail和职业尾加到文本区中，否则在输入E-mail的文本框中提示用户输入了非法格式的E-mail地址。**

答: import javax.swing.\*;

import javax.swing.border.\*;

import java.awt.\*;

import java.awt.event.\*;

class WindowBox extends Frame implements ActionListener

{

TextField text1,text2,text3;

TextArea textarea;

Box baseBox,boxV1,boxV2;

Button button;

WindowBox()

{

button=new Button("确定");

textarea=new TextArea(6,12);

text1=new TextField(12);

text2=new TextField(12);

text3=new TextField(12);

boxV1=Box.createVerticalBox();

boxV1.add(new Label("姓名"));

boxV1.add(Box.createVerticalStrut(8));

boxV1.add(new Label("email"));

boxV1.add(Box.createVerticalStrut(8));

boxV1.add(new Label("职业"));

boxV2=Box.createVerticalBox();

boxV2.add(text1);

boxV2.add(Box.createVerticalStrut(8));

boxV2.add(text2);

boxV2.add(Box.createVerticalStrut(8));

boxV2.add(text3);

baseBox=Box.createHorizontalBox();

baseBox.add(boxV1);

baseBox.add(Box.createHorizontalStrut(10));

baseBox.add(boxV2);

setLayout(new FlowLayout());

add(baseBox);

add(button);

add(textarea);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

textarea.setEditable(false);

button.addActionListener(this);

setBounds(100,100,210,250);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

boolean a;

int b;

String s;

if(e.getSource()==button)

{

s=text2.getText();

a=s.endsWith(".com");

b=s.indexOf("@");

if(a&&b>0)

{

String str1=text1.getText()+"\n"+text2.getText()+"\n"+text3.getText()+"\n";

String str2=textarea.getText();

textarea.setText(str2+str1);

}

else

{

text2.setText("输入了非法格式的E-mail地址");

}

}

}

}

public class Test

{

public static void main(String args[])

{

new WindowBox();

}

}

**8. 写一个应用程序，要求编写一个Panel的子类MyPanel，MyPanel中有一个文本框和一个按钮，要求MyPanel的实例作为其按钮的ActionEvent事件的监视器，当单击按钮时，程序获取文本框中的文本，并将该文本作为按钮的名称。然后在编写一个Frame的子类，即窗口。窗口的布局为BorderLayout布局。窗口中添加两个MyPanel面板，分别添加到窗口的东部区域和西部区域。**

答: import java.awt.\*;

import java.awt.event.\*;

class MyPanel extends Panel implements ActionListener

{

String name;

TextField text;

Button button;

MyPanel()

{

text=new TextField(10);

button=new Button("确定");

add(text);

add(button);

button.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==button)

{

name=text.getText();

button.setLabel(name);

}

}

}

class MyFrame extends Frame

{

MyPanel panel1,panel2;

MyFrame()

{

panel1=new MyPanel();

panel2=new MyPanel();

add(panel1,BorderLayout.EAST);

add(panel2,BorderLayout.WEST);

setBounds(100,100,400,100);

setVisible(true);

validate();

}

}

public class Test

{

public static void main(String args[])

{

MyFrame win=new MyFrame();

}

}

**9. 参照例子7.18编写一个应用程序，要求有一个画布，在画布上绘制一个矩形，用户通过文本框输入矩形的宽和高以及矩形左上角的位置坐标。**

答: import java.awt.\*;

import java.awt.event.\*;

class Mycanvas extends Canvas

{

int x,y,w,h;

Mycanvas()

{

setBackground(Color.cyan);

}

public void setX(int x)

{

this.x=x;

}

public void setY(int y)

{

this.y=y;

}

public void setW(int w)

{

this.w=w;

}

public void setH(int h)

{

this.h=h;

}

public void paint(Graphics g)

{

g.drawRect(x,y,w,h);

}

}

class WindowCanvas extends Frame implements ActionListener

{

Mycanvas canvas;

TextField text1,text2,text3,text4;

Button button;

WindowCanvas()

{

canvas=new Mycanvas();

text1=new TextField(4);

text2=new TextField(4);

text3=new TextField(5);

text4=new TextField(5);

Panel pNorth=new Panel(),pSouth=new Panel();

button=new Button("确定");

button.addActionListener(this);

pNorth.add(new Label("矩形的宽: "));

pNorth.add(text3);

pNorth.add(new Label("矩形的高: "));

pNorth.add(text4);

pSouth.add(new Label("左上角位置坐标:"));

pSouth.add(text1);

pSouth.add(text2);

pSouth.add(button);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

add(canvas,BorderLayout.CENTER);

add(pNorth,BorderLayout.NORTH);

add(pSouth,BorderLayout.SOUTH);

setBounds(100,100,500,500);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

int x,y,w,h;

try

{

x=Integer.parseInt(text1.getText());

y=Integer.parseInt(text2.getText());

w=Integer.parseInt(text3.getText());

h=Integer.parseInt(text4.getText());

canvas.setX(x);

canvas.setY(y);

canvas.setW(w);

canvas.setH(h);

canvas.repaint();

}

catch(NumberFormatException ee)

{

x=0;y=0;w=0;h=0;

}

}

}

public class Test

{

public static void main(String args[])

{

new WindowCanvas();

}

}

**10.编写应用程序，有一个窗口对象，该窗口取它的默认布局: BorderLayout布局,北面添加一个List组件，该组件有四个商品名称的选项。中心添加一个文本区，当选择List组件中的某个选项后，文本区显示对该商品的价格和产地:当双击List组件中的某个选项后，文本区显示该商品的详细广告。**

答: import java.awt.\*;

import java.awt.event.\*;

class WindowGoods extends Frame implements ActionListener,ItemListener

{

String s[]={"产地:北京","产地:上海","产地:沈阳","产地:广东"};

String p[]={"价格:3200","价格:158","价格:13.2","价格:320/打"};

String a[]={"本商品\*\*\*\*","本商品\*\*\*\*\*","本商品\*\*\*\*\*\*","本商品\*\*\*\*\*\*\*"};

List list;

TextArea text;

WindowGoods()

{

list=new List(3,false);

text=new TextArea(6,20);

text.setEditable(false);

list.add("商品1");

list.add("商品2");

list.add("商品3");

list.add("商品4");

add(list,BorderLayout.NORTH);

add(text,BorderLayout.CENTER);

list.addItemListener(this);

list.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

setBounds(100,100,300,300);

setVisible(true);

validate();

}

public void itemStateChanged(ItemEvent e)

{

if(e.getItemSelectable()==list)

{

int m=list.getSelectedIndex();

text.setText(p[m]+'\n'+s[m]);

}

}

public void actionPerformed(ActionEvent e)

{

int n=list.getSelectedIndex();

text.setText(a[n]);

}

}

public class Test

{

public static void main(String args[])

{

new WindowGoods();

}

}

**11.编写程序，观察各种组件设置背景色和前景色的情况。**

答: import java.awt.\*;

import java.awt.event.\*;

class WindowColor extends Frame implements ActionListener

{

Button button; //按钮

TextField textfield; //文本框

TextArea textarea; //文本区

Mypanel panel; //面板

Checkbox box; //选择框

Choice choice; //下拉列表

List list; //滚动列表

Label label; //标签

Mycanvas can; //画布

Button buttonBackColor,buttonForeColor;

WindowColor()

{

button=new Button("我是按钮");

textfield=new TextField("我是文本框",10);

textarea=new TextArea(6,15);

textarea.setText("我是文本区");

textfield.setEditable(false);

textarea.setEditable(false);

panel=new Mypanel();

box=new Checkbox("我是选择框");

choice=new Choice();

choice.add("我是下拉列表");

list=new List(3,false);

list.add("我是滚动列表");

label=new Label("我是标签");

can=new Mycanvas();

buttonBackColor=new Button("背景色");

buttonForeColor=new Button("前景色");

setLayout(new FlowLayout());

add(button);

add(textfield);

add(textarea);

add(panel);

add(box);

add(choice);

add(list);

add(label);

add(can);

add(buttonBackColor);

add(buttonForeColor);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

buttonBackColor.addActionListener(this);

buttonForeColor.addActionListener(this);

setBounds(100,100,300,300);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==buttonBackColor)

{

button.setBackground(Color.yellow);

textfield.setBackground(Color.yellow);

textarea.setBackground(Color.yellow);

panel.setBackground(Color.yellow);

box.setBackground(Color.yellow);

choice.setBackground(Color.yellow);

list.setBackground(Color.yellow);

label.setBackground(Color.yellow);

can.setBackground(Color.yellow);

}

else if(e.getSource()==buttonForeColor)

{

button.setForeground(Color.blue);

textfield.setForeground(Color.blue);

textarea.setForeground(Color.blue);

panel.setForeground(Color.blue);

box.setForeground(Color.blue);

choice.setForeground(Color.blue);

list.setForeground(Color.blue);

label.setForeground(Color.blue);

can.setForeground(Color.blue);

}

}

}

class Mycanvas extends Canvas

{

Mycanvas(){ }

public void paint(Graphics g)

{

g.drawString("我是画布",5,5);

}

}

class Mypanel extends Panel

{

Button button1;

Mypanel()

{

button1=new Button("我是面板");

add(button1);

}

}

public class Test

{

public static void main(String args[])

{ new WindowColor();

}

}

**12.编写应用程序，有一个标题为“移动”的窗口，窗口的布局为null，在窗口中有两个按钮，单击一个按钮让另一个按钮移动。**

答: import java.awt.\*;

import java.awt.event.\*;

class WindowMove extends Frame implements ActionListener

{

Button button1,button2;

WindowMove(String s)

{

super(s);

setLayout(null);

button1=new Button("我让它横向走动");

button2=new Button("我让它纵向走动");

button1.setBackground(Color.blue);

button2.setBackground(Color.green);

button1.addActionListener(this);

button2.addActionListener(this);

button1.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

button2.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

add(button1);

add(button2);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

button1.setBounds(20,80,100,30);

button2.setBounds(100,180,100,30);

setBounds(100,100,500,500);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

Rectangle rect1=button1.getBounds();

int x1=(int)rect1.getX();

int y1=(int)rect1.getY();

Rectangle rect2=button2.getBounds();

int x2=(int)rect2.getX();

int y2=(int)rect2.getY();

if(e.getSource()==button1)

{

x2=x2+5;

button2.setLocation(x2,y2);

}

else if(e.getSource()==button2)

{

y1=y1+5;

button1.setLocation(x1,y1);

}

}

}

public class Test

{

public static void main(String args[])

{

new WindowMove("移动");

}

}

**13.编写应用程序，有一个标题为“改变颜色”的窗口，窗口的布局为null，在窗口中有3个按钮和一个画布，3个按钮的颜色分别是红、绿、蓝。单击相应的按钮，画布绘制相应颜色的圆。**

答: import java.awt.\*;

import java.awt.event.\*;

class WindowChangeColor extends Frame implements ActionListener

{

Button buttonRed,buttonGreen,buttonBlue;

Mycanvas canvas;

WindowChangeColor(String s)

{

super(s);

setLayout(null);

buttonRed=new Button("红色");

buttonGreen=new Button("绿色");

buttonBlue=new Button("蓝色");

canvas=new Mycanvas();

buttonRed.setBackground(Color.red);

buttonGreen.setBackground(Color.green);

buttonBlue.setBackground(Color.blue);

add(canvas);

canvas.setBounds(10,10,150,150);

add(buttonRed);

buttonRed.setBounds(10,170,50,30);

add(buttonGreen);

buttonGreen.setBounds(70,170,50,30);

add(buttonBlue);

buttonBlue.setBounds(130,170,50,30);

buttonRed.addActionListener(this);

buttonGreen.addActionListener(this);

buttonBlue.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

setBounds(100,100,200,250);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==buttonRed)

{

canvas.setX(1);

canvas.repaint();

}

else if(e.getSource()==buttonGreen)

{

canvas.setX(2);

canvas.repaint();

}

else if(e.getSource()==buttonBlue)

{

canvas.setX(3);

canvas.repaint();

}

}

}

class Mycanvas extends Canvas

{

int x=0;

Mycanvas()

{

setBackground(Color.white);

}

public void setX(int y)

{

x=y;

}

public void paint(Graphics g)

{

switch(x){

case 1:

g.setColor(Color.red);

break;

case 2:

g.setColor(Color.green);

break;

case 3:

g.setColor(Color.blue);

break;

default:

g.setColor(Color.white);

}

g.drawOval(65,65,50,50);

}

}

public class Test

{

public static void main(String args[])

{

new WindowChangeColor("改变颜色");

}

}

**14.编写应用程序，测试Cursor类中表示鼠标形状的静态常量。**

答: import java.awt.\*;

import java.awt.event.\*;

class WindowCursor extends Frame implements ActionListener

{

Button button,button1;

TextField text;

int n=-1;

WindowCursor()

{

button=new Button("单击我");

text=new TextField("将鼠标放在上面的按钮上看形状");

button1=new Button("看看你鼠标的形状");

add(button,BorderLayout.NORTH);

add(button1,BorderLayout.CENTER);

add(text,BorderLayout.SOUTH);

button.setBackground(Color.cyan);

button1.setBackground(Color.pink);

button.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

setBounds(100,100,190,150);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

n=(n+1)%10;

switch(n)

{

case 0:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

break;

case 1:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.CROSSHAIR\_CURSOR));

break;

case 2:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.TEXT\_CURSOR));

break;

case 3:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.WAIT\_CURSOR));

break;

case 4:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.N\_RESIZE\_CURSOR));

break;

case 5:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.NW\_RESIZE\_CURSOR));

break;

case 6:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.NE\_RESIZE\_CURSOR));

break;

case 7:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.W\_RESIZE\_CURSOR));

break;

case 8:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.MOVE\_CURSOR));

break;

case 9:

button1.setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT\_CURSOR));

break;

}

}

}

public class Test

{

public static void main(String args[])

{

new WindowCursor();

}

}

**15.改进本章例子7.30，当释放鼠标键时，如果当前组件和其他组件相交，就将其他组件设置为不可见状态。**

答: import java.awt.\*;

import java.awt.event.\*;

import javax.swing.SwingUtilities;

class Win extends Frame implements MouseListener,MouseMotionListener

{

Button button;

TextField text;

int x,y;

boolean move=false;

Win()

{

button=new Button("用鼠标拖动我");

text=new TextField("用鼠标拖动我",8);

text.setBackground(Color.green);

button.setBackground(Color.blue);

button.addMouseListener(this);

button.addMouseMotionListener(this);

text.addMouseListener(this);

text.addMouseMotionListener(this);

addMouseMotionListener(this);

setLayout(new FlowLayout());

add(button);

add(text);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

setBounds(10,10,350,300);

setVisible(true);

validate();

}

public void mousePressed(MouseEvent e){}

public void mouseReleased(MouseEvent e)

{

move=false;

Rectangle rect=button.getBounds();

int x=(int)rect.getX();

int y=(int)rect.getY();

if(rect.intersects(text.getBounds()))

{

if(e.getSource()==button)

{

text.setVisible(false);

}

else if(e.getSource()==text)

{

button.setVisible(false);

}

}

}

public void mouseEntered(MouseEvent e){}

public void mouseExited(MouseEvent e){}

public void mouseClicked(MouseEvent e){}

public void mouseMoved(MouseEvent e){}

public void mouseDragged(MouseEvent e)

{

Component com=null;

if(e.getSource() instanceof Component)

{

com=(Component)e.getSource();

if(com!=this)

move=true;

e=SwingUtilities.convertMouseEvent(com,e,this);

if(move)

{

x=e.getX();

y=e.getY();

int w=com.getSize().width,

h=com.getSize().height;

com.setLocation(x-w/2,y-h/2);

}

}

}

}

public class Test

{

public static void main(String args[])

{

Win win=new Win();

}

}

**16.进一步改进本章例子7.33，要求一个按钮在移动时，不允许和其他按钮相交。**

答: import java.awt.\*;

import java.awt.event.\*;

class Win extends Frame implements KeyListener

{

Button b[]=new Button[8];

int x,y;

Win()

{

setLayout(new FlowLayout());

for(int i=0;i<8;i++)

{

b[i]=new Button(""+i);

b[i].addKeyListener(this);

add(b[i]);

}

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

setBounds(10,10,300,300);

setVisible(true);

validate();

}

public void keyPressed(KeyEvent e)

{

boolean move=false;

Button button=(Button)e.getSource();

Rectangle buttonrect=button.getBounds();

String str=button.getLabel();

int xx=Integer.parseInt(str);

x=button.getBounds().x;

y=button.getBounds().y;

if(e.getKeyCode()==KeyEvent.VK\_UP)

{

y=y-2;

if(y<=0) y=0;

}

else if(e.getKeyCode()==KeyEvent.VK\_DOWN)

{

y=y+2;

if(y>=300) y=300;

}

else if(e.getKeyCode()==KeyEvent.VK\_LEFT)

{

x=x-2;

if(x<=0) x=0;

}

else if(e.getKeyCode()==KeyEvent.VK\_RIGHT)

{

x=x+2;

if(x>=300) x=300;

}

buttonrect.setLocation(x,y);

for(int k=0;k<8;k++)

{

Rectangle rect=b[k].getBounds();

if((buttonrect.intersects(rect))&&(xx!=k))

{

move=true;

}

}

if(move==false)

{

button.setLocation(x,y);

}

}

public void keyTyped(KeyEvent e){}

public void keyReleased(KeyEvent e){}

}

public class Test

{

public static void main(String args[])

{

Win win=new Win();

}

}

**1. 编写一个应用程序，用户可以在一个文本框里输入数字字符，按Enter键后将数字放入一个文本区。当输入的数字大于1000时，弹出一个有模式的对话框，提示用户数字已经大于1000，是否继续将该数字放入文本区。**

答: import java.awt.event.\*;

import java.awt.\*;

import javax.swing.JOptionPane;

class WindowSaveNumber extends Frame implements ActionListener

{

TextField textInputNumber;

TextArea textSaveNumber;

WindowSaveNumber()

{

textInputNumber=new TextField(15);

textSaveNumber=new TextArea();

add(textInputNumber,BorderLayout.NORTH);

add(textSaveNumber,BorderLayout.CENTER);

textInputNumber.addActionListener(this);

setBounds(100,100,400,300);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

}

public void actionPerformed(ActionEvent e)

{

String s=textInputNumber.getText();

int num=Integer.parseInt(s);

if(num<=1000)

{

textSaveNumber.append('\n'+s);

textInputNumber.setText(null);

}

else

{

int n=JOptionPane.showConfirmDialog(this,"数字已经大于1000 是否继续将该数字放入文本区","确认对话框",JOptionPane.YES\_NO\_OPTION);

if(n==JOptionPane.YES\_OPTION)

{

textSaveNumber.append('\n'+s);

}

else if(n==JOptionPane.NO\_OPTION)

{

textInputNumber.setText(null);

}

}

}

}

public class Test

{

public static void main(String args[])

{

new WindowSaveNumber();

}

}

**2. 参考Windows平台的NotePad，编写一个简单的“记事本”程序。**

答: import java.awt.\*;

import java.awt.event.\*;

import javax.swing.JOptionPane;

class Notepad extends Frame implements ActionListener

{

MenuBar menubar;

Menu menu1,menu2,menu3,menu5;

MenuItem item[]=new MenuItem[23];

TextArea text;

FileDialog filedialog\_save;

FileDialog filedialog\_load;

Notepad(String s)

{

super(s);

menubar=new MenuBar();

menu1=new Menu("文件");

menu2=new Menu("编辑");

menu3=new Menu("格式");

menu5=new Menu("帮助");

item[0]=new MenuItem("新建");

item[0].setShortcut(new MenuShortcut(KeyEvent.VK\_N));

item[1]=new MenuItem("打开");

item[1].setShortcut(new MenuShortcut(KeyEvent.VK\_O));

item[3]=new MenuItem("保存");

item[3].setShortcut(new MenuShortcut(KeyEvent.VK\_S));

item[4]=new MenuItem("页面设置");

item[5]=new MenuItem("打印");

item[5].setShortcut(new MenuShortcut(KeyEvent.VK\_P));

item[6]=new MenuItem("退出");

item[7]=new MenuItem("撤销");

item[7].setShortcut(new MenuShortcut(KeyEvent.VK\_Z));

item[8]=new MenuItem("剪切");

item[8].setShortcut(new MenuShortcut(KeyEvent.VK\_X));

item[9]=new MenuItem("复制");

item[9].setShortcut(new MenuShortcut(KeyEvent.VK\_C));

item[10]=new MenuItem("粘贴");

item[10].setShortcut(new MenuShortcut(KeyEvent.VK\_V));

item[11]=new MenuItem("删除");

item[12]=new MenuItem("查找");

item[13]=new MenuItem("查找下一个");

item[14]=new MenuItem("替换");

item[15]=new MenuItem("转到");

item[16]=new MenuItem("全选");

item[17]=new MenuItem("时间/日期");

item[18]=new CheckboxMenuItem("自动换行");

item[21]=new MenuItem("帮助主题");

item[22]=new MenuItem("关于记事本");

text=new TextArea();

filedialog\_save=new FileDialog(this,"保存文件",FileDialog.SAVE);

filedialog\_load=new FileDialog(this,"打开文件",FileDialog.LOAD);

filedialog\_save.setVisible(false);

filedialog\_load.setVisible(false);

filedialog\_save.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

filedialog\_save.setVisible(false);

}

}

);

filedialog\_load.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

filedialog\_load.setVisible(false);

}

}

);

menu1.add(item[0]);

menu1.add(item[1]);

menu1.add(item[3]);

menu1.addSeparator();

menu1.add(item[4]);

menu1.add(item[5]);

menu1.addSeparator();

menu1.add(item[6]);

menu2.add(item[7]);

menu2.addSeparator();

menu2.add(item[8]);

menu2.add(item[9]);

menu2.add(item[10]);

menu2.add(item[11]);

menu2.addSeparator();

menu2.add(item[12]);

menu2.add(item[13]);

menu2.add(item[14]);

menu2.add(item[15]);

menu2.addSeparator();

menu2.add(item[16]);

menu2.add(item[17]);

menu3.add(item[18]);

menu5.add(item[21]);

menu5.addSeparator();

menu5.add(item[22]);

menubar.add(menu1);

menubar.add(menu2);

menubar.add(menu3);

menubar.add(menu5);

item[0].addActionListener(this);

item[1].addActionListener(this);

item[3].addActionListener(this);

item[6].addActionListener(this);

item[21].addActionListener(this);

item[22].addActionListener(this);

setMenuBar(menubar);

add(text,BorderLayout.CENTER);

setBounds(100,100,600,400);

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

}

);

setVisible(true);

validate();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==item[0]) //完成新建

{

text.setText(null);

}

else if(e.getSource()==item[1]) //完成打开

{

filedialog\_load.setVisible(true);

}

else if(e.getSource()==item[3]) //完成保存

{

filedialog\_save.setVisible(true);

}

else if(e.getSource()==item[6]) //完成退出

{

System.exit(0);

}

else if(e.getSource()==item[21])

{

JOptionPane.showMessageDialog(this,"此版本中有很多功能没有实现\n剩余功能正在开发中\n请等待下一版本","帮助主题",JOptionPane.INFORMATION\_MESSAGE);

}

else if(e.getSource()==item[22])

{

JOptionPane.showMessageDialog(this,"名称: 自己做的NotePad\n作者: 王财\n版本: NotePad 1.1\n运行平台: Windows平台","关于记事本",JOptionPane.INFORMATION\_MESSAGE);

}

}

}

public class Test

{

public static void main (String[] args) {

new Notepad("自己做的NotePad");

}

}

**11.参照例子9.11编写一个应用程序，有两个线程，一个负责模仿垂直上抛运动，另一个模仿45°的抛体运动。**

答: import java.awt.\*;

import java.awt.event.\*;

public class Test

{

public static void main(String args[]){

MyFrame frame=new MyFrame();

frame.setBounds(10,10,500,500);

frame.setVisible(true);

frame.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

});

}

}

class MyFrame extends Frame implements Runnable

{

Thread 红色球,蓝色球;

MyCanvas red,blue;

double t=0;

MyFrame()

{

红色球=new Thread(this);

蓝色球=new Thread(this);

red=new MyCanvas(Color.red);

blue=new MyCanvas(Color.blue);

setLayout(null);

add(red);

add(blue);

red.setLocation(60,100);

blue.setLocation(60,100);

红色球.start();

蓝色球.start();

}

public void run()

{

while(true)

{

t=t+0.2;

int v0=20;

if(t>20)

t=0;

if(Thread.currentThread()==红色球)

{

int x=60;

int h=300-((int)(v0\*t-1.0/2\*t\*t\*3.8)+60);

red.setLocation(x,h);

try{

红色球.sleep(50);

}

catch(InterruptedException e){}

}

else if(Thread.currentThread()==蓝色球)

{

int x=60+(int)(26\*t);

int h=300-((int)(v0\*t-1.0/2\*t\*t\*3.8)+60);

blue.setLocation(x,h);

try{

蓝色球.sleep(50);

}

catch(InterruptedException e){}

}

}

}

}

class MyCanvas extends Canvas

{

Color c;

MyCanvas(Color c)

{

setSize(20,20);

this.c=c;

}

public void paint(Graphics g)

{

g.setColor(c);

g.fillOval(0,0,20,20);

}

}

**12.参照例子9.14，模拟3个人排队买票，张某、李某和赵某买电影票，售票员只有3张五元的钱，电影票五元一张。张某拿二十元一张的人民币排在李某的前面买票，李某排在赵某的前面拿一张十元的人民币买票，赵某拿一张五元的人民币买票。**

答: import java.awt.\*;

import java.awt.event.\*;

public class Test

{

public static void main(String args[])

{

new MyFrame();

}

}

class MyFrame extends Frame

implements Runnable,ActionListener

{

售票员 王小姐;

Thread 张某,李某,赵某;

static TextArea text;

Button start=new Button("排队买票");

MyFrame()

{

王小姐=new 售票员(); //只有3张五元钱 电影票五元一张

张某=new Thread(this); //二十元排在最前面

李某=new Thread(this); //十元 排在中间

赵某=new Thread(this); //五元 排在最后

text=new TextArea(10,30);

start.addActionListener(this);

add(text,BorderLayout.CENTER);

add(start,BorderLayout.NORTH);

setVisible(true);

setSize(360,300);

validate();

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

});

}

public void actionPerformed(ActionEvent e)

{

try{

张某.start();

李某.start();

赵某.start();

}

catch(Exception exp){}

}

public void run()

{

if(Thread.currentThread()==张某)

{

王小姐.售票规则(20);

}

else if(Thread.currentThread()==李某)

{

王小姐.售票规则(10);

}

else if(Thread.currentThread()==赵某)

{

王小姐.售票规则(5);

}

}

}

class 售票员

{

int 五元钱个数=3,十元钱个数=0,二十元钱个数=0;

String s=null;

public synchronized void 售票规则(int money)

{

if(money==5)

{

五元钱个数=五元钱个数+1;

s="给您入场卷您的钱正好";

MyFrame.text.append("\n"+s);

}

else if(money==10)

{

while(五元钱个数<1)

{

try{

wait();

}

catch(InterruptedException e){}

}

五元钱个数=五元钱个数-1;

十元钱个数=十元钱个数+1;

s="给您入场卷"+" 您给我10元，找您5元";

MyFrame.text.append("\n"+s);

}

else if(money==20)

{

while((五元钱个数<4)&&(五元钱个数<1||十元钱个数<1))

{

try{

wait();

}

catch(InterruptedException e){}

}

if(五元钱个数>=4)

{

五元钱个数=五元钱个数-3;

二十元钱个数=二十元钱个数+1;

s="给您入场卷"+" 您给我20元，找您15元";

MyFrame.text.append("\n"+s);

}

else if(五元钱个数>=1&&十元钱个数>=1)

{

五元钱个数=五元钱个数-1;

十元钱个数=十元钱个数-1;

二十元钱个数=二十元钱个数+1;

s="给您入场卷"+"您给我20元，找您15元";

MyFrame.text.append("\n"+s);

}

}

notifyAll();

}

}

**13.参照例子9.9，要求有3个线程:student1、student2和teacher，其中student1准备睡10分钟后再开始上课，其中student2准备睡一小时后再开始上课。teacher在输出3句“上课”后，吵醒休眠的线程student1；student1被吵醒后，负责再吵醒休眠的线程student2。**

答: class InterruptSleep implements Runnable

{

Thread student1,student2,teacher;

InterruptSleep()

{

teacher=new Thread(this);

student1=new Thread(this);

student2=new Thread(this);

}

public void run()

{

if(Thread.currentThread()==student2)

{

try{

System.out.println("student2要睡一小时再听课，现在不听课");

Thread.sleep(1000\*60\*60);

}

catch(InterruptedException e)

{

System.out.println("student2还没睡够呢，但是被student1给叫醒了");

}

System.out.println("student2开始上课");

}

else if(Thread.currentThread()==student1)

{

try{

System.out.println("student1要睡10分钟再上课，现在不听课");

Thread.sleep(1000\*60\*10);

}

catch(InterruptedException e)

{

System.out.println("student1还没睡够呢，但是被teacher给叫醒了");

}

System.out.println("student1开始上课");

student2.interrupt();

}

else if(Thread.currentThread()==teacher)

{

for(int i=1;i<=3;i++)

{

System.out.println("teacher: 上课!");

try{

Thread.sleep(500);

}

catch(InterruptedException e){}

}

student1.interrupt();

}

}

}

public class Test

{

public static void main (String[] args) {

InterruptSleep a=new InterruptSleep();

a.student2.start();

a.student1.start();

a.teacher.start();

}

}

**14.参照例子9.19，编写一个Java应用程序，在主线程中再创建3个线程:“运货司机”、“装运工”和“仓库管理员”。要求线程“运货司机”占有CPU资源后立刻联合线程“装运工”，也就是让“运货司机”一直等到“装运工”完成工作才能开车，而“装运工” 占有CPU资源后立刻联合线程“仓库管理员”，也就是让“装运工” 一直等到“仓库管理员”打开仓库才能开始搬运货物。**

答: public class Test

{

public static void main(String args[]){

ThreadJoin a=new ThreadJoin();

a.运货司机.start();

a.装运工.start();

a.仓库管理员.start();

}

}

class ThreadJoin implements Runnable

{

Goods goods;

Thread 运货司机,装运工,仓库管理员;

ThreadJoin()

{

运货司机=new Thread(this);

装运工=new Thread(this);

仓库管理员=new Thread(this);

运货司机.setName("运货司机");

装运工.setName("装运工");

仓库管理员.setName("仓库管理员");

}

public void run()

{

if(Thread.currentThread()==运货司机)

{

System.out.println(运货司机.getName()+"等"+装运工.getName()+"装运货物");

try{

装运工.join();

}

catch(InterruptedException e){}

System.out.println(运货司机.getName()+"开始运输"+goods.name);

}

else if(Thread.currentThread()==装运工)

{

System.out.println(装运工.getName()+"等"+仓库管理员.getName()+"打开仓库");

try{

仓库管理员.join();

}

catch(InterruptedException e){}

System.out.println(装运工.getName()+"开始搬运"+goods.name);

}

else if(Thread.currentThread()==仓库管理员)

{

System.out.println(仓库管理员.getName()+"准备打开仓库,请等待......");

try{

仓库管理员.sleep(2000);

}

catch(InterruptedException e){}

goods=new Goods("一批货物");

System.out.println(仓库管理员.getName()+"已经打开仓库，可以搬运了");

}

}

}

class Goods

{

String name;

Goods(String name)

{

this.name=name;

}

}

**7.使用RandomAccessFile流将一个文本文件倒置输出**

import java.awt.Container;

import java.io.\*;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class daozhi {

public static void main(String args[]) {

RandomAccessFile in = null;

try {

in = new RandomAccessFile("d:/demo.txt", "rw");

long length = in.length(); //获取文件的长度

long i=length-1;

char ch;

String s="";

while(i>=0){

in.seek(i);//设置文件指针

ch = (char) in.read();

s += ch;

i--;

}

showText(s);

} catch (IOException e) {

}

}

public static void showText(String s){

JFrame f=new JFrame("SetText");//frame

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);//close

f.setBounds(0, 0, 300, 200);//location,sise

Container content=f.getContentPane();//panel

JPanel p=new JPanel();

JTextArea t=new JTextArea();//area

JScrollPane scrollPane=new JScrollPane(t);//scroll

t.setRows(1);//row

t.setColumns(10);//col

t.setText(s);//set text

p.add(scrollPane);

content.add(p);

f.setVisible(true);

}

}

**8. 使用java的输入输出流技术将一个文本文件的内容按行读出，每读出一行就顺序添加行号，并写入到另一个文件中。**

import java.io.\*;

public class CommontTest{

public static void main(String ar[]) throws IOException {

BufferedReader reader = null;

PrintWriter writer = null;

int line = 0;

try {

reader = new BufferedReader(new FileReader("a.txt"));

} catch (FileNotFoundException e) {

System.out.println("输入文件 a.txt 不存在");

System.exit(-1);

}

writer = new PrintWriter(new FileWriter("b.txt"));

while(reader.ready()){

line++;

writer.println(line+"\t"+reader.readLine());

}

writer.flush();

writer.close();

reader.close();

}

}