【问题描述】

用类描述计算机中CPU的速度和硬盘的容量。要求Java应用程序有4个类，名字分别是PC、CPU、HardDisk和Main,其中，Main是主类。

CPU类有int型speed成员变量，提供getSpeed()方法返回speed的值；setSpeed(int m)方法设置speed的值。

HardDisk类有int型amount成员变量，提供getAmount()方法返回amount的值，setAmount((int m)方法设置amount的值。

PC类组合CPU和HardDisk类的对象，即PC类有CPU类型的成员变量cpu和HardDisk类型的成员变量HD。PC类提供setCUP(CPU c)方法和setHardDisk(HardDisk h)方法。PC类有show（）方法可以输出CPU的速度和HD的容量。

【输入形式】  
【输出形式】  
【样例输入】

2200

200  
【样例输出】

电脑的速度是：2200

电脑的容量是：200

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
public  class  Main{  
          public  static  void  main(String[]  args)  {  
                Scanner  r  =  new  Scanner(System.in);  
                //键盘录入速度和容量  
                    int speed=r.nextInt();

int amount=r.nextInt();              
                CPU  cpu  =  new  CPU();  
                cpu.setSpeed(speed);  
                HardDisk  disk  =  new  HardDisk();  
                disk.setAmount(amount);  
                PC  pc  =  new  PC();  
                pc.setCPU(cpu);  
                pc.setHardDisk(disk);  
                pc.show();  
        }  
}  
class  CPU{  
        //CPU类有int型speed成员变量，提供getSpeed()方法返回speed的值；setSpeed(int  m)方法设置speed的值。  
       int speed;

int getSpeed(){

return speed;

}

void setSpeed(int m){

speed=m;

}     
}  
class  HardDisk{  
        //HardDisk类有int型amount成员变量，提供getAmount()方法返回amount的值，setAmount((int  m)方法设置amount的值。  
      private int amount;

public void setAmount(int amount){

this.amount=amount;

}

public int getAmount(){

return amount;

}

}  
class  PC{  
              //PC类组合CPU和HardDisk类的对象，PC类提供setCPU(CPU  c)方法和setHardDisk(HardDisk  h)方法。PC类有show（）方法可以输出CPU的速度和HD的容量。

窗体底端

CPU cpu;

HardDisk HD;

public void setCPU(CPU c){

this.cpu=c;

}

public void setHardDisk(HardDisk h){

this.HD=h;

}

public void show(){

System.out.println("电脑的速度是："+cpu.getSpeed());

System.out.println("电脑的容量是："+HD.getAmount());

}

}

【问题描述】

已有一个汽车Auto类:包括轮胎个数1个成员变量，1个构造方法，3个实例方法实现加速、减速和停车的功能(行为)。

要求再定义一个小汽车类Car继承Auto并添加空调、CD机成员变量,同时要求重写父类的加速和减速的方法。

注意：

1、接收键盘输入轮胎个数、CD机和空调品牌名称；

2、输出小轿车装备了X轮小轿车装备了Y牌CD机和Z牌空调

      轿车正在加速！

      轿车正在减速！

      停车啦！

上面的X代表轮胎个数，Y代表CD机的品牌名称，Z代表空调机的品牌名称。

【输入形式】  
【输出形式】  
【样例输入】

4

zhizao

wangpai

【样例输出】

4轮小轿车装备了zhizao牌CD机和wangpai牌空调

轿车正在加速！

轿车正在减速！

停车啦！

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                //通过键盘输入轮胎个数n、CD机和空调品牌名称airC  
                Scanner r=new Scanner(System.in);

int n=r.nextInt();

String CD=r.next();

String airC=r.next();                  
                car  c  =  new  car(n,  CD,  airC);  
                System.out.println(n  +  "轮小轿车装备了"  +  c.getCD()  +  "牌CD机和"  +  c.getAir\_condition()  +  "牌空调");  
                c.jiasu();  
                c.jiansu();  
                c.tingzhi();  
        }  
}  
  
class  Auto  {  
        private  int  luntai;  
        public  Auto(int  luntai)  {  
                this.luntai  =  luntai;  
        }  
        public  void  jiasu()  {  
                System.out.println("加速啦！");  
        }  
        public  void  jiansu()  {  
                System.out.println("减速啦！");  
        }  
        public  void  tingzhi()  {  
                System.out.println("停车啦！");  
        }  
}  
  
class  car  extends  Auto  {  
        //定义一个小汽车类Car继承Auto并添加空调、CD机成员变量,同时要求重写父类的加速和减速的方法。  
    public String air\_condition,cd;

car(int luntai,String cd,String air\_condition){

super(luntai);

this.air\_condition=air\_condition;

this.cd=cd;

}

public String getCD(){

return cd;

}

public String getAir\_condition(){

return air\_condition;

}

public void jiasu(){

System.out.println("轿车正在加速！");

}

public void jiansu(){

System.out.println("轿车正在减速！");

}            
}

窗体底端

【问题描述】

设计一个动物声音“模拟器”,希望模拟器可以模拟许多动物的叫声，要求如下。

①编写抽象类Animal：Animal抽象类有2个抽象方法cry()和getAnimalName()，即要求通过键盘输入各种具体的动物名称。

②编写模拟器类Simulator：该类有一个playSound(Animal animal)方法，该方法的参数是Animal类型。即参数animal可以调用Animal的子类重写的cry()方法,调用子类重写的getAnimalName()方法显示动物种类的名称。

③编写Animal类的子类：Dog类和Cat类。

④编写Main类，创建Simulator类对象s，进行测试。

【输入形式】  
【输出形式】

【样例输入】

猫

狗

【样例输出】

猫

喵喵喵！

狗

汪汪汪！

【样例说明】  
【评分标准】

窗体顶端

import  java.util.\*;  
abstract  class  Animal{  
//Animal抽象类有2个抽象方法cry()和getAnimalName()，表示叫声和种类名称。  
    abstract void cry();

abstract void getAnimalName();      
}  
class  Simulator{  
        //  Simulator类中有一个playSound(Animal  animal)方法，该方法的参数是Animal类型。即参数animal可以调用Animal的子类重写的cry()方法通过键盘输入传入具体动物的声音，调用子类重写的getAnimalName()方法显示动物种类的名称。  
           public void playSound(Animal animal){

animal.getAnimalName();

animal.cry();

}     
}  
class  Dog  extends  Animal{  
                //编写Animal类的子类Dog类。  
        private String name;

Dog (String name ){

this.name=name;

}

void cry(){

System.out.println("wang wang!");

}

void getAnimalName(){

System.out.println(name);

}  
}  
class  Cat  extends  Animal{  
                //编写Animal类的子类Cat类。  
        private String name;

Cat(String name){

this.name=name;

}

void cry(){

System.out.println("miao miao!");

}

void getAnimalName(){

System.out.println(name);

}  
}  
class  Main{  
        public  static  void  main(String[]  args)  {  
                Scanner  r  =  new  Scanner(System.in);  
                String  name1  =  r.next();  
                String  name2  =  r.next();  
                Simulator  b  =  new  Simulator();  
                Animal  animal;  
                animal  =  new  Cat(name1);  
                b.playSound(animal);  
                animal  =  new  Dog(name2);  
                b.playSound(animal);  
                    
        }  
}

窗体底端

【问题描述】

创建一个类Main。为该类定义三个构造方法：分别执行下列操作

1、传递前两个整数值并输出其中最大的一个值，输出格式为：最大值是：X

2、传递三个int值并求出其乘积 ，输出格式为：三数的乘积是：Y

3、传递两个字符串值并检查其是否相同，输出格式为：两字符串相同或

                                                                 输出格式为：两字符串不相同

4、在main方法中测试构造函数的调用

【输入形式】  
【输出形式】  
【样例输入】

5 3 6  XYza   AbcXy

【样例输出】

 最大值是：5

 三数的乘积是：90

 两字符串不相同

【样例说明】  
【评分标准】

窗体顶端

import  java.util.\*;  
public  class  Main{  
        //按题目要求创建3个构造方法分别求最大值，乘积和字符串匹配  
      public Main(int n1,int n2){

System.out.println(n1>n2?"最大值是："+n1:"最大值是："+n2);

}

public Main(int n1,int n2,int n3){

System.out.println("三数的乘积是："+n1\*n2\*n3);

}

public Main(String str1,String str2){

System.out.println(str1.equals(str2)?"两字符串相同":"两字符串不相同");

}  
        public  static  void  main(String[]  dasd)  {  
                  //键盘录入3个整型和2个字符串  
           Scanner in=new Scanner(System.in);

int n1=in.nextInt();

int n2=in.nextInt();

int n3=in.nextInt();

String str1=in.next();

String str2=in.next();

                Main  c=new  Main(n1,n2);  
              Main  c1=new  Main(n1,n2,n3);  
              Main  c2=new  Main(str1,str2);  
        }  
}

窗体底端

【问题描述】

1.写一个类A ,该类创建的对象可以调用方法f ,输岀英文字母表(a~z)；

2.然后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，而且可以调用子类新增的方法g输岀希腊字母表(α~π)。

注意：（1）英文字母表输出在第一行，希腊字母表输出在第二行。

即每一字母表输出一行后要输出换行。

（2）字母之间不能有其它字符。

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

abcdefghijklmnopqrstuvwxyz

αβγδεζηθικλμνξοπ

【样例说明】  
【评分标准】

窗体顶端

class  A  {  
        //创建方法f  ,输岀英文字母表(a~z)  
               public final void f() {

for (char i = 'a'; i <= 'z'; i++) {

System.out.print(i);

}

}  
}  
class  B  extends  A{  
        //创建方法g，输岀希腊字母表(α~π)  
              public void g() {

for (char c = 'α'; c <= 'π'; c++) {

System.out.print(c);

}

}  
}  
public  class  Main{  
            public  static  void  main(String[]  args)  {  
                B  b=new  B();  
                b.f();  
                System.out.println();  
                b.g();  
        }  
}

窗体底端

【问题描述】

设计一个灯泡类(Buble)，灯泡类有发亮的方法faLiang()，输出灯泡可以发光，其中有红灯泡类(RedBuble)和  
绿灯泡类(GreenBuble)他们都继承灯泡类(Buble)一个发亮的方法，分别输出发红光和发绿光。  
最后写一个测试类Main，声明红灯泡类和绿灯泡类对象，分别调用发光方法

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

发红光

发绿光

【样例说明】  
【评分标准】

窗体顶端

public  class  Main  {    
public  static  void  main(String[]  args)  {  
            RedBuble  redBuble=new  RedBuble();  
            redBuble.faLiang();  
    GreenBuble  blueBuble=new  GreenBuble();  
    blueBuble.faLiang();  
        }  
}  
  
class  Buble    {//灯泡类  
public  void  faLiang(){  
        System.out.println("灯泡可以发亮");  
}  
}  
  
class  GreenBuble  extends  Buble  {  
//重写faLiang()，输出发绿光  
public void faLiang(){

System.out.println("发绿光");

}

}  
  
class  RedBuble  extends  Buble  {  
//重写faLiang()，输出发红光  
public void faLiang(){

System.out.println("发红光");

}

}

窗体底端

【问题描述】

利用构造方法建立一个汽车类Car：包括轮胎个数，汽车颜色，车身重量等属性。

在主类Main中通过构造方法创建对象。 要求： 汽车能够加速，减速，停车。

 注意：

1、接收键盘输入依次输入车轮个数，车身自重和车身颜色；

2、输出X轮的Y色Z重量的汽车将有以下情况：

     接着分别调用加速、减速和停车方法，依次分别输出：加速啦!

                                                                                        赶快减速!

                                                                                        车已停了!

其中X代表车轮数，Y代表车身颜色，Z代表车身自重。

例如输入车轮个数、车身自重和车身颜色的数据如下：

4  1.6  Black

则输出的结果是：

4轮的Black色1.6吨汽车将有以下情况：

加速啦!

赶快减速!

车已停了!

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】  
【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                                //按要求创建汽车类对象，并通过键盘录入轮胎个数，汽车颜色，车身重量，并按要求输出  
      Scanner n=new Scanner(System.in);

int luntai=n.nextInt();

String color=n.next();

double weight=n.nextDouble();

Car c=new Car(luntai,color,weight);

System.out.println(luntai + "轮的" + color + "色" + weight + "吨汽车将有以下情况：");

                c.SpeedUp();  
                c.SpeedDown();  
                c.Stop();  
        }  
}  
  
class  Car  {  
                //利用构造方法建立一个汽车类Car：包括轮胎个数，汽车颜色，车身重量等属性,同时创建3个实例方法，要求  汽车能够加速，减速，停车。  
    private int luntai;

private String color;

private double weight;

public Car(int x,String y,double z){

this.luntai=x;

this.color=y;

this.weight=z;

}

public void SpeedUp(){

System.out.println("加速啦!");

}

public void SpeedDown(){

System.out.println("赶快减速!");

}

public void Stop(){

System.out.println("车已停了!");

}

}

窗体底端

【问题描述】

①Main类是一个简单的计算器，其中有两个整型数据成员n1和n2声明为私有，1个构造方法赋予n1和n2初始值。

②再为该类定义加的方法addition、减的方法subtraction、乘的方法multiplication、除的方法division等公有成员方法，分别对两个成员变量执行加、减、乘、除的运算。

③在main方法中创建对象调用各个方法并显示计算结果。

注意：其中两数的商是指第一个操作数除以第二个操作数的结果，且输出是8.3f格式的实数(即输出的商占用8个字符长度，其中保留3位小数)，使用printf格式输出。输出格式如样例输出所示。

【输入形式】  
【输出形式】  
【样例输入】

40 7

【样例输出】

Add:47

Sub:33

Mul:280

Div:   5.714

【样例说明】  
【评分标准】

窗体顶端

  import  java.util.Scanner;  
public  class  Main  {  
        //定义2个成员变量，1个构造方法，4个实例方法  
          private int n1,n2;

public Main(int n1,int n2){

this.n1=n1;

this.n2=n2;

}

public void addition(){

System.out.println("Add:"+(this.n1+this.n2));}

public void subtraction(){

System.out.println("Sub:"+(this.n1-this.n2));}

public void multiplication(){

System.out.println("Mul:"+(this.n1\*this.n2));}

public void division(){

System.out.printf("Div:%8.3f",((float)this.n1/this.n2));}        
        public  static  void  main(String[]  args)  {  
                int  Op1,  Op2;  
                //  通过键盘录入两个操作数  
                      Scanner n=new Scanner(System.in);

Op1=n.nextInt();

Op2=n.nextInt();          
                Main  lianxi  =  new  Main(Op1,  Op2);  
                lianxi.addition();  
                lianxi.subtraction();  
                lianxi.multiplication();  
                lianxi.division();  
        }  
}

窗体底端

【问题描述】

（1）编写类Person，该类定义了3个成员变量（name代表姓名、gender代表性别、age代表年龄）和两个构造方法（无参数的使用默认值初始化成员变量，有参数的使用给定值初始化成员变量）。3个get方法分别用来获得姓名、性别和年龄。

（2）编写类Main，该类用来测试Person类。在该类的main()方法中，创建了两个Person对象，接受键盘输入姓名、性别、年龄并输出其属性。

【输入形式】  
【输出形式】  
【样例输入】

zhangsan

男

23

【样例输出】

调用无参构造方法，并输出默认信息：

无参,输出默认值!

姓名:null

性别:null

年龄:0

调用有参构造方法，通过键盘录入个人信息并输出 ：

姓名:zhangsan

性别:男

年龄:23

【样例说明】系统自动进行输入输出测试，输出格式如上所示

【评分标准】

窗体顶端

import  java.util.Scanner;  
class  Person  {  
//编写类Person，该类定义了3个成员变量（name代表姓名、gender代表性别、age代表年龄）和两个构造方法（无参数的使用默认值初始化成员变量，有参数的使用给定值初始化成员变量）。3个get方法分别用来获得姓名、性别和年龄。  
 String name;

String gender;

int age;

Person(){

System.out.println("无参,输出默认值!");

}

Person(String name,String gender,int age){

this.name=name;

this.gender=gender;

this.age=age;

}

public String getName(){

return name;

}

public String getGender(){

return gender;

}

public int getAge(){

return age;

}

}  
  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                System.out.println("调用无参构造方法，并输出默认信息：  ");  
                                  //创建对象p1，调用Person类的无参构造方法  
                Person n=new Person();

System.out.println("name:"+n.getName());

System.out.println("gender:"+n.getGender());

System.out.println("age:"+n.getAge());  
                                System.out.println("姓名:"  +  p1.getName());  
                System.out.println("性别:"  +  p1.getGender());  
                System.out.println("年龄:"  +  p1.getAge());  
                System.out.println("调用有参构造方法，通过键盘录入个人信息并输出  ：");  
                              //  创建对象p2，调用Person类的有参构造方法，并通过键盘录入个人信息  
   Scanner x=new Scanner(System.in);

String name=x.next();

String gender=x.next();

int age=x.nextInt();

Person z=new Person(name,gender,age);

System.out.println("name:"+z.getName());

System.out.println("gender:"+z.getGender());

System.out.println("age:"+z.getAge());               
                                System.out.println("姓名:"  +  p2.getName());  
                System.out.println("性别:"  +  p2.getGender());  
                System.out.println("年龄:"  +  p2.getAge());  
  
  
        }  
}

【问题描述】

（1）编写一个圆类Circle，该类拥有：一个成员变量radius(私有，浮点型)，存放圆的半径，二个构造方法：Circle():将半径设为0，Circle(double r)：创建Circle对象时将半径初始化为r;

三个成员方法：double getArea():获取圆的面积、double getPerimeter():获取圆的周长、void show():将圆的半径，周长、面积输出到屏幕

（2）编写一个圆柱类Cylinder,它继承与上面的Circle类，还拥有：一个成员变量double hight(私有 浮点型)：圆柱体的高；构造方法Cylinder(double r,double h):创建Circle对象时将半径初始化为r;成员方法double getVolume():获取圆柱体的体积 void showVolume():将圆柱体的体积输出到屏幕。

编写测试类Main，创建类的对象，分别设置圆的半径、圆柱体的高，计算并分别显示圆半径、圆面积、圆周长，圆柱体的体积。

注意：1、接收键盘输入圆的半径和圆柱的高度

           2、输出圆的半径、周长、面积和圆柱体积的采用语句：

                System.out.println("圆半径="+radius+"圆周长"+getPerimeter()+"圆面积"+getArea());

                System.out.println("圆柱体积="+getVolume());

         3、不可输出其它任何信息

【输入形式】  
【输出形式】  
【样例输入】

3 9

【样例输出】

圆半径=3.0圆周长18.84圆面积28.259999999999998

圆半径=3.0圆周长18.84圆面积28.259999999999998

圆柱体积=254.33999999999997

【样例说明】  
【评分标准】

窗体顶端

import  java.util.\*;  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                System.out.println("输入圆的半径和圆柱的高：");  
                //通过键盘录入圆的半径和圆柱的高  
Scanner n=new Scanner(System.in);

double r=n.nextDouble();

double h=n.nextDouble();

                Circle  ci=new  Circle(r);  
                ci.show();  
                Cylinder  cyl=new  Cylinder(r,h);  
                cyl.show();  
                cyl.showVolume();  
          }  
}  
  
class  Circle  {  
      private  double  radius;  
      Circle(){  
              radius=0.0;  
      }  
    Circle(double  r){  
            radius=r;  
      }  
      double  getPerimeter()  {  
                return  2\*3.14\*radius;  
        }  
        double  getArea()  {  
          return  3.14\*radius\*radius;  
        }  
        void  show()  {  
                  System.out.println("圆半径="+radius+"圆周长"+getPerimeter()+"圆面积"+getArea());  
          }  
}  
  
class  Cylinder  extends  Circle  {  
        //Cylinder类包含：一个成员变量double  hight(私有  浮点型)：圆柱体的高；构造方法Cylinder(double  r,double  h):创建Circle对象时将半径初始化为r;成员方法double  getVolume():获取圆柱体的体积  void  showVolume():将圆柱体的体积输出到屏幕。  
private double hight;

Cylinder(double r,double h){

super(r);

hight=h;

}

public double getVolume(){

return getArea()\*hight;

}

public void showVolume(){

System.out.println("圆柱体积="+getVolume());

}

}

窗体底端

【问题描述】

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。在应用程序的主类中创建一个父类的对象，例如Oa，该对象Oa调用方法f返回最大公约数，然后Oa作为子类对象的上转型对象调用方法f返回最小公倍数。【输入形式】

【输出形式】  
【样例输入】

8 6

【样例输出】

8和6的最大公约数 ：2

8和6的最小公倍数 ：24

【样例说明】  
【评分标准】

窗体顶端

import  java.util.\*;  
public  class  Main  {  
        public  static  void  main  (String  args[  ])  {  
                  int  a,b;  
                  Scanner  reader=new  Scanner(System.in);  
                  a=reader.nextInt();  
                  b=reader.nextInt();  
                A  Oa=new  A();  
                System.out.println(a+"和"+b+"的最大公约数  ："+Oa.f(a,b));  
                Oa  =  new  B();  
                System.out.print(a+"和"+b+"的最小公倍数  ："+Oa.f(a,b));    
      }  
}  
class  A  {  
      //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  
 public int f(int a,int b){

int r=b%a;

while(r!=0){

b=a;

a=r;

r=b%a;}

return a;

}       
}  
class  B  extends  A  {  
    //重写方法f，而且重写的方法将返回a和b的最小公倍数  
 public int f(int a,int b){

int division=super.f(a,b);

return (a\*b)/division;

}       
}

窗体底端

【问题描述】使用多态实现酒店住宿房费计算。

民宿Inn类与星级酒店StarHotel类都属于旅店Hotel类，都有品牌和地址属性，都具备计算住宿费的功能，但是计算方法不同，所以可以将计算住宿费的方法定义为抽象方法。

根据品牌的不同，民宿住宿一晚300元，星级酒店住宿一晚1500元。请按要求补充完整代码，实现酒店的住宿房费计算功能。

【输入形式】  
【输出形式】  
【样例输入】

3  
【样例输出】

大理印象云南

房费总计：900

希尔顿三亚

房费总计：4500

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                Scanner  r  =  new  Scanner(System.in);  
                int  days  =  r.nextInt();  
                Hotel  h;  
                h  =  new  Inn("大理印象",  "云南");  
                h.show();  
                System.out.println("房费总计："  +  h.callRent(days));  
                h  =  new  StarHotel("希尔顿",  "三亚");  
                h.show();  
                System.out.println("房费总计："  +  h.callRent(days));  
        }  
  
}  
  
abstract  class  Hotel  {  
        private  String  brand,  adress;  
        public  Hotel(String  brand,  String  adress)  {  
                this.brand  =  brand;  
                this.adress  =  adress;  
        }  
        public  abstract  int  callRent(int  days);//根据天数计算住宿费  
        //定义一个实例方法show（）,该方法可以直接输出品牌和地址  
public void show(){

System.out.println(brand+adress);

}                        
}  
//定义两个类民宿Inn类与星级酒店StarHotel类继承自Hotel类

窗体底端

class Inn extends Hotel{

public Inn (String brand,String adress){

surer(brand,adress);}

public int callRent(int days){

return 300\*days;

}

}

class StarHotel extends Hotel{

public StarHotel(String brand,String adress){

surer(brand,adress);}

public int callRent(int days){

return 1500\*days;

}

}

有一个接口UnitPrice，只有一个抽象类double unitprice( )用于返回商品的单价，要求：

1）编写电视机类、计算机类和手机类来实现接口UnitPrice。

2）编写一个公共Main类，其中包含main方法和static double get(UnitPrice u)方法。

3）在main方法中，从键盘上输入电视、计算机和手机的销售个数，调用get方法可以分别得到商店各个商品的单价，之后输出商品的总销售额。

【输入形式】  
【输出形式】  
【样例输入】

120 150 200

【样例输出】

电视总销售额为：480000.0

计算机总销售额为：900000.0

手机总销售额为：600000.0

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
  
public  class  Main  {  
        UnitPrice  u;  
                //通过调用static  double  get(UnitPrice  u)方法可以得到商品的单价  
   static double get(UnitPrice u){

return u.unitprice();

}       
        public  static  void  main(String[]  args)  {  
                Scanner  in  =  new  Scanner(System.in);  
                int  a,  b,  c;  
                System.out.println("依次输入电视、计算机和手机的销售个数：");  
                a  =  in.nextInt();  
                b  =  in.nextInt();  
                c  =  in.nextInt();  
                System.out.println("电视总销售额为："  +  a  \*  Main.get(new  TV(4000.0)));  
                System.out.println("计算机总销售额为："  +  b  \*  Main.get(new  Computer(6000.0)));  
                System.out.println("手机总销售额为："  +  c  \*  Main.get(new  Phone(3000.0)));  
        }  
  
}  
interface  UnitPrice  {  
        public  double  unitprice();  //  返回商品的单价  
}  
//定义三个类电视TV、计算机Computer和手机Phone来实现接口UnitPrice。

窗体底端

class TV implements UnitPrice{

private double price;

TV(double price){

this.price=price;

}

public double unitprice(){

return price;

}

}

class Computer implements UnitPrice{

private double price;

Computer(double price){

this.price=price;

}

public double unitprice(){

return price;

}

}

class Phone implements UnitPrice{

private double price;

Phone(double price){

this.price=price;

}

public double unitprice(){

return price;

}

}

【问题描述】

有一个抽象类ObjectVolume，该类只有一个抽象方法double getVolume()用于返回物体的体积，要求：

1）编写球体类，立方体类和圆柱体类，它们是抽象类ObjectVolume的子类。

2）编写一个Main类，其中包含main方法和static void get(ObjectVolume obj)方法（可以看到，get方法的形参是抽象类ObjectVolume声明的对象）。

（3）在main方法中，从键盘上输入1（表示球体）、2（表示立方体）、3（表示圆柱体）时，调用get方法可以分别得到球体、立方体、圆柱体的体积。

【输入形式】  
【输出形式】  
【样例输入】

1

2  
【样例输出】

请输入1-3之间的任一个数字进行选择：

请输入半径：

25.12

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
  
public  class  Main  {  
        public  static  void  get(ObjectVolume  obj)  {  
                System.out.println(obj.getVolume());  
        }  
  
        public  static  void  main(String[]  args)  {  
                Scanner  st  =  new  Scanner(System.in);  
                System.out.println("请输入1-3之间的任一个数字进行选择：");  
                int  n  =  st.nextInt();  
                                ObjectVolume  obj=null;  
                if  (n  ==  1)  {//  球体  
                        System.out.println("请输入半径：");  
                        int  r  =  st.nextInt();  
                                                obj=new  Sphere(r);  
                }  
                if  (n  ==  2)  {//  立方体  
                        System.out.println("请输入边长：");  
                        int  a  =  st.nextInt();  
                                              obj=new  Cube(a);  
                }  
                if  (n  ==  3)  {//  圆柱体  
                        System.out.println("请输入半径和高：");  
                        int  r  =  st.nextInt();//  半径  
                        int  h  =  st.nextInt();//  高  
                                              obj=new  Cylinder(r,  h);                  
                }  
                                          Main.get(obj);  
        }  
  
}  
abstract  class  ObjectVolume  {  
        abstract  double  getVolume();  //  返回物体的体积  
}  
//分别定义三个类球体（Sphere）、立方体（Cube）、圆柱体（Cylinder），让它们继承ObjectVolume类并重写getVolume()方法。

窗体底端

class Sphere extends ObjectVolume{

int r;

Sphere(int r){

this.r=r;

}

public double getVolume(){

return 4/3\*3.14\*r\*r\*r;

}

}

class Cube extends ObjectVolume{

int r;

Cube(int r){

this.r=r;

}

public double getVolume(){

return r\*r\*r;

}

}

class Cylinder extends ObjectVolume{

int r,h;

Cylinder(int r,int h){

this.r=r;

this.h=h;

}

public double getVolume(){

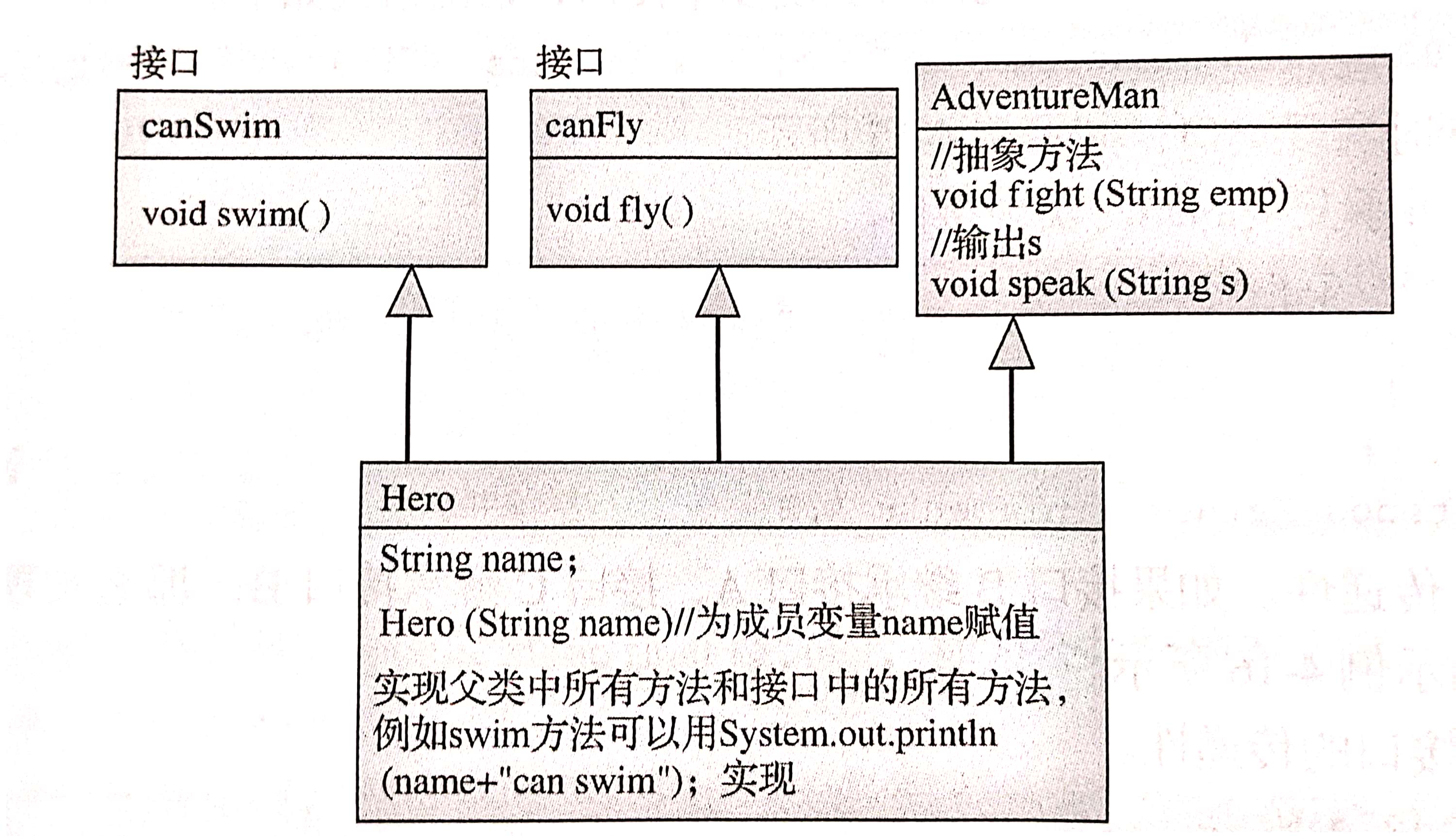
return 3.14\*r\*r\*h;

}

}

【问题描述】

现有游泳接口（Swim）、飞行接口（Fly），还有一个冒险者抽象类（AdventureMan）。英雄类（Hero）是冒险者抽象类（AdventureMan）的子类，其中英雄能有用，会飞行。现需要设计一个英雄类的具体人物测试其上技能是否具备，具体任务描述如下图所示：



【输入形式】  
【输出形式】  
【样例输入】

杨过

多国语言

降龙十八掌

【样例输出】

杨过can swim

杨过can fly

杨过can speak多国语言

杨过can fight降龙十八掌

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                Scanner  in  =  new  Scanner(System.in);  
                String  name  =  in.next();//输入英雄名  
                String  s  =  in.next();//输入英雄所掌握的语言名  
                String  emp  =  in.next();//输入英雄所掌握的功夫名  
                Swim  swim  =  new  Hero(name);  
                swim.swim();  
                Fly  fly  =  new  Hero(name);  
                fly.fly();  
                AdventureMan  a  =  new  Hero(name);  
                a.speak(s);  
                a.fight(emp);  
        }  
}  
interface  Swim  {  
        void  swim();  
}  
interface  Fly  {  
        void  fly();  
}  
abstract  class  AdventureMan  {  
        abstract  void  fight(String  emp);  
        abstract  void  speak(String  s);  
}  
  
class  Hero  //Hero类继承了AdventureMan的同时也实现了Swim和Fly接口

extends AdventureMan implements Swim,Fly  
  {  
        //Hero类中包含一个成员变量和给该成员变量赋初始值的构造方法  
String name;

Hero(String name){

this.name=name;

}

public void swim(){

System.out.println(name+"can swim");

}

public void fly(){

System.out.println(name+"can fly");

}

public void fight(String emp){

System.out.println(name+"can fight"+emp);

}

public void speak(String s){

System.out.println(name+"can speak"+s);

}

}

窗体底端

【问题描述】

定义一个Java应用程序，用Rectangle类来封装长方形。要求Rectangle类具有double型的长、宽属性和计算周长、面积的方法。

要求属性赋值合理，如果是0或负数则输出相应提示信息，并无法求出周长和面积。

【输入形式】  
【输出形式】  
【样例输入】

-3 0  
【样例输出】

长方形的长不能为0或负数

长方形的宽不能为0或负数

长方形的长为：0.0,宽为：0.0

长和宽为0或负数时无法求出周长

长和宽为0或负数时无法求出面积

【样例输入】

5 9  
【样例输出】

长方形的长为：5.0,宽为：9.0

长方形的周长为：28.0

长方形的面积为：45.0

【样例输入】

-4 4  
【样例输出】

长方形的长不能为0或负数

长方形的长为：0.0,宽为：4.0

长和宽为0或负数时无法求出周长

长和宽为0或负数时无法求出面积

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                Scanner  in  =  new  Scanner(System.in);  
                double  len  =  in.nextDouble();  
                double  wid  =  in.nextDouble();  
                Rectangle  r  =  new  Rectangle();  
                r.setLength(len);  
                r.setWidth(wid);  
                System.out.println("长方形的长为："  +  r.getLength()  +  ",宽为："  +  r.getWidth());  
                r.getPer();  
                r.getArea();  
        }  
}  
class  Rectangle  {  
        private  double  length,  width;  
        //定义一个setLength()方法，判断如果length<=0,则输出"长方形的长不能为0或负数",否则给length赋值。  
public void setLength(double length){

if(length<=0)

System.out.println("长方形的长不能为0或负数");

else

this.length=length;

}                  
        //定义一个setWidth()方法，判断如果width<=0,则输出"长方形的宽不能为0或负数",否则给width赋值。  
public void setWidth(double width){

if(width<=0)

System.out.println("长方形的宽不能为0或负数");

else

this.width=width;

}                  
        public  double  getLength()  {  
                return  length;  
        }  
        public  double  getWidth()  {  
                return  width;  
        }  
//定义一个getPer()方法，判断如果length  <=  0  ||  width  <=  0,则输出“长和宽为0或负数时无法求出周长”，否则按格式输出周长  
public void getPer(){

if(length <= 0 || width <= 0)

System.out.println("长和宽为0或负数时无法求出周长");

else

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

}            
//定义一个getArea()  方法，判断如果length  <=  0  ||  width  <=  0,则输出“长和宽为0或负数时无法求出面积”，否则按格式输出面积  
            
public void getArea(){

if(length <= 0 || width <= 0)

System.out.println("长和宽为0或负数时无法求出面积");

else

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

}  
}

窗体底端

【问题描述】

（1）编写类Book，在该类中定义了3个成员变量，即title表示书名、author表示作者、price表示价格。此外，还有3个get方法分别用来获取书名、作者和价格。

（2）编写Main类，该类用来测试Book类。在该类的main()方法中，创建一个Book对象，接收键盘输入依次输入书名，作者，价格，并输出其属性。

【输入形式】  
【输出形式】  
【样例输入】

Java

Liwei

49

【样例输出】

title: Java

author：Liwei

price: 49.0

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
              //通过键盘输入书名、作者和价格  
 Scanner n=new Scanner(System.in);

String title=n.next();

String author=n.next();

double price=n.nextDouble();         
                Book  book=new  Book(title,author,price);  
                System.out.println("title:  "+book.getTitle());  
                System.out.println("author:  "+book.getAuthor());  
                System.out.println("price:  "+book.getPrice());  
        }  
}  
class  Book  {  
//编写类Book，在该类中定义了3个成员变量，即title表示书名、author表示作者、price表示价格。此外，还有3个get方法分别用来获取书名、作者和价格。  
 private String title;

private String auther;

private double price;

Book(String title,String auther,double price){

this.title=title;

this.auther=auther;

this.price=price;

}

public String getTitle(){

return title;

}

public String getAuthor(){

return auther;

}

public double getPrice(){

return price;

}

}

窗体底端

【问题描述】

①编写类Shape，该类是一个抽象类，在该类中有一个抽象的方法getArea()，用来计算形状的面积。

②编写类Circle（圆形），该类继承自Shape，并实现了其抽象方法getArea()。在该类的构造方法中，获取了圆形的半径，以此在getArea()中计算面积(∏=3.14)。

③编写类Rectangle（矩形），该类继承自Shape，并实现了其抽象方法getArea()。在该类的构造方法中，获取了矩形的长和宽，以此在getArea()中计算面积。

④编写Main类，创建Shape类的变量s，利用上转型对象，分别输出圆形和矩形的面积。

【输入形式】  
【输出形式】

【样例输入】

2

12

13  
【样例输出】

Circle area:12.56

Rectangle area:156.0

【样例说明】

【评分标准】

窗体顶端

import  java.util.\*;  
abstract  class  Shape{  
          abstract  void  getArea();  
}  
  
class  Circle  extends  Shape{  
        //该类继承自Shape，并实现了其抽象方法getArea()。在该类的构造方法中，获取了圆形的半径，以此在getArea()中计算面积。  
 private double r;

Circle(double r){

this.r=r;

}

public void getArea(){

double area;

area=3.14\*r\*r;

System.out.println("Circle area:"+area);

}               
          
}  
class  Rectangle  extends  Shape{  
        //该类继承自Shape，并实现了其抽象方法getArea()。在该类的构造方法中，获取了矩形的长和宽，以此在getArea()中计算面积。  
 double length;

double width;

Rectangle(double length,double width){

this.length=length;

this.width=width;

}

public void getArea(){

double area;

area=length\*width;

System.out.println("Rectangle area:"+area);

}             
}  
  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                Scanner  s=new  Scanner(System.in);  
                double  r=s.nextDouble();  
                Shape  shape;  
                shape= new Circle(r);//利用上转型对象，输出圆形的面积  
                shape.getArea();  
                double  length=s.nextDouble();  
                double  width=s.nextDouble();  
                shape= new Rectangle(length,width);//利用上转型对象，输出矩形的面积  
                shape.getArea();  
                  
                  
        }  
}

窗体底端

【问题描述】

①创建一个由100个按照2，4，6，8，10.........规律的整数构成的数组。

②提示用户输入数组的下标，然后显示对应的元素值。如果指定的下标越界，就显示消息“yuejie!”。

【输入形式】

【输出形式】  
【样例输入】

33  
【样例输出】

68

【样例输入】

102  
【样例输出】

yuejie!

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
class  Main  {  
        public  static  void  main(String[]  args)  {  
                int[]  a  =  new  int[100];  
                //按照2，4，6，8，10.........规律将整数存入数组a中。  
 for(int i=0,j=0;i<a.length;i++){

j=j+2;

a[i]=j;

}                                 
                System.out.println("请输入一个数组下标:");  
                Scanner  in  =  new  Scanner(System.in);  
                int  i  =  in.nextInt();  
                //当输入的值大于数组下标时，使用try-catch语句捕获异常，并输出提示信息；否则直接输出下表对应的数组值  
                              try  {  
System.out.println(a[i]);

}

catch(Exception e){

System.out.println("yuejie!");

                          }  
        }  
  
}

窗体底端

【问题描述】用Thread类的子类创建线程对象，输出5遍“Hello!”,并通过Thread.currentThread().getName() 输出线程名。

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

Thread-0:Hello!

Thread-0:Hello!

Thread-0:Hello!

Thread-0:Hello!

Thread-0:Hello!

【样例说明】  
【评分标准】

窗体顶端

public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                MyThread  t  =  new  MyThread();  
                t.start();  
  
        }  
}  
class  MyThread  extends  Thread  {  
public void run(){

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

System.out.println(Thread.currentThread().getName()+":Hello!");

}

}                              
}

窗体底端

【问题描述】

使用Scanner中的useDelimiter( )方法来解析字符串 "数学87.0分，物理76.0分，英语96.0分"，将正则表达式做为分隔标记。hasNext()：返回一个boolean值，只要对象中还有单词，就返回true。next()：依次返回字符串中的单词。对解析出的分数进行求和和平均值计算。

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

sum:259.0

avg:86.33333333333333

【样例说明】  
【评分标准】

窗体顶端

import  java.util.InputMismatchException;  
import  java.util.Scanner;  
  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                String  cost  =  "数学87.0分，物理76.0分，英语96.0分";

Scanner scanner=new Scanner(cost);

scanner.useDelimiter("[^0123456789.]+");

double sum=0;

int count=0;

                while  (scanner.hasNext())  {  
                        try  {  
                                double  score  =  scanner.nextDouble();  
                                count++;  
                                sum  =  sum  +  score;  
                        }  catch  (InputMismatchException  exp)  {  
                                String  t  =  scanner.next();  
                        }  
                }  
                System.out.println("sum:"  +  sum  );  
                System.out.println("avg:"  +  sum  /  count);  
        }  
  
}

窗体底端

【问题描述】使用合Runnable接口创建一个线程，让它输出5遍Hello。（注意样例输出格式）

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

t:Hello!

t:Hello!

t:Hello!

t:Hello!

t:Hello!

【样例说明】  
【评分标准】

窗体顶端

public  class  Main  {  
        public  static  void  main(String[]  args)  {  
              //结合Runnable接口创建线程，启动线程

Runnable n=new MyThread();

Thread name=new Thread(n,"t");

name.start();  
  
        }  
  
}  
  
class  MyThread  implements  Runnable  {  
        public  void  run()  {  
                for  (int  i  =  0;  i  <  5;  i++)  {  
                        System.out.println(Thread.currentThread().getName()  +  ":Hello!");  
                }  
        }  
}

窗体底端

对于我们常见的收费单据:    市话费：28.89元,长途话费：128.87元,上网费：298元。

 用StringTokenizer解析出通信费用并输出。

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

28.89

128.87

298

【样例说明】  
【评分标准】

窗体顶端

import  java.util.StringTokenizer;  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                String  s  =  new  String("市话费：28.89元,长途话费：128.87元,上网费：298元");  
 String s1=s.replaceAll("[^0-9.]","#");

StringTokenizer token=new StringTokenizer(s1,"#");                 
                while  (token.hasMoreTokens())  {  
                        String  str  =  token.nextToken();  
                        System.out.println(str);  
                }  
        }  
}

窗体底端

【问题描述】

****现有一行数据：“电视机2567元/台，洗衣机3562元/台，冰箱6573元/台”，利用正则表达式将价格解析出来，并计算他们的总价。****

【输入形式】无  
【输出形式】  
【样例输入】

【样例输出】

sum=12702.0

average=4234.0

【样例说明】  
【评分标准】

窗体顶端

import  java.util.regex.Matcher;  
import  java.util.regex.Pattern;  
import  java.util.Scanner;  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                          //      Scanner  in=new  Scanner(System.in);  
                                String  str="电视机2567元/台，洗衣机3562元/台，冰箱6573元/台";  
                double  sum  =  0;  
                int  count  =  0;  
                              double  price  =  0;  
                double  average  =  0;  
                try  {  
                        //定义一个正则表达式rex:除了数字和.以外的数；然后根据该正则表达式,使用String类的split()方法分解出str中的价格,并将分解出的价格存放在String类型的一维数组s中。  
String rex="[^0-9.]+";

Pattern p=Pattern.compile(rex);

String[] s=p.split(str);                                            
                                                //计算总价格和平均价格，分别存入sum和average中  
                            for(int  i=1;i<s.length;i++)  {  
                                                            price=Double.valueOf(s[i]);  
                                count++;  
                                sum+=price;  
                        }  
                                                  average=sum/count;  
                        System.out.println("sum="  +  sum);  
                        System.out.println("average="  +  average);  
                }  catch  (Exception  e)  {  
                        e.getMessage();  
  
                }  
  
        }  
}

窗体底端

【问题描述】

①自定义一个异常类，名为NopositiveException，该类提供一个能够接受并保存m，n值的构造方法，以及一个能返回m，n值的实例方法。

②定义Computer类，Computer类中有一个getMaxCommonDivisor(int m,int n)方法，当m>0并且n>0时，方法返回两个数的最大公约数。当m<0或n<0时，方法抛出NopositiveException异常。

③自定义Main类，创建一个Computer类对象c，让c调用getMaxCommonDivisor(int m,int n)方法，进行测试。

【输入形式】

【输出形式】

【样例输入】

3 -9

【样例输出】

m:3 or n:-9 <0

【样例输入】

12 9

【样例输出】

3

【样例说明】

【评分标准】

窗体顶端

import  java.util.\*;  
class  NopositiveException  extends  Exception{  
        //类提供一个能够接受并保存m，n值的构造方法，以及一个能返回m，n值的实例方法  
String message;

NopositiveException(int m,int n){

message="m:"+m+" or n:"+n+"<0";

}

public String getMessage(){

return message;

}                    
}  
class  Computer  {  
//Computer类中有一个getMaxCommonDivisor(int  m,int  n)方法，当m>0并且n>0时，方法返回两个数的最大公约数。当m<0或n<0时，方法抛出NopositiveException异常。  
 public void getMaxCommonDivisor(int m,int n)throws NopositiveException{

if(m<0||n<0)throw new NopositiveException(m,n);

int r=m%n;

while(r!=0){

m=n;

n=r;

r=m%n;

}

System.out.println(n);

}                     
}  
public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                Scanner  s=new  Scanner(System.in);  
                int  a=s.nextInt();  
                int  b=s.nextInt();  
                Computer  c=new  Computer();  
                try  {  
                        c.getMaxCommonDivisor(a,  b);  
                }  
                catch(NopositiveException  e)  {  
                        System.out.println(e.getMessage());  
                }  
        }  
}

窗体底端

【问题描述】

①自定义一个异常类，名为ArgumentException，该类提供一个能够接受并保存n值的构造方法，以及一个能返回n值的实例方法。

②定义一个Sum类，Sum类中有一个addContinue(int n)方法，当n>0时，方法返回1+2+…+n的和；当n<=0 时，方法抛出一个ArgumentException异常。

③自定义Main类，创建一个Sum类对象s，让s调用addContinue(int n)方法，进行测试。如分别输入-100和100进行测试。

【输入形式】从标准输入输入一个整型数据  
【输出形式】  
【样例输入】-100  
【样例输出】-100<=0

【样例输入】100  
【样例输出】5050

【样例说明】  
【评分标准】本题要求若输入的数是正整数，则输出1+2+......+n的和；若输入的是负整数或0，则输出：n<=0

窗体顶端

import  java.util.Scanner;  
  
public  class  Main  {  
  
        public  static  void  main(String[]  args)  {  
                Scanner  r  =  new  Scanner(System.in);  
                int  n  =  r.nextInt();  
                Sum  sum  =  new  Sum();  
                try  {  
                        sum.addContinue(n);  
                }  catch  (ArgumentException  e)  {  
                        System.out.println(e.getMessage());  
                }  
  
        }  
  
}  
  
class  ArgumentException  extends  Exception  {  
        //该类提供一个变量n，一个能够接受并保存n值的构造方法，以及一个能返回n值的实例方法  
  String message;

ArgumentException(int n){

message=n+"<=0";

}

public String getMessage(){

return message;

}

}  
  
class  Sum  {  
        //Sum类中有一个addContinue(int  n)方法，当n>0时，方法返回1+2+…+n的和；当n<=0  时，方法抛出一个ArgumentException异常。  
 public void addContinue(int n)throws ArgumentException{

if(n<=0)throw new ArgumentException(n);

int sum=0;

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

sum=sum+i;

}

System.out.println(sum);

}                 
}

窗体底端

【问题描述】按要求将代码补充完整

****利用接口做参数，写个计算器，能完成加减乘除运算。****

****(1)定义一个接口 ICompute含有一个方法int computer（int  n,int m).****

****(2)设计四个类分别实现此接口，完成加减乘除运算****

****(3)设计一个类UseComputer，类中含有方法：public void useCom( ICompute com,int one,int two)，此方法能够用传递过来的对象调用computer方法完成运算，并输出运算结果。****

****(4)设计一个主类Main，调用UseComputer中的方法useCom来完成加减乘除运算，方法的参数通过键盘录入。****

【输入形式】  
【输出形式】  
【样例输入】8 5

【样例输出】

13

3

40

1

【样例说明】  
【评分标准】

窗体顶端

import  java.util.Scanner;  
public  class  Main  {  
public  static  void  main(String[]  args)  {  
//通过键盘录入两个变量one,two  
Scanner n=new Scanner(System.in);

int one=n.nextInt();

int two=n.nextInt();  
UseComputer  use=new  UseComputer();  
use.useCom(new  Add(),  one,  two);  
use.useCom(new  Sub(),  one,  two);  
use.useCom(new  Mul(),  one,  two);  
use.useCom(new  Div(),  one,  two);  
  
}  
  
}  
interface  ICompute{  
int  computer(int  n,int  m);  
}  
class  Add  implements  ICompute{  
      //重写接口中的方法实现加法运算  
public int computer(int n,int m){

return n+m;

}        
}  
class  Sub  implements  ICompute{  
  //重写接口中的方法实现减法运算  
public int computer(int n,int m){

return n-m;

}        
}  
class  Mul  implements  ICompute{  
  //重写接口中的方法实现乘法运算  
 public int computer(int n,int m){

return n\*m;

}       
}  
class  Div  implements  ICompute{  
      //重写接口中的方法实现除法运算。并要求如果除数为0，则返回0，否则返回除结果  
 public int computer(int n,int m){

if((n/m)<0)

return 0;

else

return n/m;

}        
}  
  
class  UseComputer{  
      //类中含有方法：public  void  useCom(  ICompute  com,int  one,int  two)，此方法能够用传递过来的对象调用computer方法完成运算，并输出运算结果。  
public void useCom( ICompute com,int one,int two){

System.out.println(com.computer(one,two));

}          
}

窗体底端

【问题描述】

按下面要求补充完整代码。

1.设计一个接口Advertisement,内含1个抽象方法：getCorpName()（显示公司的名称 ）。

2.设计两个类PhilipseCorp和LenovoCorp，实现Advertisement接口。

3.设计一个AdvertisementBoard类（广告牌），该类有一个Advertisement类型的成员变量adver（广告牌对外留有接口）， adver可以存放任何实现Advertisement接口的类的对象的引用。

一个setAdvertisement方法，给adver设置值；一个show方法，调用接口的getCorpName() 。

4.一个Main类测试,要求分别输出 Lenovo和Philipse

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

Lenovo

Philipse

【样例说明】  
【评分标准】

窗体顶端

interface  Advertisement{  
              public  abstract  void  getCorpName();  
}  
class  LenovoCorp  implements  Advertisement{  
                  //重写Advertisement中的方法  
 public void getCorpName(){

System.out.println("Lenovo");

}                        
}  
class  PhilipseCorp  implements  Advertisement{  
                  //重写Advertisement中的方法  
 public void getCorpName(){

System.out.println("Philipse");

}                    
}  
class  AdvertisementBoard  {  
//设计一个AdvertisementBoard类（广告牌），该类有一个Advertisement类型的成员变量adver（广告牌对外留有接口），  adver可以存放任何实现Advertisement接口的类的对象的引用。一个setAdvertisement方法，给adver设置值；一个show方法，调用接口的getCorpName()  。  
  Advertisement adver;

public void setAdvertisement(Advertisement adver){

this.adver=adver;

}

public void show(){

adver.getCorpName();

}

}  
public  class  Main  {  
public  static  void  main(String[]  args)  {  
AdvertisementBoard  abd=new  AdvertisementBoard();  
abd.setAdvertisement(new  LenovoCorp());  
abd.show();  
abd.setAdvertisement(new  PhilipseCorp());  
abd.show();  
  
}  
}

窗体底端

【问题描述】

单例模式：要保证类有且仅有一个实例，就需要其他的类不能实例化该类。因此，需要将其构造方法设置成私有的，同时，在类中提供一个静态方法，该方法的返回值是该类的一个对象。

①中国历史上有个很特殊的职业，通常其从业者有且仅有一人，那就是皇帝。编写类Emperor，该类的构造方法是私有的。它提供了一个getInstance()方法来获得该类的实例，一个getName()方法输出皇帝的名字。

②编写类Main，该类用来测试Emperor类。在该类的main()方法中，创建了3个Emperor对象，并输出了其名字。

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】  
【样例说明】  
【评分标准】

窗体顶端

class  Emperor  {  
        private  static  Emperor  emperor=null;//声明一个Emperor类的引用（变量）  
        //将构造方法设置为私有的  
private Emperor(){

}

        public  static  Emperor  getInstance()  {//类方法：此方法中返回一个Emperor类的对象  
                if(emperor==null)  {  
                        emperor=new  Emperor();  
                }  
                return  emperor;  
        }  
      //实例方法（普通方法）：用来输出皇帝的名字  singleton  Pattern  
public void getName(){

System.out.println("singleton Pattern");

}            
}  
public  class  Main  {  
          public  static  void  main(String[]  args)  {  
                System.out.println("Emperor1:  ");  
                Emperor  e1=Emperor.getInstance();//创建皇帝对象  
                e1.getName();//输出皇帝的名字  
                System.out.println("Emperor2:  ");  
                Emperor  e2=Emperor.getInstance();  
                e2.getName();  
                System.out.println("Emperor3:  ");  
                Emperor  e3=Emperor.getInstance();  
                e3.getName();  
        }  
}

窗体底端

【问题描述】使用合Runnable接口创建一个线程，让它输出5遍Hello

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

t:Hello!

t:Hello!

t:Hello!

t:Hello!

t:Hello!

【样例说明】  
【评分标准】

窗体顶端

public  class  Main  {  
        public  static  void  main(String[]  args)  {  
              
                Runnable  r  =  new  MyThread();  
                Thread  t  =  new  Thread(r,  "t");  
                t.start();  
  
        }  
  
}  
class  MyThread  implements  Runnable  {  
public void run(){

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

System.out.println(Thread.currentThread().getName()+":Hello!");

}}          
}

窗体底端

【问题描述】用Thread类的子类创建线程对象，输出5遍“Hello!”,并通过Thread.currentThread().getName() 输出线程名。

【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

Thread-0:Hello!

Thread-0:Hello!

Thread-0:Hello!

Thread-0:Hello!

Thread-0:Hello!

【样例说明】  
【评分标准】

窗体顶端

public  class  Main  {  
        public  static  void  main(String[]  args)  {  
                //创建线程，启动线程  
                  
MyThread n=new MyThread();

n.start();  
        }  
}  
class  MyThread  extends  Thread  {  
        public  void  run()  {  
                for  (int  i  =  0;  i  <  5;  i++)  {  
                        System.out.println(Thread.currentThread().getName()  +  ":Hello!");  
                }  
        }  
}

窗体底端

【问题描述】

****编写类Main，在该类中定义了3个方法：一个info()方法是没有参数的，输出：“****0 years old****”；另一个info()方法需要使用一个整型参数，根据整形参数的值，输出：" age years old"，main()方法用来进行测试。****【输入形式】  
【输出形式】  
【样例输入】  
【样例输出】

0 years old

1 years old

2 years old

3 years old

4 years old

5 years old

6 years old

7 years old

8 years old

9 years old

10 years old

【样例说明】

【评分标准】

窗体顶端

public  class  Main  {  
//定义没有参数的info()方法  
public void info(){

System.out.println("0 years old");

}  
//定义有整型参数的info()方法  
int age;

public int info(int age){

System.out.println(age+" years old");

}          
        public  static  void  main(String[]  args)  {  
                Main  ov=new  Main();//创建Main类对象  
                ov.info();//测试无参数的info()方法  
                for(int  i=1;i<=10;i++)  {//测试有参数的info()方法  
                        ov.info(i);  
                }  
        }  
}

窗体底端