

大二的java实验

原创

风吹过的烟花 于 2018-05-16 18:43:48 发布 5520 收藏 51

分类专栏: [我的小天地](#) 文章标签: [java](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/zhang_liuchen/article/details/80341178

版权



[我的小天地](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

这里会附上我的仓库, 进去下载我的 [java实验_self](#), 这个我写的实验, [戳我到仓库地址](#)

java实验上到实验7, 包括实验7, 后边的没时间了, 感觉还是有时间的话, 自己学完吧, 希望这些能帮到你!!!

(我给出实验要求, 一样的可以参考我的。)

Java 上机实验指导

上机实践1 初识Java.....	3
实验1 一个简单的应用程序.....	3
实验2 一个简单的JavaApplet程序.....	3
实验3 联合编译.....	4
上机实践2 基本数据类型与控制语句.....	5
实验1 输出希腊字母表.....	5
实验2 回文数.....	6
实验3 猜数字游戏.....	8
上机实践3 类与对象.....	9
实验1 三角形、梯形和圆形的类封装.....	9
实验2 实例成员与类成员.....	12

实验3 使用package语句与import语句.....	14
上机实践4 继承与接口.....	16
实验1 继承.....	16
实验2 上转型对象.....	19
实验3 接口回调.....	21
上机实践5 字符串、时间与数字.....	23
实验1 String类的常用方法.....	23
实验2 比较日期的大小.....	24
实验3 处理大整数.....	25
上机实践6 组件及事件处理.....	26
实验1 算术测试.....	26
实验2 信号灯.....	29
实验3 布局与日历.....	31
上机实践7 组件及事件处理2.....	35
实验1 方程求根.....	35
实验2 字体对话框.....	38
实验3 英语单词拼写训练.....	41
上机实践8 多线程.....	45
实验1 汉字打字练习.....	45
实验2 旋转的行星.....	47
实验3 多线程接力.....	51
上机实践9 输入输出流.....	54
实验1 学读汉字.....	54
实验2 统计英文单词字.....	57
实验2 读取Zip文件.....	61
上机实践10 Java 中的网络编程.....	62
实验1 读取服务器端文件.....	62
实验2 使用套接字读取服务器端对象.....	64
实验3 基于UDP的图像传输.....	67
上机实践11 数据结构.....	71
实验1 扫雷小游戏.....	71

实验2 排序与查找.....	76
实验3 使用TreeSet排序.....	77
上机实践12 java Swing.....	79
实验1 JLayeredPane分层窗格.....	79
实验2 使用表格显示日历.....	80
实验3 多文档界面 (MDI)	83

上机实践1 初识Java

实验1 一个简单的应用程序

Hello.java

```
public class Hello
{
    public static void main (String args[ ])
    {
        【代码1】 //命令行窗口输出"你好，很高兴学习Java"
        A a=new A();
        a.fA();
    }
}
class A
{
    void fA()
    {
        【代码2】 //命令行窗口输出"We are students"
    }
}
```

实验2 一个简单的Java Applet程序

1. 答案:

【代码1】： g.drawString("这是一个Java Applet 程序",10,30);

【代码2】： g.drawString("我改变了字体",20,50);

2. 模板代码

FirstApplet.java

```
import java.applet.*;
import java.awt.*;

public class FirstApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.blue);
        【代码1】 //在Java Applet中绘制一行文字：“这是一个Java Applet 程序”
        g.setColor(Color.red);
        g.setFont(new Font("宋体",Font.BOLD,36));
        【代码2】 //在Java Applet中绘制一行文字：“我改变了字体”
    }
}
```

实验3 联合编译

模板1: Hello.java

```
public class MainClass
{
    public static void main (String args[ ])
    {
        【代码1】 //命令行窗口输出"你好，只需编译我"
        A a=new A();
        a.fA();
        B b=new B();
        b.fB();
    }
}
```

模板2: A.java

```
public class A
{
```

```
void fA()
{
    【代码2】 //命令行窗口输出"I am A"
}
}
```

模板3：B.java

```
publicclass B
{
    void fB()
    {
        【代码3】 //命令行窗口输出"I am B"
    }
}
```

模板4：C.java

```
publicclass C
{
    void fC()
    {
        【代码4】 //命令行窗口输出"I am C"
    }
}
```

上机实践2 基本数据类型与控制语句

实验1 输出希腊字母表

GreekAlphabet.java

```
public class GreekAlphabet
{
    public static void main (String args[ ])
    {
        int startPosition=0,endPosition=0;
        char cStart='α',cEnd='ω';
        【代码1】 //cStart做int型转换据运算，并将结果赋值给startPosition
    }
}
```

【代码2】 //cEnd做int型转换运算，并将结果赋值给endPosition

```
System.out.println("希腊字母'\alpha'在unicode表中的顺序位置:"+(int)c);
```

```
System.out.println("希腊字母表: ");
```

```
for(int i=startPosition;i<=endPosition;i++)
```

```
{
```

```
    char c='\0';
```

【代码3】 //i做char型转换运算，并将结果赋值给c

```
    System.out.print(" "+c);
```

```
    if((i-startPosition+1)%10==0)
```

```
        System.out.println("");
```

```
    }
```

```
}
```

```
}
```

实验2 回文数

Number.java

```
import javax.swing.JOptionPane;
```

```
public class Number
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        int number=0,d5,d4,d3,d2,d1;
```

```
        Stringstr=JOptionPane.showInputDialog("输入一个1至99999之间的数");
```

```
        number=Integer.parseInt(str);
```

```
        if(【代码1】) //判断number在1至99999之间的条件
```

```
        {
```

```
            【代码2】 //计算number的最高位（万位）d5
```

```
            【代码3】 //计算number的千位d4
```

```
            【代码4】 //计算number的百位d3
```

```
            d2=number%100/10;
```

```
            d1=number%10;
```

```
            if(【代码5】) //判断number是5位数的条件
```

```
{
System.out.println(number+"是5位数");
if(【代码6】) //判断number是回文数的条件
{
    System.out.println(number+"是回文数");
}
else
{
    System.out.println(number+"不是回文数");
}
}
else if(【代码7】) //判断number是4位数的条件
{
    System.out.println(number+"是4位数");
    if(【代码8】) //判断number是回文数的条件码
    {
        System.out.println(number+"是回文数");
    }
    else
    {
        System.out.println(number+"不是回文数");
    }
}
else if(【代码9】) //判断number是3位数的条件
{
    System.out.println(number+"是3位数");
    if(【代码10】) //判断number是回文数的条件
    {
        System.out.println(number+"是回文数");
    }
    else
```

```

        {
            System.out.println(number+"不是回文数");
        }
    }
else if(d2!=0)
    {
        System.out.println(number+"是2位数");
        if(d1==d2)
            {
                System.out.println(number+"是回文数");
            }
        else
            {
                System.out.println(number+"不是回文数");
            }
    }
else if(d1!=0)
    {
        System.out.println(number+"是1位数");
        System.out.println(number+"是回文数");
    }
}
else
    {
        System.out.printf("\n%d不在1至99999之间",number);
    }
}
}

```

实验3 猜数字游戏

GuessNumber.java

```
import javax.swing.JOptionPane;
```



```

public class GuessNumber
{
    public static void main (String args[ ])
    {
        System.out.println("给你一个1至100之间的整数,请猜测这个数");
        int realNumber=(int)(Math.random()*100)+1;
        int yourGuess=0;
        String str=JOptionPane.showInputDialog("输入您的猜测:");
        yourGuess=Integer.parseInt(str);
        while(【代码1】) //循环条件
        {
            if(【代码2】) //条件代码
            {
                str=JOptionPane.showInputDialog("猜大了,再输入你的猜测:");
                yourGuess=Integer.parseInt(str);
            }
            else if(【代码3】) //条件代码
            {
                str=JOptionPane.showInputDialog("猜小了,再输入你的猜测:");
                yourGuess=Integer.parseInt(str);
            }
        }
        System.out.println("猜对了!");
    }
}

```

上机实践3 类与对象

实验1 三角形、梯形和圆形的类封装

AreaAndLength.java

```

class Trangle
{
    double sideA,sideB,sideC,area,length;

```

```

boolean boo;

public Trangle(double a,double b,double c)
{
    【代码1】 //参数a,b,c分别赋值给sideA,sideB,sideC
    if(【代码2】 )//a,b,c构成三角形的条件表达式
    {
        【代码3】 //给boo赋值。
    }
    else
    {
        【代码4】 //给boo赋值。
    }
}

double getLength()
{
    【代码5】 //方法体，要求计算出length的值并返回
}

public double getArea()
{
    if(boo)
    {
        double p=(sideA+sideB+sideC)/2.0;
        area=Math.sqrt(p*(p-sideA)*(p-sideB)*(p-sideC));
        return area;
    }
    else
    {
        System.out.println("不是一个三角形,不能计算面积");
        return 0;
    }
}
}

```

```

public void setABC(double a,double b,double c)
{
    【代码6】 //参数a,b,c分别赋值给sideA,sideB,sideC
    if(【代码7】 )//a,b,c构成三角形的条件表达式
    {
        【代码8】 //给boo赋值。
    }
    else
    {
        【代码9】 //给boo赋值。
    }
}
}

class Lader
{
    double above,bottom,height,area;
    Lader(double a,double b,double h)
    {
        【代码10】 //方法体，将参数a,b,c分别赋值给above,bottom,height
    }
    double getArea()
    {
        【代码11】 //方法体，要求计算出area返回
    }
}
}

```

```

class Circle
{
    double radius,area;
    Circle(double r)
    {

```

```

    【代码12】 //方法体
}
double getArea()
{
    【代码13】 //方法体，要求计算出area返回
}
double getLength()
{
    【代码14】 //getArea方法体的代码,要求计算出length返回
}
void setRadius(double newRadius)
{
    radius=newRadius;
}
double getRadius()
{
    return radius;
}
}
public class AreaAndLength
{
    public static void main(String args[])
    {
        double length,area;
        Circle circle=null;
        Trangle trangle;
        Lader lader;
        【代码15】 //创建对象circle
        【代码16】 //创建对象trangle。
        【代码17】 //创建对象lader
        【代码18】 // circle调用方法返回周长并赋值给length
    }
}

```

```
System.out.println("圆的周长:"+length);
```

【代码19】 // circle调用方法返回面积并赋值给area

```
System.out.println("圆的面积:"+area);
```

【代码20】 // trangle调用方法返回周长并赋值给length

```
System.out.println("三角形的周长:"+length);
```

【代码21】 // trangle调用方法返回面积并赋值给area

```
System.out.println("三角形的面积:"+area);
```

【代码22】 // lader调用方法返回面积并赋值给area

```
System.out.println("梯形的面积:"+area);
```

【代码23】 // trangle调用方法设置三个边，要求将三个边修改为12,34,1。

【代码24】 // trangle调用方法返回面积并赋值给area

```
System.out.println("三角形的面积:"+area);
```

【代码25】 // trangle调用方法返回周长并赋值给length

```
System.out.println("三角形的周长:"+length);
```

```
}
```

```
}
```

实验2 实例成员与类成员

Example.java

```
class A
```

```
{
```

【代码1】 //声明一个float型实例变量a

【代码2】 //声明一个float型类变量b,即static变量b

```
void setA(float a)
```

```
{
```

【代码3】 //将参数a的值赋值给成员变量a

```
}
```

```
void setB(float b)
```

```
{
```

【代码4】 //将参数b的值赋值给成员变量b

```
}
```

```
float getA()
```

```

{
    return a;
}

float getB()
{
    return b;
}

void inputA()
{
    System.out.println(a);
}

static void inputB()
{
    System.out.println(b);
}
}

public class Example
{
    public static void main(String args[])
    {
        【代码5】        //通过类名操作类变量b,并赋值100
        【代码6】        //通过类名调用方法inputB()
        A cat=new A();
        A dog=new A();

        【代码7】    //cat象调用方法setA(int a)将cat的成员a的值设置为200
        【代码8】    //cat调用方法setB(int b)将cat的成员b的值设置为400
        【代码9】    //dog象调用方法setA(int a)将dog的成员a的值设置为150
        【代码10】 //dog调用方法setB(int b)将dog的成员b的值设置为300
        【代码11】 //cat调用inputA()。
        【代码12】 //cat调用inputB()。
        【代码13】 //dog调用inputA()。
    }
}

```

【代码14】 //dog调用inputB()。

```
}  
}
```

实验3 使用package语句与import语句

SquareEquation.java

```
package tom.jiafei;  
  
public class SquareEquation  
{  
    double a,b,c;  
    double root1,root2;  
    boolean boo;  
    public SquareEquation(double a,double b,double c)  
    {  
        this.a=a;  
        this.b=b;  
        this.c=c;  
        if(a!=0)  
        {  
            boo=true;  
        }  
        else  
        {  
            boo=false;  
        }  
    }  
    public void getRoots()  
    {  
        if(boo)  
        {  
            System.out.println("是一元2次方程");  
            double disk=b*b-4*a*c;
```

```

if(disk>=0)
{
    root1=(-b+Math.sqrt(disk))/(2*a);
    root2=(-b-Math.sqrt(disk))/(2*a);
    System.out.printf("方程的根:%f,%f\n",root1,root2);
}
else
{
    System.out.printf("方程没有实根\n");
}

}
else
{
    System.out.println("不是一元2次方程");
}
}
public void setCoefficient(double a,double b,double c)
{
    this.a=a;
    this.b=b;
    this.c=c;
    if(a!=0)
    {
        boo=true;
    }
    else
    {
        boo=false;
    }
}
}

```



```
}
```

SunRise.java

```
import tom.jiafei.*;

class SunRise

{

    publicstatic void main(String args[ ])

    {

        SquareEquation equation=new SquareEquation(4,5,1);

        equation.getRoots();

        equation.setCoefficient(-3,4,5);

        equation.getRoots();

    }

}
```

上机实践4 继承与接口

实验1 继承

Example.java

```
class People

{

    protected double weight,height;

    public void speakHello()

    {

        System.out.println("yayawawa");

    }

    public void averageHeight()

    {

        height=173;

        System.out.println("average height:"+height);

    }

    public void averageWeight()

    {
```

```

weight=70;

System.out.println("averageweight:"+weight);
}
}
classChinaPeople extends People
{
【代码1】 //重写public voidspeakHello()方法，要求输出类似“你好，吃了吗”这样的
//汉语信息
【代码2】 //重写public voidaverageHeight()方法，要求输出类似
//“中国人的平均身高：168.78厘米”这样的汉语信息
【代码3】 //重写public voidaverageWeight()方法，
//要求输出类似“中国人的平均体重：65公斤”这样的汉语信息
public void chinaGongfu()
{
【代码4】 //输出中国武术的信息，例如：“坐如钟,站如松,睡如弓”等
}
}
classAmericanPeople extends People
{
【代码5】 //重写public voidspeakHello()方法，要求输出类似
//“How do you do”这样的英语信息。
【代码6】 //重写public voidaverageHeight()方法
【代码7】 //重写public void averageWeight()方法
public void americanBoxing()
{
【代码8】 //输出拳击的信息，例如，“直拳”、“钩拳”等
}
}
classBeijingPeople extends ChinaPeople
{
【代码9】 //重写public voidspeakHello()方法，要求输出类似“您好”这样的汉语信息

```

【代码10】 //重写public voidaverageHeight()方法

【代码11】 //重写public void averageWeight()方法

```
public void beijingOpera()
```

```
{
```

```
    【代码12】 //输出京剧的信息
```

```
}
```

```
}
```

```
public classExample
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        ChinaPeople chinaPeople=newChinaPeople();
```

```
        AmericanPeople americanPeople=newAmericanPeople();
```

```
        BeijingPeople beijingPeople=newBeijingPeople();
```

```
        chinaPeople.speakHello();
```

```
        americanPeople.speakHello();
```

```
        beijingPeople.speakHello();
```

```
        chinaPeople.averageHeight();
```

```
        americanPeople.averageHeight();
```

```
        beijingPeople.averageHeight();
```

```
        chinaPeople.averageWeight();
```

```
        americanPeople.averageWeight();
```

```
        beijingPeople.averageWeight();
```

```
        chinaPeople.chinaGongfu();
```

```
        americanPeople.americanBoxing();
```

```
        beijingPeople.beijingOpera() ;
```

```
        beijingPeople.chinaGongfu();
```

```
    }
```

```
}
```

实验2 上转型对象

HardWork.java

```
abstract class Employee
{
    public abstract double earnings();
}
class YearWorker extends Employee
{
    【代码1】 //重写earnings()方法
}
class MonthWorker extends Employee
{
    【代码2】 //重写earnings()方法。
}
class WeekWorker extends Employee
{
    【代码3】 //重写earnings()方法。
}
class Company
{
    Employee[] employee;
    double salaries=0;
    Company(Employee[] employee)
    {
        this.employee=employee;
    }
    public double salariesPay()
    {
        salaries=0;
        【代码4】 //计算salaries。
        return salaries;
    }
}
```

```

public class HardWork
{
    public static void main(String args[])
    {
        Employee[] employee=new Employee[20];
        for(int i=0;i<employee.length;i++)
        {
            if(i%3==0)
                employee[i]=new WeekWorker();
            else if(i%3==1)
                employee[i]=new MonthWorker();
            else if(i%3==2)
                employee[i]=new YearWorker();
        }
        Company company=new Company(employee);
        System.out.println("公司年工资总额:"+company.salariesPay());
    }
}

```

实验 3 接口回调

Road.java

```

interface ComputerWeight
{
    public double computeWeight();
}

class Television implementsComputerWeight
{
    【代码1】 //实现computeWeight()方法。
}

class Computer implements ComputerWeight
{
    【代码2】 //实现computeWeight()方法。
}

```

```
class WashMachine implements ComputerWeight
```

```
{ 【代码3】 //实现computeWeight()方法。
```

```
}
```

```
class Car
```

```
{ ComputerWeight[] goods;
```

```
double totalWeights=0;
```

```
Car(ComputerWeight[] goods)
```

```
{
```

```
    this.goods=goods;
```

```
}
```

```
public double getTotalWeights()
```

```
{
```

```
    totalWeights=0;
```

```
    【代码4】 //计算totalWeights
```

```
    return totalWeights;
```

```
}
```

```
}
```

```
public class Road
```

```
{
```

```
public static void main(String args[])
```

```
{ ComputerWeight[] goodsOne=new ComputerWeight[50],
```

```
        goodsTwo=new ComputerWeight[22] ;
```

```
for(int i=0;i<goodsOne.length;i++)
```

```
{ if(i%3==0)
```

```
    goodsOne[i]=new Television();
```

```
else if(i%3==1)
```

```
    goodsOne[i]=new Computer();
```

```
else if(i%3==2)
```

```
    goodsOne[i]=new WashMachine();
```

```
}
```

```
for(int i=0;i<goodsTwo.length;i++)
```

```

    { if(i%3==0)
        goodsTwo[i]=new Television();
    else if(i%3==1)
        goodsTwo[i]=new Computer();
    else if(i%3==2)
        goodsTwo[i]=new WashMachine();
    }
    Car 大货车=new Car(goodsOne);
    System.out.println("大货车装载的货物重量:"+大货车.getTotalWeights());
    Car 小货车=new Car(goodsTwo);
    System.out.println("小货车装载的货物重量:"+小货车.getTotalWeights());
}
}

```

上机实践5 字符串、时间与数字

实验1 String类的常用方法

模板代码

StringExample.java

```

class StringExample
{
    public static void main(String args[])
    {
        String s1=new String("you are a student"),
            s2=new String("how areyou");

        if(【代码1】) // 使用equals方法判断s1与s2是否相同
        {
            System.out.println("s1与s2相同");
        }
        else
        {
            System.out.println("s1与s2不相同");
        }

        String s3=newString("22030219851022024");
    }
}

```

```

if(【代码2】) //判断s3的前缀是否是“220302”。
{
    System.out.println("吉林省的身份证");
}

String s4=new String("你"),
    s5=new String("我");
if(【代码3】)//按着字典序s4大于s5的表达式。
{
    System.out.println("按字典序s4大于s5");
}
else
{
    System.out.println("按字典序s4小于s5");
}

int position=0;

Stringpath="c:\\java\\jsp\\A.java";

position=【代码5】 //获取path中最后出现目录分隔符号的位置
System.out.println("c:\\java\\jsp\\A.java中最后出现\\的位置:"+position);

String fileName=【代码6】 //获取path中“A.java”子字符串。
System.out.println("c:\\java\\jsp\\A.java中含有的文件名:"+fileName);

String s6=new String("100"),
    s7=newString("123.678");

int n1=【代码7】 //将s6转化成int型数据。

double n2=【代码8】 //将s7转化成double型数据。

double m=n1+n2;

System.out.println(m);

String s8=【代码9】 //String调用valueOf(int n)方法将m转化为字符串对象
position=s8.indexOf(".");

String temp=s8.substring(position+1);

System.out.println("数字"+m+"有"+temp.length()+"位小数");

String s9=newString("ABCDEF");

```



```

char a[]= 【代码10】 //将s8存放到数组a中。
for(int i=a.length-1;i>=0;i--)
{
    System.out.print(""+a[i]);
}
}
}

```

实验2 比较日期的大小

模板代码

DateExample

```

import java.util.*;
import javax.swing.JOptionPane;
public class DateExample
{
    public static void main(String args[ ])
    {
        Stringstr=JOptionPane.showInputDialog("输入第一个日期的年份:");
        int yearOne=Integer.parseInt(str);
        str=JOptionPane.showInputDialog("输入该年的月份:");
        int monthOne=Integer.parseInt(str);
        str=JOptionPane.showInputDialog("输入该月份的日期:");
        int dayOne=Integer.parseInt(str);
        str=JOptionPane.showInputDialog("输入第二个日期的年份:");
        int yearTwo=Integer.parseInt(str);
        str=JOptionPane.showInputDialog("输入该年的月份:");
        int monthTwo=Integer.parseInt(str);
        str=JOptionPane.showInputDialog("输入该月份的日期:");
        int dayTwo=Integer.parseInt(str);
        Calendar calendar= 【代码1】 //初始化日历对象
        【代码2】 //将calendar的时间设置为yearOne年monthOne月dayOne日
        long timeOne= 【代码3】 //calendar表示的时间转换成毫秒
    }
}

```

```

【代码4】          //将calendar的时间设置为yearTwo年monthTwo月dayTwo日

long timeTwo= 【代码5】  //calendar表示的时间转换成毫秒。

Date date1= 【代码6】    // 用timeOne做参数构造date1

Date date2= 【代码7】    // 用timeTwo做参数构造date2

if(date2.equals(date1))

{

    System.out.println("两个日期的年、月、日完全相同");

}

else if(date2.after(date1))

{

    System.out.println("您输入的第二个日期大于第一个日期");

}

else if(date2.before(date1))

{

    System.out.println("您输入的第二个日期小于第一个日期");

}

long days= 【代码8】 //计算两个日期相隔天数

System.out.println(yearOne+"年"+monthOne+"月"+dayOne+"日和"

    +yearTwo+"年"+monthTwo+"月"+dayTwo+"相隔"+days+"天");

}

}

```

实验3 处理大整数

模板代码

BigIntegerExample

```

import java.math.*;

class BigIntegerExample

{

    public static void main(String args[])

    { BigInteger n1=new BigInteger("987654321987654321987654321"),

        n2=new BigInteger("123456789123456789123456789"),

        result=null;

```

```

result= 【代码1】 //n1和n2做加法运算
System.out.println("和:"+result.toString());
result= 【代码2】 //n1和n2做减法运算
System.out.println("差:"+result.toString());
result= 【代码3】 //n1和n2做乘法运算
System.out.println("积:"+result.toString());
result= 【代码4】 //n1和n2做除法运算
System.out.println("商:"+result.toString());
BigInteger m=new BigInteger("1968957"),
        COUNT=newBigInteger("0"),
        ONE=newBigInteger("1"),
        TWO=newBigInteger("2");
System.out.println(m.toString()+"的因子有:");
for(BigInteger i=TWO;i.compareTo(m)<0;i=i.add(ONE))
{ if((n1.reminder(i).compareTo(BigInteger.ZERO))==0)
  { COUNT=COUNT.add(ONE);
    System.out.print(" "+i.toString());
  }
}
System.out.println("");
System.out.println(m.toString()+"一共有"+COUNT.toString()+"个因子");
}
}

```

上机实践6 组件及事件处理

实验1 算术测试

模板代码

Teacher.java

```

publicclass Teacher
{ int numberOne,numberTwo;
  String operator="";
  boolean right;

```

```
public int giveNumberOne(int n)
{ numberOne=(int)(Math.random()*n)+1;
  return numberOne;
}

public int giveNumberTwo(int n)
{ numberTwo=(int)(Math.random()*n)+1;
  return numberTwo;
}

public String giveOperator()
{ double d=Math.random();
  if(d>=0.5)
    operator="+";
  else
    operator="-";
  return operator;
}

public boolean getRight(int answer)
{ if(operator.equals("+"))
  { if(answer==numberOne+numberTwo)
    right=true;
    else
    right=false;
  }
  else if(operator.equals("-"))
  { if(answer==numberOne-numberTwo)
    right=true;
    else
    right=false;
  }
  return right;
}
```

```
}
```

ComputerFrame.java

```
import java.awt.*;
import java.awt.event.*;

public class ComputerFrame extends Frame implements ActionListener
{
    TextField textOne, textTwo, textResult;
    Button getProblem, giveAnswer;
    Label operatorLabel, message;
    Teacher teacher;

    ComputerFrame(String s)
    {
        super(s);
        teacher = new Teacher();
        setLayout(new FlowLayout());
        textOne = 【代码1】 //创建textOne,其可见字符长是10
        textTwo = 【代码2】 //创建textTwo,其可见字符长是10
        textResult = 【代码3】 //创建textResult,其可见字符长是10
        operatorLabel = new Label("+");
        message = new Label("你还没有回答呢");
        getProblem = new Button("获取题目");
        giveAnswer = new Button("确认答案");
        add(getProblem);
        add(textOne);
        add(operatorLabel);
        add(textTwo);
        add(new Label("="));
        add(textResult);
        add(giveAnswer);
        add(message);
        textResult.requestFocus();
        textOne.setEditable(false);
        textTwo.setEditable(false);
    }
}
```

【代码4】 //将当前窗口注册为getProblem的ActionEvent事件监视器

【代码5】 //将当前窗口注册为giveAnswer的ActionEvent事件监视器

【代码6】 //将当前窗口注册为textResult的ActionEvent事件监视器

```
setBounds(100,100,450,100);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

    { public void windowClosing(WindowEvent e)

        { System.exit(0);

          }

        }

    );

}

public void actionPerformed(ActionEvent e)

{ if( 【代码7】 ) //判断事件源是否是getProblem

    { int number1=teacher.giveNumberOne(100);

      int number2=teacher.giveNumberTwo(100);

      String operator=teacher.giveOperator();

      textOne.setText(""+number1);

      textTwo.setText(""+number2);

      operatorLabel.setText(operator);

      message.setText("请回答");

textResult.setText(null);

    }

if( 【代码8】 ) //判断事件源是否是giveAnswer

    { String answer=textResult.getText();

      try{

          intresult=Integer.parseInt(answer);

          if(teacher.getRight(result)==true)

              { message.setText("你回答正确");

                }

            }

        }
```

```

        else
        { message.setText("你回答错误");
        }
    }

    catch(NumberFormatException ex)
    { message.setText("请输入数字字符");
    }
}

textResult.requestFocus();

validate();

}
}

```

MainClass.java

```

publicclass MainClass
{public static void main(String args[])
{ ComputerFrame frame;
    frame= 【代码9】 //创建窗口，其标题为：算术测试
}
}

```

实验2 信号灯

2. 模板代码

SignalCanvas.java

```

importjava.awt.*;

publicclass SignalCanvas extends Canvas
{ int red,green,yellow,x,y,r;

    SignalCanvas()
    { setBackground(Color.white);
    }

    public void setRed(int r)
    { red=r;
    }
}

```

```

public void setGreen(int g)
{
    green=g;
}

public void setYellow(int y)
{
    yellow=y;
}

public void setPosition(int x,int y)
{
    this.x=x;
    this.y=y;
}

public void setRadius(int r)
{
    this.r=r;
}

public void paint(Graphics g)
{
    if(red==1)
    {
        g.setColor(Color.red);
    }
    else if(green==1)
    {
        g.setColor(Color.green);
    }
    else if(yellow==1)
    {
        g.setColor(Color.yellow);
    }
    g.fillOval(x,y,2*r,2*r);
}
}

```

SignalFrame.java

```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class SignalFrame extends Frame implements ItemListener

```



```

{ Choice choice;

SignalCanvas signal=null;

String itemRed="红灯",itemYellow="黄灯",itemGreen="绿灯";

public SignalFrame()

{ choice= 【代码1】      //创建choice

    【代码2】      //创建choice添加itemRed

    【代码3】      //创建choice添加itemYellow

    【代码4】      //创建choice添加itemGreen

    【代码5】 //将当前窗口注册为choice的ItemEvent事件监视器

add(choice, BorderLayout.NORTH);

try{ Class cs=Class.forName("SignalCanvas");

    signal=(SignalCanvas)cs.newInstance();

    add(signal, BorderLayout.CENTER);

    }

catch(Exception e)

    { add(new Label("您还没有编写SignalCanvas类"), BorderLayout.CENTER);

    }

setBounds(100,100,360,300);

setVisible(true);

validate();

addWindowListener(new WindowAdapter()

    { public void windowClosing(WindowEvent e)

        { System.exit(0);

        }

    }

    );

}

public void itemStateChanged(ItemEvent e)

{ String item= 【代码6】 // choice返回被选中的条目

int w=signal.getBounds().width;

int h=signal.getBounds().height;

```

```
int m=Math.min(w,h);
signal.setRadius(m/6);
if(item.equals(itemRed))
{ if(signal!=null)
  { signal.setRed(1);
    signal.setYellow(0);
    signal.setGreen(0);
    signal.setPosition(w/3,0);
    signal.repaint();
  }
}
else if(item.equals(itemYellow))
{ if(signal!=null)
  { signal.setRed(0);
    signal.setYellow(1);
    signal.setGreen(0);
    signal.setPosition(w/3,h/3);
    signal.repaint();
  }
}
else if(item.equals(itemGreen))
{ if(signal!=null)
  { signal.setRed(0);
    signal.setYellow(0);
    signal.setGreen(1);
    signal.setPosition(w/3,2*h/3);
    signal.repaint();
  }
}
}
```

SignalMainClass.java

```
public class SignalMainClass
{
    public static void main(String args[])
    {
        SignalFrame frame;
        frame=new SignalFrame();
        frame.setTitle("信号灯");
    }
}
```

实验3 布局与日历

CalendarBean.java

```
import java.util.Calendar;

public class CalendarBean
{
    String day[];
    int year=2005,month=0;
    public void setYear(int year)
    {
        this.year=year;
    }
    public int getYear()
    {
        return year;
    }
    public void setMonth(int month)
    {
        this.month=month;
    }
    public int getMonth()
    {
        return month;
    }
    public String[] getCalendar()
    {
        String a[]=new String[42];
        Calendar 日历=Calendar.getInstance();
        日历.set(year,month-1,1);
    }
}
```

```

int 星期几=日历.get(Calendar.DAY_OF_WEEK)-1;

int day=0;

if(month==1||month==3||month==5||month==7||month==8||month==10||month==12)
{
    day=31;
}

if(month==4||month==6||month==9||month==11)
{
    day=30;
}

if(month==2)
{
    if(((year%4==0)&&(year%100!=0))||(year%400==0))
    {
        day=29;
    }
    else
    {
        day=28;
    }
}

for(int i=星期几,n=1;i<星期几+day;i++)
{
    a[i]=String.valueOf(n);
    n++;
}

return a;
}
}

```

CalendarFrame.java

```

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class CalendarFrame extends Frame implements ActionListener
{
    Label labelDay[]=new Label[42];

```

```

Button titleName[]=new Button[7];

String name[]={"日","一","二","三","四","五","六"};

Button nextMonth,previousMonth;

int year=2006,month=10;

CalendarBean calendar;

Label showMessage=newLabel("",Label.CENTER);

public CalendarFrame()

{ Panel pCenter=new Panel();

    【代码1】 //将pCenter的布局设置为7行7列的GridLayout 布局。

    for(int i=0;i<7;i++)

    { titleName[i]=new Button(name[i]);

        【代码2】 //pCenter添加组件titleName[i]。

    }

    for(int i=0;i<42;i++)

    {

        labelDay[i]=newLabel("",Label.CENTER);

        【代码3】 //pCenter添加组件labelDay[i]。

    }

    calendar=new CalendarBean();

    calendar.setYear(year);

    calendar.setMonth(month);

    String day[]=calendar.getCalendar();

    for(int i=0;i<42;i++)

    { labelDay[i].setText(day[i]);

    }

    nextMonth=new Button("下月");

    previousMonth=new Button("上月");

    nextMonth.addActionListener(this);

    previousMonth.addActionListener(this);

    Panel pNorth=new Panel(),

        pSouth=new Panel();

```

```

pNorth.add(previousMonth);

pNorth.add(nextMonth);

pSouth.add(showMessage);

showMessage.setText("日历: "+calendar.getYear()+"年"+calendar.getMonth()+"月");

ScrollPane scrollPane=new ScrollPane();

scrollPane.add(pCenter);

    【代码4】 // 窗口添加scrollPane在中心区域

    【代码5】 // 窗口添加pNorth 在北面区域

    【代码6】 // 窗口添加pSouth 在南区域。

}

public void actionPerformed(ActionEvent e)

{ if(e.getSource()==nextMonth)

    { month=month+1;

        if(month>12)

            month=1;

        calendar.setMonth(month);

        String day[]=calendar.getCalendar();

        for(int i=0;i<42;i++)

            { labelDay[i].setText(day[i]);

                }

        }

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

        { month=month-1;

            if(month<1)

                month=12;

            calendar.setMonth(month);

            String day[]=calendar.getCalendar();

            for(int i=0;i<42;i++)

                { labelDay[i].setText(day[i]);

                    }

            }

    }
}

```

```
        showMessage.setText("日历: "+calendar.getYear()+"年"+calendar.getMonth()+"月" );  
    }  
}
```

CalendarMainClass.java

```
public class CalendarMainClass  
{ public static void main(String args[])  
    { CalendarFrame frame=new CalendarFrame();  
      frame.setBounds(100,100,360,300);  
      frame.setVisible(true);  
      frame.validate();  
      frame.addWindowListener(newjava.awt.event.WindowAdapter()  
          { public voidwindowClosing(java.awt.event.WindowEvent e)  
              { System.exit(0);  
              }  
            }  
          );  
    }  
}
```

上机实践7 组件及事件处理2

实验1 方程求根

SquareEquation.java

```
public classSquareEquation  
{ double a,b,c;  
  double root1,root2;  
  public void setA(double a)  
  { this.a=a;  
  }  
  public void setB(double b)
```

```

{ this.b=b;
}
public void setC(double c)
{this.c=c;
}
public double getRootOne() throws NoRealRootException, NoSquareEquationException
{ if(a!=0)
    { double disk=b*b-4*a*c;
      if(disk>=0)
        {root1=(-b+Math.sqrt(disk))/(2*a);
        }
      else
        { throw new NoRealRootException("没有实根");
        }
    }
  else
    {
      throw new NoRealRootException("不是二次方程");
    }
  return root1;
}
public double getRootTwo() throws NoRealRootException, NoSquareEquationException
{ if(a!=0)
    { double disk=b*b-4*a*c;
      if(disk>=0)
        {root2=(-b-Math.sqrt(disk))/(2*a);
        }
      else
        { throw new NoRealRootException("没有实根");
        }
    }
}

```



```

else
{
    throw new NoRealRootException("不是二次方程");
}
return root2;
}
}

```

```

class NoRealRootException extends Exception

```

```

{ String message;
    NoRealRootException(String s)
    { message=s;
    }
    public String getMessage()
    { return message;
    }
}

```

```

class NoSquareEquationException extends Exception

```

```

{ String message;
    NoSquareEquationException(String s)
    { message=s;
    }
    public String getMessage()
    { return message;
    }
}

```

EquationFrame.java

```

import java.awt.*;
import java.awt.event.*;
public class EquationFrame extends Frame implements ActionListener
{ SquareEquation equation;
    TextField textA, textB, textC;
}

```

```

TextArea showRoots;

Button controlButton;

public EquationFrame()
{ equation=new SquareEquation();

  textA=new TextField(8);

  textB=new TextField(8);

  textC=new TextField(8);

  showRoots=new TextArea();

  controlButton=new Button("确定");

  Panel pNorth=new Panel();

  pNorth.add(new Label("二次项系数:"));

  pNorth.add(textA);

  pNorth.add(new Label("一次项系数:"));

  pNorth.add(textB);

  pNorth.add(new Label("常数项系数:"));

  pNorth.add(textC);

  pNorth.add(controlButton);

  【代码1】 //当前窗口作为controlButton的ActionEvent事件的监视器

  add(pNorth,BorderLayout.NORTH);

  add(showRoots,BorderLayout.CENTER);

  setBounds(100,100,630,160);

  setVisible(true);

  validate();

  addWindowListener(new WindowAdapter()

    { public void windowClosing(WindowEvent e)

      { System.exit(0);

        }

      }

    );

}

public void actionPerformed(ActionEvent e)

```

```

{ try{
    double a=Double.parseDouble(【代码2】); //textA调用方法获取其中的文本
    double b=Double.parseDouble(【代码3】); //textB调用方法获取其中的文本
double c=Double.parseDouble(【代码4】); // textC调用方法获取其中的文本

    equation.setA(a);
    equation.setB(b);
    equation.setC(c);
    textA.setText(""+a);
    textB.setText(""+b);
    textC.setText(""+c);
    showRoots.append("\n 根: "+equation.getRootOne());
    showRoots.append(" 根: "+equation.getRootTwo());
}
catch(Exception ex)
{ showRoots.append("\n"+ex+"\n");
}
}
}

```

EquationMainClass.java

```

public class EquationMainClass
{ public static void main(String args[])
{ EquationFrame frame=new EquationFrame();
}
}

```

实验2 字体对话框

FontFamilyNames.java

```

import java.awt.GraphicsEnvironment;
public class FontFamilyNames
{ String fontName[];
    public String [] getFontName()
{ GraphicsEnvironment ge=GraphicsEnvironment.getLocalGraphicsEnvironment();

```

```

    fontName=ge.getAvailableFontFamilyNames();
    return fontName;
}
}

```

FontDialog.java

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.JLabel;
public class FontDialog extends Dialog implements ItemListener, ActionListener
{
    FontFamilyNames fontFamilyNames;
    int fontSize=38;
    String fontName;
    Choice fontNameList;
    JLabel label;
    Font font;
    Button yes, cancel;
    static int YES=1, NO=0;
    int state=-1;
    FontDialog(Frame f)
    {
        super(f);
        fontFamilyNames=new FontFamilyNames();
        【代码1】           //对话框设置为有模式
        yes=new Button("Yes");
        cancel=new Button("cancel");
        yes.addActionListener(this);
        cancel.addActionListener(this);
        label=new JLabel("hello, 奥运", JLabel.CENTER);
        fontNameList=new Choice();
        Stringname[]=fontFamilyNames.getFontName();
        for(int k=0;k<name.length;k++)
        {
            fontNameList.add(name[k]);
        }
    }
}

```

```

}

fontNameList.addItemListener(this);

add(fontNameList, BorderLayout.NORTH);

add(label, BorderLayout.CENTER);

Panel pSouth=new Panel();

pSouth.add(yes);

pSouth.add(cancel);

add(pSouth, BorderLayout.SOUTH);

setBounds(100,100,280,170);

addWindowListener(new WindowAdapter()

    {   public void windowClosing(WindowEvent e)

        { state=NO;

            setVisible(false);

        }

    }

);

validate();

}

public void itemStateChanged(ItemEvent e)

{fontName=(String)fontNameList.getSelectedItem();

font=newFont(fontName,Font.BOLD,fontSize);

label.setFont(font);

label.repaint();

validate();

}

public void actionPerformed(ActionEvent e)

{ if(e.getSource()==yes)

    { state=YES;

        【代码2】      //对话框设置为不可见

    }

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

```

```

    { state=NO;
        【代码3】    //对话框设置为不可见
    }
}

public int getState()
{ return state;
}

public Font getFont()
{ return font;
}
}

```

FrameHaveDialog.java

```

import java.awt.event.*;
import java.awt.*;
import javax.swing.JTextArea;

public class FrameHaveDialog extends Frame implements ActionListener
{
    JTextArea text;
    Button buttonFont;

    FrameHaveDialog()
    {
        buttonFont=new Button("设置字体");
        text=new JTextArea("Java 2实用教程（第三版）");
        buttonFont.addActionListener(this);
        add(buttonFont,BorderLayout.NORTH);
        add(text);
        setBounds(60,60,300,300);
        setVisible(true);
        validate();
        addWindowListener(new WindowAdapter()
            {
                public void windowClosing(WindowEvent e)
                {
                    System.exit(0);
                }
            }
        );
    }
}

```



```

for(int k=0;k<m;k++)
{
    int index=(int)(Math.random()*strBuffer.length());
    char c=strBuffer.charAt(index);
    str=str+c;
    strBuffer.deleteCharAt(index);
}
return str;
}
}

```

LetterLabel.java

```

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

public class LetterLabel extends Button implements FocusListener,MouseListener
{
    LetterLabel()
    {
        /*代码1*/ //将当前对象注册为自身的焦点视器
        /*代码2*/ //将当前对象注册为自身的标监视器
        setBackground(Color.cyan);
        setFont(new Font("",Font.BOLD,30));
    }

    public static LetterLabel[] getLetterLabel(int n)
    {
        LetterLabel a[]=new LetterLabel[n];
        for(int k=0;k<a.length;k++)
        {
            a[k]=new LetterLabel();
        }
        return a;
    }

    public void focusGained(FocusEvent e)
    {
        setBackground(Color.red);
    }

    public void focusLost(FocusEvent e)
    {
        setBackground(Color.cyan);
    }
}

```



```

}

public void mousePressed(MouseEvent e)
{ requestFocus();
}

public void setText(char c)
{ setLabel(""+c);
}

public void mouseReleased(MouseEvent e){}
public void mouseEntered(MouseEvent e) {}
public void mouseExited(MouseEvent e) {}
public void mouseClicked(MouseEvent e){}
}

```

SpellingWordFrame.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.Box;

public class SpellingWordFrame extends Frame implements KeyListener, ActionListener
{ TextField inputWord;
  Button button;
  LetterLabel label[];
  Panel northP, centerP;
  Box wordBox;

  String hintMessage="用鼠标单击字母，按左右箭头交换字母，将其排列成所输入的单词";
  Label messageLabel=new Label(hintMessage);
  String word="";

  SpellingWordFrame()
  { inputWord=new TextField(12);
    button=new Button("确定");
    button.addActionListener(this);
    inputWord.addActionListener(this);
    northP=new Panel();

```

```

northP.add(new Label("输入一个英文单词: "));
northP.add(inputWord);
northP.add(button);
centerP=new Panel();
wordBox=Box.createHorizontalBox();
centerP.add(wordBox);
add(northP,BorderLayout.NORTH);
add(centerP,BorderLayout.CENTER);
add(messaageLabel,BorderLayout.SOUTH);
setBounds(100,100,350,180);
setVisible(true);
validate();
addWindowListener(new WindowAdapter()
    { public void windowClosing(WindowEvent e)
        { System.exit(0);
        }
    }
);
}

public void actionPerformed(ActionEvent e)
{ word=inputWord.getText();
  int n=word.length();
  RondonString rondom=new RondonString();
  StringrandomWord=rondom.getRondonString(word);
  wordBox.removeAll();
  messaageLabel.setText(hintMessage);
  if(n>0)
  { label=LetterLabel.getLetterLabel(n);
    for(int k=0;k<label.length;k++)
    { 【代码3】 //将当前窗口注册为label[k]的键盘监视器
      label[k].setText(""+randomWord.charAt(k));
    }
  }
}

```

```

        wordBox.add(label[k]);
    }
    validate();
    inputWord.setText(null);
    label[0].requestFocus();
}
}

public void keyPressed(KeyEvent e)
{ LetterLabel sourceLabel=(LetterLabel)e.getSource();
    int index=-1;
    if( 【代码4】 ) //判断按下的是否是←键)
    { for(int k=0;k<label.length;k++)
        { if(label[k]==sourceLabel)
            { index=k;
                break;
            }
        }
    }
    if(index!=0)
    { String temp=label[index].getText();
        label[index].setText(label[index-1].getText());
        label[index-1].setText(temp);
    label[index-1].requestFocus();
    }
}

else if( 【代码5】 ) //判断按下的是否是→键
{ for(int k=0;k<label.length;k++)
    { if(label[k]==sourceLabel)
        { index=k;
            break;
        }
    }
}
}

```

```

        if(index!=label.length-1)
        { String temp=label[index].getText();
          label[index].setText(label[index+1].getText());
          label[index+1].setText(temp);
        label[index+1].requestFocus();
        }
    }
validate();
}
public void keyTyped(KeyEvent e){}
public void keyReleased(KeyEvent e)
{ String success="";
  for(int k=0;k<label.length;k++)
  { String str=label[k].getText();
    success=success+str;
  }
  if(success.equals(word))
  { messageLabel.setText("恭喜你，你成功了");
    for(int k=0;k<label.length;k++)
    {label[k].removeKeyListener(this);
      label[k].removeFocusListener(label[k]);
      label[k].setBackground(Color.green);
    }
    inputWord.requestFocus();
  }
}
}
}

```

WordMainClass.java

```

public class WordMainClass
{ public static void main(String args[])
  { new SpellingWordFrame();
  }
}

```

}

}



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)