**中软国际考题-笔试**

1. 单选题

1、接口需要用什么关键词来声明（B）

1. abstract;
2. interface;
3. extends;
4. static;

2、给出以下代码，请问以下哪个描述是正确的？public XXXX extends something1 , something2（C）

A．如果XXXX是一个接口，something1和something2取消掉，则代码段合法

B．如果XXXX是一个类，something1和something2均是接口，则代码段合法

C．如果XXXX、something1和something2均是接口，则代码段合法

D．因为Java语言不支持多继承机制，所以代码段不合法

3、在使用interface声明一个接口时，只可使用\_\_\_\_\_\_\_修饰符修饰该接口（C）

A.private B.protected C. private protected D.public

4、下面选项中，哪个是Comparator接口提供的方法（D）

A、compare(T o1, T o2)

B、compareTo(T o)

C、equals(Object obj)

D、A和C

5、定义一个接口必须使用的关键字是（C）

A class

B abstract

C interface

D static

6、下列关于接口的定义哪项是正确的？ （C）

    A．interface C{int a；)

    B. public interface A implements B  {)

    C.  public interface A  {int a()；  )

D.  abstract interface D  {)

7、在Java中，已定义两个接口B和C，要定义一个类来实现这两个接口，以下语句正确的是（ C ）。

A. interface A extends B,C B. inteface A implements B,C

C. class A implements B,C D. class A implements B,implements C

8、下面的程序中定义了一个java接口，其中包含（C）处错误

public interface Utility{

private int MAX\_SIZE=20;

int MIN\_SIZE=10;

void use(){

System.out.println("using it");

}

private int getSize();

void setSize(int i);

}

A.1 B.2 C.3 D.4

9、下列描述中正确的是（B）

接口关键词拼写

A. interfase B. interface C. inteface D. intseface

10、下列关于接口描述错误的是（B）

A. 接口中的子类将重写接口中的所有抽象方法

B. 接口的子类必须为实体类

C. 接口中的成员变量必须为常量

D. 接口中没有构造方法。

11、java中接口描述正确的是（D）

A.接口是java中的单继承模式

B.接口方法默认是public

C. 抽象类不存在构造方法

D.接口成员方法只能是抽象方法

12、关于接口的使用，正确的是：

public interface Usb {

}

public abstrace class Interface {

}

A: public interface Usb1 extends Usb {}

B: public interface Usb1 implements Usb {}

C: public interface Usb1 extends Interface {}

D: public interface Usb1 implements Interface {}

13在java中，已定义两个接口B和C，要定义一个实现这两个接口的类，以下语句正确的是(C )

a) interface A extends B,C

b) interface A implements B,C

c) class A implements B,C

d) class A implements B,implements C

14、以下哪个接口的定义是正确的？ （D）

A.interface B{ void print() { } ; }

B.abstract interface B { void print() ; }

C.abstract interface B extends A1,A2 { abstract void print(){ }; }

D.interface B { void print();}

15、定义一个接口时，下列哪个关键字用不到？（D）

A.public

B.extends

C.interface

D.class

16、定义一个接口时，要使用如下哪个关键字？（C）

A.abstract

B.final

C.interface

D.class

17、在使用interface声明一个接口时，只可以使用哪个修饰符修饰该接口。（D）

A.private

B.protected

C.private或者protected

D.public

18、下列叙述正确的是？（C）

A.Java中允许多重继承

B.Java一个类只能实现一个接口

C.Java中只能单重继承

D.Java中一个类可以继承多个抽象类

1. 多选题
2. 下列抽象类语法错误的是？（A、C、D）
3. abstract public class A{};
4. public interface A{};
5. interface public A{};
6. public interface class A;
7. **请选择所有的正确答案**（B、C）

A .在接口中定义的方法默认为private方法

B．在接口中定义的常量默认为public、static、final的

C．一个接口可以继承多个接口

D．关键字implements代表接口之间的继承关系

3、在实现接口的时候，需要注意的是（A、B）

A.一个类可以同时实现多个接口。

B.一个类只能继承一个类，但是能实现多个接口。

C.接口之间不存在继承关系

D.一个接口能继承另一个接口，这和类之间的继承比较相似。

1. 在Java接口中，下列选项中属于有效的方法声明是（ AC ）

A.public void aMethod( );

1. final void aMethod( );

C、void aMethod(){ }

1. private void aMethod( );

5、下面哪些项描述不正确？（A、B）

1. 接口的构造方法由代码隐式生成
2. 接口不存在成员变量
3. 接口必须设定静态的成员变量
4. 接口成员方法修饰符默认为public

6、下面哪些为真（ACD）（多选）

a) 接口里可定义变量，并且变量的值不能修改

b) 任何情况下，类都不能被private修饰

c) 类可以被protected修饰

d) 接口不能实例化

7、下面说法不正确得是(A C D E)

A 不能和abstract一起使用的修饰符有static final public final

B 抽象类之间是继承关系,接口之间也是。

C 接口不能存在普通方法和常量

D局部内部类public，private修饰符修饰 default/static E接口不可以直接实例化，抽象类可以

1. 填空题
2. 一个类在实现一个接口方法的时候，方法的\_\_\_\_\_名字 \_\_\_\_\_ , \_\_\_\_ 返回类型 \_\_\_\_\_、参数个数以及类型。必须与接口中的完全一致
3. Java接口允许\_\_\_\_\_\_\_\_\_\_ 继承（多）
4. 接口的方法声明必须是\_\_\_\_\_\_\_\_即便不写默认也是。(public abstract)
5. JAVA中能实现多继承的方式是\_\_\_\_\_\_\_\_\_\_。(实现多个接口)
6. 一个类通过使用\_\_\_\_\_\_\_\_\_\_\_关键字，声明自己实现一个或多个接口。(implements)
7. 为了克服\_\_\_\_\_\_\_\_\_\_\_缺点，Java使用了接口，一个类可以实现多个接口。(单继承)

7、类实现接口格式： class \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ 接口名{} 修饰。(class 类名 implements)

8、Java8中接口内可以有\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_\_ 组成。（静态常亮，抽象方法，普通方法）

9、接口的方法必须定义为 权限，并且只能是抽象方法 （public）

10、接口不能存在 和  （普通方法 常量）

1. 判断题
2. 一个类只可以实现一个接口（F）（一个类可以实现多个接口）
3. 接口体中必须提供方法实现（F）（接口体中只允许声明方法，不允许提供方法实现）
4. 如果父类实现了某接口，子类必须显式的使用关键字implements声明自己实现这个接口（F）（如果父类实现了某接口，子类也就自然实现了该接口，子类不必在显式地使用关键字去声明自己实现这个接口）
5. Java 类允许多继承。（ F ）
6. 接口和类都允许多继承（F）
7. 接口和类都不允许多继承（F）
8. 接口不能被实例化，但是可以被实现类创建。（ T ）
9. 接口里面可以包含方法具体实现。（ F ）
10. 接口可以多继承。（ T ）
11. 类实继承接口必须实现接口里申明的全部方法，除非该类是抽象类。（ T ）
12. 接口的本质是类。（ T ）
13. 接口中只有常量，没有变量；只有抽象方法，并且全部都是public方法。（ T ）
14. 接口并不要求实现类和接口在概念本质上一致的，仅仅是实现了接口定义的约定或者能力而已。接口定义了“做什么”，而实现类负责完成“怎么做”，体现了功能（规范）和实现分离的原则。（T）
15. 接口中的方法默认为公有的抽象方法。（ T ）
16. 抽象类中implements关键词可简写为import。（ N ）
17. 接口里可定义变量，并且变量的值不能修改。（ T ）
18. 接口不能实例化。（T）
19. 接口可以被多重implements，抽象类只能被单一extends（T）
20. 接口中只能定义常量和抽象方法（T）
21. 所有接口都继承自Object类（F）

五、简答题

# 1、解释接口回调？

答案：接口回调是多态的另一种体现，是指：可以把使用某一接口的类创建的对象的引言赋给该接口声明的接口变量中，那么该接口变量就可以调用被类实现的接口中的方法，当接口变量调用被类实现的接口中的方法时，就是通知相应的对象调用接口的方法，这一过程为对象功能的接口回调。

2、abstract class和interface有什么区别。(分别写出2条即可)（5分）

答案：声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义static final成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，instanceof 运算符可以用来决定某对象的类是否实现了接口。

抽象类与接口都用于抽象，但是抽象类(JAVA中)可以有自己的部分实现，而接口则完全是一个标识(同时有多重继承的功能)。

3、接口与类相似点是什么（5分）

答案：一个接口可以有多个方法。

接口文件保存在 .java 结尾的文件中，文件名使用接口名。

接口的字节码文件保存在 .class 结尾的文件中。

接口相应的字节码文件必须在与包名称相匹配的目录结构中。

4、接口与类的区别是什么（5分）

答案：接口不能用于实例化对象。

接口没有构造方法。

接口中所有的方法必须是抽象方法。

接口不能包含成员变量，除了 static 和 final 变量。

接口不是被类继承了，而是要被类实现。

接口支持多继承。

5、接口特性是什么（5分）

答案：接口中每一个方法也是隐式抽象的,接口中的方法会被隐式的指定为 public abstract（只能是 public abstract，其他修饰符都会报错）。

接口中可以含有变量，但是接口中的变量会被隐式的指定为 public static final 变量（并且只能是 public，用 private 修饰会报编译错误）。

接口中的方法是不能在接口中实现的，只能由实现接口的类来实现接口中的方法。

6、抽象类和接口的区别（5分）

答案：1. 抽象类中的方法可以有方法体，就是能实现方法的具体功能，但是接口中的方法不行。

2. 抽象类中的成员变量可以是各种类型的，而接口中的成员变量只能是 public static final 类型的。

3. 接口中不能含有静态代码块以及静态方法(用 static 修饰的方法)，而抽象类是可以有静态代码块和静态方法。

4. 一个类只能继承一个抽象类，而一个类却可以实现多个接口。

7、重写接口中声明的方法时，需要注意些什么（5分）

答案：类在实现接口的方法时，不能抛出强制性异常，只能在接口中，或者继承接口的抽象类中抛出该强制性异常。

类在重写方法时要保持一致的方法名，并且应该保持相同或者相兼容的返回值类型。

如果实现接口的类是抽象类，那么就没必要实现该接口的方法。

8、接口和抽象类的联系和区别。（5分）

答案：抽象类：

（1）一个类中有抽象方法，这个类就变成了抽象类。

（2）抽象类中class的前面必须有abstract修饰符。

（3）抽象类中可以有普通方法，也可以有抽象方法，而抽象方法的个数可以是0个，也可以是多个。

（4）子类继承父类，必须重写全部的抽象方法，除非这个类也变成了抽象类。

接口：

（1）表面上看，接口是一种特殊的抽象类，但是类是类，接口是接口，是并列的关系。

（2）接口中所有方法都必须是抽象的。

（3）接口中方法定义默认为public abstract类型，成员变量默认为public static final 类型。（如果省略，系统会默认补全）。

区别主要体现在：

（1）抽象类可以有构造方法，接口中不能有构造方法。

（2）抽象类中可以有成员变量，接口中没有成员变量。（被final修饰变成了常量）

（3）抽象类中可以有普通方法，接口中所有方法都必须是抽象的。

（4）抽象类中抽象方法的访问类型可以是public，protected，但接口中抽象方法的访问类型只能是public，并且默认为public abstract（省略则自动默认补全）。

（5）抽象类中可以有静态方法，接口中不能有静态方法。

9、请简述接口成员特点（5分）

答案： 成员变量；只能是常量，并且是静态的。

默认修饰符：public static final

建议：自己手动给出。

构造方法：接口没有构造方法。

成员方法：只能是抽象方法。

默认修饰符：public abstract

建议：自己手动给出。

10、请列举jdk1.8相对于之前版本的jdk分别有什么新特性。(分别写出2条即可)（5分）

答案：

Jdk1.8: Java 8允许我们给接口添加一个非抽象的方法实现，只需要使用 default关键字即可，这个特征又叫做扩展方法

java 8 允许你使用 :: 关键字来传递方法或者构造函数引用

11、接口中方法的修饰符都有哪些？属性的修饰符有哪些？（5分）

12、接口的作用是什么？简述接口与类的关系

13、请写出下列输出结果。

interface A {

int x = 1;

void showX();

}

interface B {

int y = 2;

void showY();

}

class InterfaceTest implements A, B {

int z = 3;

public void showX() {

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

}

public void showY() {

System.out.println("y=" + y);

}

public void showMe() {

System.out.println("z=" + (z + x + y));

}

}

public class Class1 {

public static void main(String[] args) {

InterfaceTest myObject = new InterfaceTest();

myObject.showX();

myObject.showY();

myObject.showMe();

}

}

X=1

Y=2

Z=6

六、程序实现题

1、定义一个ShowMessage接口，什么方法得名字为showTradeMark；定义两个类TV与PC分别实现这个接口，接口方法showTradeMar实现内容是：一个打印“我是电视”，一个打印“我是电脑”；在main方法中使用接口回调分别执行showTradeMark方法

|  |
| --- |
| **public** **interface** ShowMessage {  **void** showTradeMark();  }  **public** **class** TV **implements** ShowMessage {  @Override  **public** **void** showTradeMark() {  // **TODO** Auto-generated method stub  System.***out***.println("我是电视");  }  }  **public** **class** PC **implements** ShowMessage {  @Override  **public** **void** showTradeMark() {  // **TODO** Auto-generated method stub  System.***out***.println("我是电脑");  }  }  **public** **class** Test {  **public** **static** **void** main(String[] args) {  ShowMessage sm;  sm=**new** TV();  sm.showTradeMark();  sm=**new** PC();  sm.showTradeMark();  }  } |

2、创建Person接口(即“人”)，它有setData()和getData()方法对“人”属性name、sex和birthday赋值和获得这些属性组成的字符串信息。创建类Student实现Person接口，并对自己的“学生”属性的成员变量sID、speciality设置值和获得它们值所组成的字符串信息。（10分）

|  |
| --- |
| People.java(接口)  package com.interfaces;    public interface People {  public void setData(String name,String sex,String birthday);  public String getData();  }    Student.java(实现接口的类)  package com.interfaces.impl;    import com.interfaces.People;    public class Student implements People {  private String name;  private String sex;  private String birthday;  private String sID="2014";  private String speciality="写代码";  @Override  public void setData(String name, String sex, String birthday) {  // TODO Auto-generated method stub  this.name=name;  this.sex=sex;  this.birthday=birthday;  }    @Override  public String getData() {  // TODO Auto-generated method stub  return "名字: " + name + ",性別: " + sex + ",生日: " + birthday + ",ID: " + sID + ",专长: " +speciality;  }    public static void main(String[] args) {  // TODO Auto-generated method stub  People Yantai=new Student();  Yantai.setData("海哥", "男", "2017年2月19日");  System.out.println(Yantai.getData());    }    } |

3、设计一个接口, 并设计一个实现类,实现这个接口

（10分）

|  |
| --- |
| package TT;    interface 银行标准{    int MONEY=9900000;    void 取钱();    void 存钱();  }    class 工商银行 implements 银行标准{    public void 取钱(){      System.out.println("钱进银行了");      }      @Override//全部方法都拿来    public void 存钱() {      System.out.println("钱出去了");      }    }  public class Test {    public static void main(String[]args){       银行标准 bank=new 工商银行();       bank.取钱();       bank.存钱();         System.out.println("银行的老底是"+银行标准.MONEY);//此处银行标准用bank会报错,The static field 银行标准.MONEY should be accessed in a static way      }    } |

4、设计一个类,可以同时实现多个接口

（10分）

|  |
| --- |
| package TT;    interface 建筑标准 {    void kaimen();      void guanmen();  }    interface 银行标准 {    int MONEY = 9999;      void 取钱();      void 存钱();  }    class 工商银行 implements 银行标准,建筑标准 {      @Override    public void kaimen() {      System.out.println("我能开门");      }      @Override    public void guanmen() {      // TODO Auto-generated method stub      System.out.println("我能关门");      }      @Override    public void 取钱() {      // TODO Auto-generated method stub      System.out.println("我能取钱");    }      @Override    public void 存钱() {      // TODO Auto-generated method stub      System.out.println("我能存钱");      }    }    public class Test2 {    public static void main(String[] args) {      工商银行 bank = new 工商银行();      bank.取钱();      bank.存钱();      System.out.println("銀行的底是" + 工商银行.MONEY);        bank.kaimen();      bank.guanmen();    }  } |

5、设计一个类,在继承一个类的同时,实现其的两个接口(具体是什么接口自行设计)

（10分）

|  |
| --- |
| package TT;  interface 建筑标准{    void kaimen();    void guamen();  }  interface 银行标准  {    int MONEY=900000; //接口中的变量,全相当于public static final 类型的    void 取钱();    void 存钱();  }      class 金融机构{    void 货币流通(){      System.out.println("货币流通");    }  }  class 工商银行 extends 金融机构 implements 银行标准, 建筑标准 {      public void 取钱() {      System.out.println("钱进了银行");    }        public void 存钱() {      System.out.println("钱从银行进行个人手里");    }        @Override    public void kaimen() {      System.out.println("银行有个大门,能打开");      }        @Override    public void guamen() {      System.out.println("银行有个铁门,关了以后谁都进不来了");      }    }    class Test4{    public static void main(String[] args) {      工商银行 bank=new 工商银行();      bank.取钱() ;      bank.存钱() ;      System.out.println("银行的底是"+bank.MONEY);        bank.kaimen();      bank.guamen();        bank.货币流通();      }  } |

6、写一个方法对任意引用数据类型数组进行排序。具体要求如下:

1) 方法声明为public void sortArr(Object arr[]){ }

2) 方法中首先输出排序前数组内容，然后进行排序，最后输出排序后数组内容。

3) 可以是冒泡排序或其他算法实现，不直接调用Java提供的方法实现排序。。

（10分）

|  |
| --- |
| 思路：任意类实现Comparable接口来实现该引用数据类型的元素排序，在sort()方法中将Object强转成Comparable实现两个对象的比较。：  public void sort(Object[] arr)  {  //输出排序前数组  for (int i = 0; i < arr.length; i++)  {  System.out.println(arr[i]);  }  //大循环，一共n个元素，达到最终有序，至多n-1趟比较  Object temp;  for (int i = 0; i < arr.length - 1; i++)  {  //定义一个符号量  boolean flag = true;//没有交换，有序  //小循环  for (int j = 0; j < arr.length - 1 - i; j++)  {  //如果前一个大于后一个  Comparables c1 = (Comparables) arr[j];  Comparables c2 = (Comparables) arr[j + 1];  if (c1.compareTo(c2) > 0)  {  //交换  temp = arr[j];  arr[j] = arr[j + 1];  arr[j + 1] = temp;  //修改符号量  flag = false;//交换过，无序  }  }  //判断经过此趟循环后是否已经有序  if (flag)  {  break;//退出外层循环，省略后续趟循环  }  }  //输出排序后数组  for (int i = 0; i < arr.length; i++)  {  System.out.println(arr[i]);  }  } |

7、编写程序，描写影视歌三栖艺人

思路：演电影的接口-----方法：演电影

  演电视剧的接口-----方法：演电视剧

  唱歌的接口-----方法：唱歌

  定义艺人类实现多个接口

（10分）

|  |
| --- |
| **public** **interface** Filmable{  **public** **void** actFilm();  }  **public** **interface** Singable{  **public** **void** sing();  }  **public** **interface** TVable{  **public** **void** showTV();  }  **public** **class** Star **implements** Filmable, TVable, Singable {  **public** **void** sing(){  System.***out***.println("我能唱歌");  }  **public** **void** showTV(){  System.***out***.println("我能演电视剧");  }  **public** **void** actFilm(){  System.***out***.println("我能演电影唱歌");  }  } |

8、多继承： 定义一个USB的接口，并编写一个读取资料的方法 定义相机和U盘的实现类，实现读取方法 当用户插入相机，则打印“正在读取相机内容” 当用户插入U盘，则打印“正在读取U盘内容”

（10分）

|  |
| --- |
| public interface Usb {  public void read();  }  public class Phone implements Usb {  public static void main(String[] args) {  // TODO Auto-generated method stub  Phone phone=new Phone();  phone.read();  }  @Override  public void read() {  // TODO Auto-generated method stub  System.out.println("正在读取相机内容");  }  }  public class Upan implements Usb {  public static void main(String[] args) {  // TODO Auto-generated method stub  Upan upan=new Upan();  upan.read();  }  @Override  public void read() {  // TODO Auto-generated method stub  System.out.println("正在读取U盘内容");  }  } |

9、编写一个接口程序，其中定义一个计算体积的方法。然后，在设计应用程序实现这个接口，分别计算矩形柱面体积和圆形柱面体积。（10分）

|  |
| --- |
| import java.util.\*;//导入扫描仪；  public class clown {  public static void main(String[] args) {  Scanner wsq = new Scanner (System.in);  System.out.println("请以次输入矩形柱面的长，宽，高。");  double x = wsq.nextDouble();  double y = wsq.nextDouble();  double z = wsq.nextDouble();  rectangle WSQ = new rectangle();  WSQ.rectangle(x,y,z);  double yy;  yy = WSQ.volume();  System.out.println("矩形柱面体积：" + yy);  System.out.println("------------------------------------");  System.out.println("请以次输入圆形柱面的半径，高。");  double a = wsq.nextDouble();  double b = wsq.nextDouble();  ball YY = new ball();  YY.ball(a,b);  double sq ;  sq = YY.volume();  System.out.println("圆形柱面体积：" + sq);  }  }  interface volume{  public double volume();//定义一个接口；  }  class rectangle implements volume {//实现接口的其中一个计算矩形柱面体积的类；  double x,y,z;  public void rectangle(){//无参构造；  }  public void rectangle (double x,double y,double z){//有参构造；  this.x = x;  this.y = y;  this.z = z;  }  public double volume(){//重写接口这里的public不可少；  double s;  s= x \* y \* z;  return s;  }  }  class ball implements volume{//实现接口的其中一个计算圆形柱面体积的类；  double a,b;  public void ball(){//无参构造；  }  public void ball(double a, double b){//有参构造；  this.a = a;  this.b = b;  }  public double volume(){//重写接口这里的public不可少；  double w;  w = 3.14 \* a \* a \* b ;  return w;  }  } |

**10、**编写一个完整的Java  Application 程序。包含接口ShapeArea，类Circle、Test，具体要求如下：

（1）接口ShapeArea：

 接口方法

double  getArea()：求一个形状的面积 double  getPerimeter ()：求一个形状的周长

（2）类Circle：

实现ShapeArea接口，并有以下属性和方法：

 ① 属性 radius：  double类型，表示圆的半径

② 方法

Circle(double  r)：构造函数

toString()方法 ：输出圆的描述信息，如“radius=1.0,

perimeter=6.28, area=3.14”

（3）Test类作为主类要完成测试功能

① 生成Circle对象

② 调用对象的toString方法，输出对象的描述信息

（10分）

|  |
| --- |
| / File: Test.java  //下面是具体的代码,要在命令提示符中运行,再者.记得给分哦!!\_^\_  //把下面的代码复制到同一个文件中,把名字改为:Test.java   然后编译,运行就可以了~  interface ShapeArea{   double getArea();   double getPerimeter(); }  class Rectangle implements ShapeArea{      double width;      double height;       public Rectangle(double w, double h){        this.width=w;        this.height=h;      }       public String toString(){         String s="width="+this.width+"\n";               s+="Height="+this.height+"\n";                s+="Perimeter="+this.getPerimeter()+"\n";               s+="Area="+this.getArea()+"\n";             return s;      }       public double getPerimeter(){         return 2\*(width+height);       }       public double getArea(){         return height\*width;      } }  public class Test{      public static void main(String args[]){         if(args.length<2)             System.out.println("请在命令行输入两参数！");          double width=Integer.parseInt(args[0]);          double height=Integer.parseInt(args[1]);          Rectangle rec=new Rectangle(width,height);          System.out.println("该矩形的信息如下：");          System.out.println(rec);     } } |