**中软国际考题-笔试**

1. 单选题

1、给出以下代码，程序的执行结果是？（B）

interface Base {

int k = 0;

}

public class Example implements Base {

public static void main(String[] args) {

int i;

Example exm = new Example();

i = exm.k;

i = Example.k;

i = Base.k;

System.out.println(i);

}

}

A. 无内容输出 B. 打印输出0 C. 代码编译失败 D. 代码运行时输出异常信息

2、下列语句序列执行后X的值是（B）

Int a=5,b=6,x=6;

If(++a==b) x=++a\*b;

A.33 B.42 C.35 D.36

3、以下选项中关于实体类的说法错误的是（B）

A．在分层结构中，不同层之间通过实体类来传输数据

B．实体类的属性一般使用public修饰

C．实体类的方法一般使用public修饰

D．实体类最好实现java.io.Serializable.支持序列化机制

4、以下选项中关于Java分层开发技术中实体类的说法错误的是（D）

A. 在分层开发中，不同层之间可以通过实体类来传输数据

B. 实体类一般对应一个数据库表，其属性和数据库表的字段对应

C. 实体类最好实现java.io.Serializable，支持序列化机制

D. 实体类的属性和方法一般均使用public修饰

5、为了能够访问对封装的属性的访问和修改，方法往往用哪个修饰符修饰（A）

A、public B、protected

C、private D、都可以

6、下面关于类的说法不正确的是： （ B ）

A. 类是 Java 语言中的一种复合数据类型 B. 类中不包含数据变量和方法

C. 类是对所有具有一定共性的对象的抽象 D. Java 语言的代码必须编写在类中

7.设已声明了一个类A的两个对象al、a2，为了初始化a1和a2，下面语句正确的是：（ D ）A. a1=new(); a2=new(); B. a1=A new(); a2=A new();

C. a1,a2=new A(); D. a1=new A(); a2=new A();

8.为了区分重载中同名的不同方法，要求：（ C ）

A.参数名不同 B.返回值类型不同 C.采用不同的形式参数列表 D.选项 A B C 都对

9、如下哪些不是java的关键字？（ B）

A.const B.NULL C.false D.this

10、以下访问修饰符权限范围最大的是？（A）

A. private B. public C. default(缺省) D. protected

11、下面输出结果不可能是 -46 的是（ A ）

A.Math.abs(-46) B.Math.round(-46) C.Math.ceil(-46) D.Math.floor(-46)

12、下列哪一项没有遵循实体类一般规范：（D）

A、根据你的设计，定义一组你需要的私有属性。

B、根据这些属性，创建它们的setter和getter方法

C、提供带参数的构造器和无参数的构造器。

D、命名可以随便命名

13、以下哪个是有关封装优点的正确描述？（C）

A.只需要一个public方法

B.从任何方法中没有异常抛出

C.可以不需要改变接口来改变实现，以达到外部使用代码无需变动

D.可以不需要改变实现来改变接口，已达到外部使用代码无需变动

1. 多选题
2. 下面属于java中常见的实体类对象的是？（A、B、C、D、E）
3. PO B. VO; C. DAO D. BO; E. POJO

2、下面哪些项编译会出错？（A、B）

A. float[]=new float(3) B. float f2[]=new float[]; C. float[]f1=new float[3]

D. float f3[]=new float[3]; E. float f5[]={1.0f,2.0f,2.0f}

E. float f4[]=new float[]{1.0f,2.0f,3.0f}

2、以下说法不正确的是？（A、B、C）

A．不需要定义类，就能创建对象

B. 对象一经声明就可以立即使用

C. Java中的方法参数如果是对象，那么形参的变化会原样反应到实参上

D. new操作符动态地为对象按其指定的类型分配内存，并返回该类型的一个引用

3、以下哪几个是访问修饰符有（A、B、C、D）

A. private B. public C. default(缺省)

D. protected E. class

E. final

4、下面哪个Scanner的方法能获取到小数？（ BD ）

A. next() B.nextInt()

B. nextDouble() D.nextFloat()

5、 java实体类的众多理解有哪些？（A、B、C、D、E、F）

A .就是属性类，通常定义在model层里面

B. 一般的实体类对应一个数据表，其中的属性对应数据表中的字段。

C. 说白了就是为了让程序员在对数据库操作的时候不用写SQL语句

D. 就是一个数据库表生成一个类，这样做对数据库操作起来方便

编写代码较少 提高效率 可以使程序员专注逻辑关系

E. 实体类就是把对某一个表的操作全写在一个类中.

F. 在Java开发中经常要定义一些实体类，这些类的定义的好坏会直接影响，编写代码的质量和

6、以下哪些关于构造器的描述是正确的？（B、C、D）

A.子类可以继承父类的构造器

B.如果没有编写构造器，编译器会自动为类提供一个无参的默认构造器

C.构造器都没有返回值

D. 构造器可以抛出异常

1. 填空题（每空1分，共10分）
2. 实体类中的属性一般用\_\_\_\_\_\_\_\_\_\_修饰，同时属性的getter、setter方法 用\_\_\_\_\_\_\_\_\_修饰。(private、public)
3. Java中一个类可以继承\_\_\_\_\_\_个类，实现\_\_\_\_\_\_个接口。(一、多)

3、实体类中都包含 私有属性 和 属性相关的get/set方法

1. Java中可以修饰类的修饰符有\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(default、public)

5、创建Java实体类属性需要（ ）（私有化）

6、 访问修饰符有哪些：private、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_、public。(protected，default)

7、Java中的日期类Date位于哪个包下\_\_\_\_\_\_\_\_\_\_。(java.util)

1. Java中所有类的顶层父类是\_\_\_\_\_\_\_\_\_\_。（ Object ）

9、对于一个实体类，属性一般用\_\_\_\_\_\_\_\_修饰，方法一般用\_\_\_\_\_\_\_\_修饰。(private，public)

10、Java中的访问控制权限包括4个访问修饰符是：\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_ 。(public、protected、private、缺省)

1. 判断题
2. PO :(persistant object )是持久对象**，**可以看成是与数据库中的表相映射的java对象 。（ T ）
3. 实体类如果与数据库表映射时，类名应避免与数据库表名称相同或类似，容易引起歧义（F）
4. 在Java的方法中定义一个常量要用const关键字。（F）
5. java中实体类继承抽象类后不是实体类（F）
6. 在一个java文件中，可以有多个public的类存在。（ F ）
7. 类中某个成员变量访问修饰符是private，该变量是否可以被外部直接访问。（ F ）
8. 类名和该类文件的名字是否可以不同？（F）
9. 按照类名规范是否可以使用关键字，如int,String.（F）
10. 类名使用中文不会报错？（T）
11. 大驼峰法则是首写字母为小写（F）
12. 一个源文件中可以包含多个类（T）
13. 一个类中定义的方法数量不能超过1024个（F）
14. 一个类源文件大小不能超过1024KB（F）
15. 类名可以使用全数字（F）
16. Private定义的方法可以在子类中访问（F）
17. 实体类实现封装之后，需要为其添加公有get和set方法。（ T ）
18. 实体类需要必须重写父类的toString方法，已达到将对象信息以字符串形式展示( F )
19. 创建实体类，有助于开发的解耦合，提高开发效率。（ T ）
20. static不能修饰方法中定义的变量。（ T ）
21. static方法只能访问静态的成员变量。（ T ）

五、简答题

1、请简单描述POJO是什么，主要的应用场景是什么？

答案：

POJO :(Plain Old Java Objects)，简单的Java对象

实际就是普通JavaBeans,使用POJO名称是为了避免和EJB混淆起来, 而且简称比较直接.

其中有一些属性及其getter、setter方法的类,有时可以作为value object或dto(Data Transform Object)来使用.当然,如果你有一个简单的运算属性也是可以的,但不允许有业务方法,也不能携带有connection之类的方法。

2、java实体类中serialVersionUID的作用是什么？

答案：

serialVersionUID适用于Java的序列化机制。简单来说，Java的序列化机制是通过判断类的serialVersionUID来验证版本一致性的。

在进行反序列化时，JVM会把传来的字节流中的serialVersionUID与本地相应实体类的serialVersionUID进行比较，如果相同就认为是一致的，可以进行反序列化，否则就会出现序列化版本不一致的异常，即是InvalidCastException。。

3、什么是隐藏与数据封装？

答案：

数据隐藏：使用private定义的成员变量，只能在成员方法中使用，其它方法中禁止使用。 优点：保证对象中数据的一致性。

封装：基本数据和对数据进行的操作方法的结合。优点：

1、隐藏类中具体实现的细节；

2、强迫程序员使用统一的接口访问数据；

3、使代码可维护性好

4、java中实体类的的作用？

答案：[实体类](https://www.baidu.com/s?wd=%E5%AE%9E%E4%BD%93%E7%B1%BB&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)主要是作为[数据管理](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E7%AE%A1%E7%90%86&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)和[业务逻辑](https://www.baidu.com/s?wd=%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)处理层面上存在的类别； 它们主要在分析阶段区分 实体类的主要职责是存储和管理系统内部的信息，它也可以有行为，甚至很复杂的行为，但这些行为必须与它所代表的实体对象密切相关。

5、抽象类是否可以继承实体类?

答案：可以继承，但是和实体类的继承一样，也要求父类可继承，并且拥有子类可访问到的构造器。

6、类的结构有哪些？

答案：属性、方法、构造方法、块、内部类

7、Overload和Override的区别

答案：方法的重写Overriding和重载Overloading是Java多态性的不同表现。重写Overriding是父类与子类之间多态性的一种表现，重载Overloading是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被"屏蔽"了。

8、 int 和 Integer 有什么区别

答案：Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int是java的原始数据类型，Integer是java为int提供的封装类。

9、 面向对象特性

答案：封装，继承，多态和抽象。

封装给对象提供了隐藏内部特性和行为的能力。对象提供一些能被其他对象访问的方法来改变它内部的数据。

多态是编程语言给不同的底层数据类型做相同的接口展示的一种能力。一个多态类型上的操作可以应用到其他类型的值上面。

继承给对象提供了从基类获取字段和方法的能力

抽象是把想法从具体的实例中分离出来的步骤

10、 自动装箱

答案：自动装箱是Java编译器在基本数据类型和对应的对象包装类型之间做的一个转化。比如：把int转化成Integer，double转化成double，等等。反之就是自动拆箱。

11、 Java中的两种异常类型是什么？他们有什么区别？

答案：Java中有两种异常：受检查的(checked)异常和不受检查的(unchecked)异常。不受检查的异常不需要在方法或者是构造函数上声明，就算方法或者是构造函数的执行可能会抛出这样的异常，并且不受检查的异常可以传播到方法或者是构造函数的外面。

12、 构造方法及其作用

答案：构造方法作用就是对类进行初始化。 如果你没有定议任何构造方法的形式，程式会为你取一个不带任何参数的构造函数，那么你产生类的对像时只能用不带参数的方法，构造方法就是与类同名的那个方法，它的作用是可以用来初始化

13、 构造方法的作用

答案：构造出来一个类的实例。对构造出来的一个类的实例（对象）初始化.

14、 构造方法的特征

答案：与所存在的类同名，无返回类型（void也没有）

15、4种访问修饰符修饰符区别？

答案：

Public：对任何人都是可用的

Protect：继承的类可以访问以及和private一样的权限

Default：包访问权限，即在整个包内均可被访问

Private：除类型创建者和类型的内部方法之外的任何人都不能访问的元素

16、String,StringBuffer,StringBuilder的区别？（5分）

答案：

String是可不变字符串，StringBuffer和StringBuilder是可变字符串

StringBuilder 线程快，不安全

StringBuffer线程慢，安全

17、一个java实体类书写时，注意事项有哪些？：

1、实体类名称尽量和数据库表名对应

2、可以实现 java.io.Serializable 接口，有个属性 serialVersionUID

3、实体类有属性（private）和方法（public），属性对应数据库中表的字段，方法主 要有getter和setter方法

4、默认有一个无参的构造方法

5、属性若为 boolean 类型，命名时尽量不要以 is 开头，否则自动生成 getter/setter 方法时，与预期会有一定的差异，大量操作时，容易忽略纠正（或不纠正使用时 要注意下）；

18、面向对象的特征有哪些方面?

答案：1)抽象：抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2)继承：继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。

3)封装：封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

4)多态性：多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

19、verload和Override的区别。Overloaded的方法是否可以改变返回值的类型?

答案：如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被"屏蔽"了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded的方法是可以改变返回值的类型。

20、abstract class和interface有什么区别？

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)。

21、作用域public,private,protected,以及不写时的区别？

答案：public：不受任何限制，本类或非本类均可随意访问。

Protected:本类及其子类可以访问（父子友好），同一个包中的其它类也可访问（包内友好).

Private;只有本类可以访问，其余都不可以。

缺省；只有相同包中的类可以访问（包内友好）。

六、程序实现题

1、POJO实体类：创建一个用户的实体类（Person），包含私有属性用户名、密码、生日。

|  |
| --- |
| import java.util.Date;  public class Person {  private String userName;  private String password;  private Date birthday;  public String getUserName() {  return userName;  }  public void setUserName(String userName) {  this.userName = userName;  }  public String getPassword() {  return password;  }  public void setPassword(String password) {  this.password = password;  }  public Date getBirthday() {  return birthday;  }  public void setBirthday(Date birthday) {  this.birthday = birthday;  }  } |

2、用多态实现“打球”，参数为“球”，要求：

传入“篮球”时，打印“打篮球”；

传入“足球”时，打印“踢足球”。

|  |
| --- |
| public class Main {  public static void main(String[] args) {  play(new Basketball());  play(new Football());  }  public static void play(Ball b) {  System.out.println(b.play());  }  }  interface Ball {  public String play();  }  class Basketball implements Ball {  public String play() {  return "打篮球";  }  }  class Football implements Ball {  public String play() {  return "踢足球";  }  } |

3、根据自己所了解到的航班信息设计到的数据，设计航班信息实体类。.

|  |
| --- |
| **import** java.util.Date;  **public** **class** AirInfo {  **private** **int** id;//航班编号  **private** String flight\_number;//航班号  **private** String destination;//目的地  **private** Date flight\_date;//起飞日期  **public** **int** getId() {  **return** id;  }  **public** **void** setId(**int** id) {  **this**.id = id;  }  **public** String getFlight\_number() {  **return** flight\_number;  }  **public** **void** setFlight\_number(String flight\_number) {  **this**.flight\_number = flight\_number;  }  **public** String getDestination() {  **return** destination;  }  **public** **void** setDestination(String destination) {  **this**.destination = destination;  }  **public** Date getFlight\_date() {  **return** flight\_date;  }  **public** **void** setFlight\_date(Date flight\_date) {  **this**.flight\_date = flight\_date;  }  } |

4、有一个超市的会员卡

属性：卡号、手机号、积分、余额

方法：查看卡号、查看手机号、查看积分、查看余额、消费、存钱（存钱时，1元钱累计1积分）

（1）定义属性和方法

（2）消费的时候打折，如果超过1000元打8折；超过2000打5折；1000以内不打折。

（3）写测试类，使用构造方法创建对象，测试所有的方法并体现。

（10分）

|  |
| --- |
| public class SuperCard {    String cardNo;  String mobile;  double points;  double balance;      //查看卡号、查看手机号、查看积分、查看余额、消费、存钱（存钱时，1元钱累计1积分）  String getCardNo(){  return cardNo;  }    String getMobile(){  return mobile;  }    double getPoints(){  return points;  }    double balance(){  return balance;  }    void deposit(double amount){  balance = balance+amount;  points = points+amount;  }    void withdraw(double amount){  //消费的时候打折，如果超过1000元打8折；超过2000打5折；1000以内不打折。  if(amount>2000){  balance = balance-(amount\*0.5);  }else if(amount>1000){  balance = balance-(amount\*0.8);  }else{  balance = balance-amount;  }  }  }  public class Test {    public static void main(String[] args) {  Card c1 = new Card();    //c1对象进行存款2000  c1.deposit(2000);    //输出c1对象的积分  System.out.println(c1.getPoints());    }  } |

5、输出9\*9口诀

//输出九九乘法表

public class JiuJiuBiao {

public static void sum(int n){

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

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

System.out.print(j+"\*"+i+"="+i\*j+" ");

}

System.out.println();

}

}

public static void main(String[] args) {

sum(9);

}

}

6、编写实体类User,包含id，userName.userCode, userPassword

private Integer id;

private String userCode;

private String userName;

private String userPassword;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getUserCode() {

return userCode;

}

public void setUserCode(String userCode) {

this.userCode = userCode;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getUserPassword() {

return userPassword;

}

public void setUserPassword(String userPassword) {

this.userPassword = userPassword;

}

3、从object对象中获得对应的实体类的属性及属性值

Object object = super.selectByEntityId(Util.getSessionbyStation(stationid), Class.forName("com.xx.bgwl.entity." + tablename), id);

String ss= object.getClass().getDeclaredField("commSign").get(object).toString();

Field[] fields = object.getClass().getDeclaredFields();

Field[] fields = object.getClass().getFields();

fields[i].getName()

field.get(object)

4、java Object 转 json

JSONObject json = (JSONObject) JSONObject.wrap(object[0]);

5、json转实体

OrderEntity orderEntity = mapper.readValue(json.toString(), OrderEntity.class);

7、请编写一个用户实体类，包含一下属性（姓名，年龄，性别）

（10分）

|  |
| --- |
| **public** **class** Student {  **private** String name;  **private** int age;  **private** char gender;  //分别进行get,set  } |

8、获取系统时间，并对其进行格式化？

（10分）

|  |
| --- |
| public class Test {  public static void main(String[] args) {  Date d = new Date();  SimpleDateFormat sf = new SimpleDateFormat("yyyy-mm-dd");  String fr = sf.format(d);  System.out.println(fr);  }  } |

9、根据实体类规范来创建一个实体类

（10分）

|  |
| --- |
| **class** Student **implements** Serializable{  /\*\*  \* 版本号  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  //定义的私有属性  **private** **int** id;  **private** String name;  **private** **int** age;  **private** **double** score;  //无参数的构造器 public Student(){    }  //有参数的构造器  **public** Student(**int** id,String name,**int** age, **double** score){  **this**.id = id;  **this**.name = name;  **this**.age = age;  **this**.score = score;  }  //创建的setter和getter方法  **public** **int** getId() {  **return** id;  }  **public** **void** setId(**int** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** **int** getAge() {  **return** age;  }  **public** **void** setAge(**int** age) {  **this**.age = age;  }  **public** **double** getScore() {  **return** score;  }  **public** **void** setScore(**double** score) {  **this**.score = score;  }  //由于id对于学生这个类是唯一可以标识的，所以重写了父类中的id的hashCode()和equals()方法。  @Override  **public** **int** hashCode() {  **final** **int** prime = 31;  **int** result = 1;  result = prime \* result + id;  **return** result;  }  @Override  **public** **boolean** equals(Object obj) {  **if** (**this** == obj)  **return** **true**;  **if** (obj == **null**)  **return** **false**;  **if** (getClass() != obj.getClass())  **return** **false**;  Student other = (Student) obj;  **if** (id != other.id)  **return** **false**;  **return** **true**;  }    } |

10、完成“小商品管理系统”的架构设计。（1. 录入商品基本信息2. 显示全部商品信息）。要求：利用数组存储3个商品对象。

|  |
| --- |
| public class Product {  //属性  private String pid;  private String pname;  private double price;  private int count;  public Product(String pid, String pname, double price, int count) {  super();  this.pid = pid;  this.pname = pname;  this.price = price;  this.count = count;  }  public int getCount() {  return count;  }  public void setCount(int count) {  this.count = count;  }  public String getPid() {  return pid;  }  public void show(){  System.out.println(pid+"\t"+pname+"\t"+price+"\t"+count);  }  }  public class ProductDAO {  //属性  private Product[] products = new Product[3];  //放入商品信息  public void addProduct(int i,Product product){  products[i] = product;  }  //展示所有的商品信息  public void show(){  System.out.println("商品编号\t商品名称\t商品价格\t商品数量");  for(Product pro:products){  pro.show();  }  }  //入库  public boolean add (String id,int num){  //遍历所有商品  for(Product pro:products){  //条件判断 找到了进货商品  if(pro.getPid().equals(id)){  //计算新的数量  int numNew = pro.getCount()+num;  //更新数量  pro.setCount(numNew);  }  }  return true;  }  //入库  public boolean out(String id,int num){  //遍历所有商品  for(Product pro:products){  //条件判断 找到了出货商品  if(pro.getPid().equals(id)){  //判断库存够不够  if(pro.getCount()>=num){  //计算新的数量  int numNew = pro.getCount()-num;  //更新数量  pro.setCount(numNew);  return true;  }else{  System.out.println("库存不足，请重新输入数量！");  return false;  }  }  }  return true;  }  }  public class ProductDAOTest {  public static void main(String[] args) {  // TODO Auto-generated method stub  //创建商品业务对象  ProductDAO productDAO = new ProductDAO();  Scanner input = new Scanner(System.in);  while(true){  System.out.println("请选择您要进行的操作（1.录入商品2.显示所有商品3.入库4.出库5.退出）：");  String sel = input.nextLine();  if("1".equals(sel)){  for(int i=0;i<3;i++){  System.out.print("请输入商品编号：");  String pid = input.nextLine();  System.out.print("请输入商品名称：");  String pname = input.nextLine();  System.out.print("请输入商品价格：");  String price = input.nextLine();  System.out.print("请输入商品数量：");  String count = input.nextLine();  //创建商品对象  Product product = new Product(pid,pname, Double.parseDouble(price), Integer.parseInt(count));  //放入商品  productDAO.addProduct(i, product);  }  }else if("2".equals(sel)){  productDAO.show();  }else if("3".equals(sel)){  System.out.print("请输入要入库商品编号：");  String pid = input.nextLine();  System.out.print("请输入要入库商品数量：");  String num = input.nextLine();  productDAO.add(pid, Integer.parseInt(num));  }else if("4".equals(sel)){  System.out.print("请输入要出库商品编号：");  String pid = input.nextLine();  System.out.print("请输入要出库商品数量：");  String num = input.nextLine();  productDAO.out(pid, Integer.parseInt(num));  }else if("5".equals(sel)){  System.out.println("退出成功");  break;  }else{  System.out.println("输入有误，请重新操作！");  }  }  }  } |