

## 第一次实验课

1编写一个Java程序，用if-else语句判断某年份是否为闰年。

```
public class runnian{

    public static void main(String args[]){        int year
        =2015;

        if((year%4==0&&year%100!=0)||((year%400==0)))

        System.out.print(year+"是闰年");

        else

        System.out.print(year+"不是闰年");

    }

}
```

2编写一个Java程序在屏幕上输出1！+2！+3！+……+10！的和。（分别用for和while

来做）（1分）

```
int sum = 0;

for (int i = 1; i < 11; i++){

    int n = 0;

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

        n = n * j;

    }

    sum = sum + n;

}

System.out.println("1！+2！+3！+……+10！="+sum);

}
```

## 第二次 JAVA 实验

1、建立一个工程Lesson2\_3工程，里面包含两个文件

MyMain.java，Student.java。

MyMain.java放到app包中，Student.java放到

cn.swust.edu.student包中。且main函

数位于MyMain.java文件中。

2、在Student.java中，定义一个学生类。包括域“学

号”、班级、姓名、年龄；方法：获取

年龄、获取班级、获取姓名，修改年龄。

3、定义一个函数init（），实现对Student类的域初始化。

5

4、去掉init函数，通过构造函数实现Student类的域初始化

5、增加一个toString方法，将所有域全部打印出来。

6、在Main函数中，创建Student对象，该对象为自己。并将自己的信息全部打印出来。

```
public class Student {

    private String id;//学号

    private int classId;//班号

    private String name;//姓名

    private String sex;//性别

    private int age;//年龄

    //设置学生 id

    public void setId(String id) {

        this.id = id;

    }

    //获取学生 id

    public String getId() {

        return id;

    }

    //设置班级 id

    public void setClassId(int classId) {

        this.classId = classId;

    }

    //获取班级 ID

    public int getClassId() {

        return classId;

    }

    //设置年龄

    public void setAge(int age) {

        this.age = age;

    }

    //获取年龄

    public int getAge() {

        return age;

    }

    //设置姓名

    public void setName(String name){

        this.name = name;

    }

    //获取姓名

    public String getName() {

        return name;

    }

    //设置性别

    public void setSex(String sex) {

        this.sex = sex;

    }

}
```

//获取性别

```
public String getSex() {

    return sex;

}
```

//重写 toString()

```
public String toString() {

    return "学生号：" + id + " 班级号："

    + classId +

    " 姓名：" + name + " 性别："

    + sex + " 年龄：" + age;

}
```

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

    Student s = new Student();

    s.setId("201190609112");

    s.setClassId(111-1);

    s.setName("张星龙");

    s.setSex("男");

    s.setAge(21);

    System.out.println(s);//打印内容为重写

    toString()后的内容

}
```

```
1.    public class Chuangjian {

2.    public static void main(String args[]) {

3.        Student demoStudent = new Student();

4.        System.out.print("学生号：

        " + demoStudent.getStudentId() + " 班级

        号：" + demoStudent.getClassId() +

5.        " 姓名：

        " + demoStudent.getStudentName() + " 性

        别：

        " + demoStudent.getStudentSex() + " 年

        龄：" + demoStudent.getStudentAge() );

6.    }

7.    }

8.

9.    class Student {

10.        private String studentId="201190609112";

11.        private int classId=1111;

12.        private String studentName="张星龙";

13.        private char studentSex='男';

14.        private int studentAge=21;

15.

16.        String getStudentId() {

17.            return studentId;

18.        }
```

```

19.
20.  int getClassId() {
21.  return classId;
22.  }
23.
24.  char getStudentSex() {
25.  return studentSex;
26.  }
27.
28.  String getStudentName() {
29.  return studentName;
30.  }
31.
32.  int getStudentAge() {
33.  return studentAge;
34.  }
35.
36.  }

```

```

public void getarea() {
    double s = (a + b + c) / 2;
    area = Math.sqrt(s * (s - a) * (s - b) * (s - c));
    System.out.println("area=" + area);
}

public void cga(double a2) {
    a = a2;
    System.out.println("a=" + a);
}

public void cgb(double b2) {
    b = b2;
    System.out.println("b=" + b);
}

public void cgc(double c2) {
    c = c2;
    System.out.println("c=" + c);
}

public void triangle() {
    boolean t0 = true, t1 = false;
    double max = (a > b ? a : b) > c ? (a > b ? a : b) : c;
    if (max == a) {
        double s = b + c;
        if (max < s) {
            System.out.println(t0 + "能构成三角形");
        } else {
            System.out.println(t1 + "不能构成三角形");
        }
    }
    if (max == b) {
        double s = a + c;
        if (max < s) {
            System.out.println(t0 + "能构成三角形");
        } else {
            System.out.println(t1 + "不能构成三角形");
        }
    }
    if (max == c) {
        double s = b + a;
        if (max < s) {
            System.out.println(t0 + "能构成三角形");
        } else {
            System.out.println(t1 + "不能构成三角形");
        }
    }
}

package Circle;

public class Circle {
    double r,c,area;

```

```

static final double PI=3.14159;

public Circle(double r1){
    r=r1;
}

public void getc(){
    c=2*PI*r;
    System.out.println("r="+r+"周长="+c);
}

public void getCircle(){
    area=r*r*PI;
    System.out.println("圆 area="+area);
}

package Ladder;

public class Ladder {
    double a, b, h;

    public Ladder(double a1, double b1, double h1) {
        a = a1;
        b = b1;
        h = h1;
    }

    public void getArea() {
        double Area = ((a + b) * h) / 2;
        System.out.println("梯形 Area=" + Area);
    }
}

package Main;

import Circle.*;
import Ladder.*;
import Triangle.*;

public class Main {
    public static void main(String[] args) {
        Triangle p=new Triangle(3,3,6
    );
        p.triangle();
        p.getabc();
        p.getarea();
        p.cga(7);
        p.cgb(5);
        p.cgc(20);
        p.triangle();
        Ladder k=new Ladder(2,4,6);
        k.getArea();
        Circle m=new Circle(3);
        m.getc();
        m.getCircle();
    }
}

package Main;

```

### 第三次实验

一、编写一个Java应用程序，该程序中有3个类：

Triangle、Ladder和Circle，分别用来

刻画“三角形”、“梯形”和“圆形”。具体要求如下：

a) Triangle类具有类型为double的三个边，以及周长、

面积属性，Triangle类具有返回周

长、面积以及修改三个边的功能。另外，Triangle类还具

有一个boolean型的属性，该属

性用来判断三个属能否构成一个三角形。

b) Ladder类具有类型为double的上底、下底、高、面积属

性，具有返回面积的功能。

c) Circle类具有类型为double的半径、周长和面积属

性，具有返回周长、面积的功能

二、编写一个StringUtils类，实现十进制到八进制的转

换并输出。需要一个测试类加以测

试

```

package Triangle;

public class Triangle {
    double a, b, c, area;

    public Triangle(double a1, double b1, double c1) {
        a = a1;
        b = b1;
        c = c1;
    }

    public void getabc() {
        System.out.println("a=" + a + "b=" + b + "c=" + c);
    }
}

```

```

import Circle.*;

import Ladder.*;

import Trangle.*;

public class Main {

    public static void main(String[] args) {

        Trangle p=new Trangle(3,3,6

    );

    p.trangle();

    p.getabc();

    p.getarea();

    p.cga(7);

    p.cgb(5);

    p.cgc(20);

    p.trangle();

    Ladder k=new Ladder(2,4,6);

    k.getArea();

    Circle m=new Circle(3);

    m.getc();

    m.getCircle();

    }

}

```

#### 第四次实验

1. 阅读如下所示的3个Java类的定义，分析它们之间的关系，写出运行结果。

```

1 : class SuperClass {

2 : int x;

3 : SuperClass() {

4 : x=3;

5 : System.out.println("in SuperClass : x=" +x);

6 : }

7 : void doSomething() {

8 : System.out.println("in

SuperClass.doSomething()");

9 : }

11 : }

1 : class SubClass extends SuperClass {

2 : int x;

3 : SubClass() {

4 : super(); //调用父类的构造方法

5: x=5; //super( ) 要放在方法中的第一句

6 : System.out.println("in SubClass :x="+x);

7 : }

8 : void doSomething() {

9 : super.doSomething( ); //调用父类的方法

10 : System.out.println("in

SubClass.doSomething()");

11 : System.out.println("super.x="+super.x+"

sub.x="+x);

12 : }

```

```

14 : }

1 : import java.io.*;

2 : public class Inheritance {

3 : public static void main(String args[]) {

4 : SubClass subC=new SubClass();

5 : subC.doSomething();

6 : }

7 : }

2. 假定根据学生的3门学位课程的分数决定其是否可以

拿到学位，对于本科生，如果3

门课程的平均分数超过60分即表示通过，而对于研究

生，则需要平均分超过80分才能够

通过。根据上述要求，请完成以下Java类的设计：

(1) 设计一个基类Student描述学生的共同特征。

(2) 设计一个描述本科生的类Undergraduate，该类继承并

扩展Student类。

(3) 设计一个描述研究生的类Graduate，该类继承并

扩展Student类。

(4) 设计一个测试类StudentDemo，分别创建本科生

和研究生这两个类的对象，并输出

相关信息。

```

```

package eeee;

public class Student {

/**

 * @param args

 */

int total,arv;

Student(int x,int y,int z)

{

    total=x+y+z;

    arv=total/3;

}

int show()

{

    return arv;

}

public static void main(String[] args)

{

    StudentDemo sov=new StudentDemo();

    sov.m();

}

package eeee;

class Undergraduate extends Student{

int c1,c2,c3;

Undergraduate(int x,int y,int z)

{

    super(x,y,z);

    c1=x;

```

```

c2=y;

c3=z;

}

public boolean solve()

{

    if(super.show()<60)

        return false;

    else

        return true;

}

}

package eeee;

class Graduate extends Student{

int c1,c2,c3;

Graduate(int x,int y,int z)

{

    super(x,y,z);

    c1=x;

    c2=y;

    c3=z;

}

public boolean solve()

{

    if(super.show()<80)

        return false;

    else

        return true;

}

}

package eeee;

import java.util.Scanner;

class StudentDemo{

void m(){

Scanner in=new Scanner(System.in);

int x,y,z;

x=in.nextInt();

y=in.nextInt();

z=in.nextInt();

Undergraduate s1=new Undergraduate(x, y, z);

System.out.println("大学生三门课程分数 : "+s1.c1+"

"+s1.c2+" "+s1.c3);

if(s1.solve())

    System.out.println("及格");

else

    System.out.println("不及格");

x=in.nextInt();

y=in.nextInt();

z=in.nextInt();

Graduate s2=new Graduate(x, y, z);

```

```

System.out.println("研究生三门课程分数："+s2.c1+"
"+s2.c2+" "+s2.c3);

if(s2.solve())

    System.out.println("及格");

else

    System.out.println("不及格");

}

}

package eaes;

import java.io.*;

public class eae {

/**
 * @param args
 */

public static void main(String[] args) {

    // TODO Auto-generated method stub

    SubClass subC=new SubClass();

    subC.doSomething();

}

}

class SuperClass

{

int x;

SuperClass()

{

x=3;

    System.out.println("in SuperClass:x="+x);

}

}

void doSomething()

{

    System.out.println("in SuperClass.doSomething()");

}

}

class SubClass extends SuperClass{

int x;

    SubClass() {

// TODO Auto-generated constructor stub

super();

x=5;

        System.out.println("in Subclass:x="+x);

}

void doSomething()

{

super.doSomething();

System.out.println("in SubClass.doSomething()");

System.out.println("super x="+super.x+" sub.x="+x);

}

}

```

## 第五次实验

### 1．分析下列程序的输出结果。

```

interface Myface{

int getNext();

void setstart(intx);

}

class Add3 implements Myface{

int sta, val;

Add3(){

sta=0; val=0;

}

public int getNext(){

val+=3;

return val;

}

public void setstart(int i){

sta=i;

val=i;

}

}

public class Face{

public static void main(String args[]){

Add3 a=new Add3();

for(int i=0;i<5;i++){

System.out.println(a.getNext());

System.out.println();

a.setstart(10);

for(int i=0;i<5;i++){

System.out.println(a.getNext());

}

}

}

```

### 2．按照要求编写JavaApplication程序。

假定要为某个公司编写雇员工资支付程序。这个公司有各种类型的雇员（Employee），不

同类型的雇员按不同的方式支付工资：

**经理（Manager）**：每月获得一份固定的工资；

**销售人员（Salesman）**：在基本工资的基础上每月还有销售提成；

**一般工人（Worker）**：则按他每月工作的天数计算工资。

试用类的继承及相关机制描述这些功能需求，并编写一个JavaApplication程序，演示这

些类的用法。

```

abstract class Employee{

private String name;

public Employee(String name) {

this.name=name;

}

public String getName() {

```

```

return name;

}

public abstract double computeSalary();

}

class Manager extends Employee{

double monthSalary;

public Manager(String name,double monthSalary) {

super(name);

this.monthSalary=monthSalary;

}

public double computeSalary() {

return monthSalary;

}

}

class Salesman extends Employee{

double baseSalary;

int quantity;

double commision;

public Salesman(String name,double baseSalary,int quantity,double commision) {

super(name);

this.baseSalary=baseSalary;

this.quantity=quantity;

this.commision=commision;

}

public double computeSalary() {

return baseSalary+quantity*commision;

}

}

class Worker extends Employee{

int days;

double dailySalary;

public Worker(String name,int days,double dailySalary) {

super(name);

this.days=days;

this.dailySalary=dailySalary;

}

public double computeSalary() {

return days*dailySalary;

}

}

public class EmployeeDemo{

public static void main(String args[]) {

Manager m=new Manager("小明",8000);

Salesman s=new Salesman("小红",2000,20,80);

Worker w=new Worker("小黑",30,80);

System.out.println("经理"+m.getName()+"的工资为："+m.computeSalary());

System.out.println("销售人员"+s.getName()+"的工资

```

```
为:"+s.computeSalary());

    System.out.println("工人"+w.getName()+"的工资为："
+w.computeSalary());

}

}
```

## 第六次实验

### 1、使用类String类的分割split将字符串

“Solutionstoselectedexercisescanbe

foundintheelectronicdocumentTheThinkinginJavaAnn

otatedSolutionGuide,

availableforasmallfeefromBruceEckel”单词提取输出。

单词以空格或分割。

2、

2、设计一个类Student,类的属性有：姓名，学号，出生日期，性别，所在系等。并生成学

生类对象数组。按照学生的姓名将学生排序输出。使用

String类的compareTo方法。

3、设计一个程序计算 2010-05-01 日与系统当前日期相差的天数。

1.

```
public class EX1_1 {

    public static void main(String[] args) {

        String str1=new String("Solutions to selected exercises
can be found in the electronic document The Thinking in
Java Annotated Solution Guide,available for a small fee
from BruceEckel");

        String[] str2=str1.split(" |,");

        for(int i=0;i<str2.length;i++)

            System.out.println(str2[i]);

    }

}
```

2.

```
package aaasd;

public class Student {

    private String name;

    private int num;

    private String birth;

    private String sex;

    private String sdept;

    public Student(){

    }

    public Student(String name,int num,String birth,String
sex,String sdept){

        this.name=name;

        this.num=num;

        this.birth=birth;

        this.sex=sex;

        this.sdept=sdept;
```

```
}

    public String getName()

    {return name;

}

    public void setName(String name)

    {

        this.name=name;

    }

    public int getNum(){

        return num;

    }

    public void setNum(int num)

    {this.num=num;

    }

    public String getBirth()

    {

        return birth;

    }

    public void setBirth(String birth)

    { this.birth=birth;

    }

    public String getSex()

    {return sex;

    }

    }

    public void setSex(String sex)

    {

        this.sex=sex;

    }

    public String getSdept()

    {

        return sdept;

    }

    public void setSdept(String sdept)

    {

        this.sdept=sdept;

    }

    }

3.

package Hahaha;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class TimeToTime{

    public static void main(String[] args) throws

    ParseException{

        Date d1=new Date();

        SimpleDateFormat format=new
```

```
SimpleDateFormat("yyyy-MM-dd");

String s="2010-05-01";

Date d2=format.parse(s);

int days=(int)((d2.getTime()-

d1.getTime())/(1000*60*60*24));

System.out.print("2010-05-01 日与系统当前时间相差

"+(-days)+"天");

}

}
```

## 第七次实验

1、猜数字游戏。编写一个Java应用程序，实现功能：

(1) 程序随机分配给用户一个1~100之间的随机数

(2) 用户在输入对话框中输入自己的猜测

(3) 程序返回提示信息，提示信息分别是：“猜大了”、“猜小了”、“猜对了”

(4) 用户可根据提示信息再次输入猜测，直到提示信息是“猜对了”。

2.一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。编程思路 and 代码在附件。

备注：若一个自然数，它所有的真因子（即除了自身以外的约数）的和恰好等于它本身，这种数叫做完全数。简称“完数”

例如，

6=1+2+3

28=1+2+4+7+14

496=1+2+4+8+16+31+62+124+248

8128=1+2+4+8+16+32+64+127+254+508+1016+2032+4064

13

编写一个 Java 程序，输出任意数之间的所有完数。

1.

```
import java.util.Random;

import java.util.Scanner;

class JavaText{

    public static void main (String[]args){

        int num = new Random().nextInt(100)+1;

        Scanner in =new Scanner(System.in);

        int temp= in.nextInt();

        while(temp!=num){

            if(temp<num)

                System.out.println("猜小了");

            else

                System.out.println("猜大了");

            temp = in.nextInt();

        }

    }

}
```

2.这道题要注意 1 既不是完数，也不是非完数

```
import java.util.Scanner;

public class JavaTest{

    public static void main(String[] args){

        Scanner in = new Scanner(System.in);

        int one = in.nextInt();

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

            int sum=0;

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

                if(i%j==0)

                    sum+=j;

            }

            if(sum==i)

                System.out.println(i);

        }

    }

}
```

#### 第八次实验

1. 判断某一个数是否是素数
2. 统计字符串的英文字符和数字的个数
3. 实现 30s 的定时打印
4. 设计一个 Android 登陆界面

1.

```
import java.util.*;

public class zhisu{

    public static void main(String args[]){

        int x,y,j;

        Scanner i = new Scanner(System.in);

        System.out.print("请输入一个数:");

        y = i.nextInt();

        j=(int)y/2;

        for(x=2,x<=j;x++){

            if(y%x==0){

                System.out.println("不是素数");

                break;

            }

        }

    }

}
```

2.

```
import java.util.Scanner;

public class tongji{

    public static void main(String[] args){

        int charCount=0;

        int numCount=0;

        Scanner sc=new Scanner(System.in);

        System.out.println("请输入一行字符 : ");

        String str = sc.nextLine();

        char[]ch=str.toCharArray();

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

```
{

    if(Character.isDigit(ch[i]))

    {

        numCount++;

    }

    else

    {

        charCount++;

    }

}

System.out.println("字符串的个数 : "+charCount);

System.out.println("数字的个数 : "+numCount);

}
```

#### 3.

```
public class dingshidayin{

    public static void main (String[]args){

        while(true){

            try {

                Thread.sleep(30000);

                System.out.println(" helloworld");

            } catch (InterruptedException e) {

                // TODO Auto-generated catch block

                e.printStackTrace();

            }

        }

    }

}
```

#### 4.

```
import java.awt.Container;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JTextArea;

import javax.swing.JTextField;

public class andriod {

    /**

     * @param args

     */

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        JFrame jf = new JFrame();

        Container ct = jf.getContentPane();

        jf.setLayout(null);

        jf.setSize(800, 800);

        jf.setDefaultCloseOperation(0);

        JTextArea jtf =new JTextArea();
```

```
jtf.setBounds(0, 0, 100, 50);

jtf.setVisible(true);

ct.add(jtf);

JTextField one = new JTextField();

one.setBounds(0, 50, 100, 50);

one.setVisible(true);

ct.add(one);

JButton jbt = new JButton();

jbt.setLayout(null);

jbt.setText("login");

jbt.setBounds(0, 80, 10, 20);

jbt.setSize(100, 100);

jbt.setVisible(true);

ct.add(jbt);
```

```
jf.setVisible(true);

}
```