Java 常见面试题

一、基础知识

1. 同步和异步有何异同，在什么情况下分别使用他们？举例说明。

如果数据将在线程间共享。例如正在写的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。

当应用程序在对象上调用了一个需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

2. final、finally、finalize 的区别。

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

3. 面向对象的特征有哪些方面

⑴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。

⑵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。

⑶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。

⑷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。

4. sleep()和 wait()有什么区别?

sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 sleep 不会释放对象锁。

wait 是 Object 类的方法，对此对象调用 wait 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

5. heap 和 stack 有什么区别。

栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。

堆是栈的一个组成元素

6. int 和 Integer 有什么区别

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。Java 为每个原始类型提供了封装类。

原始类型封装类

booleanBoolean

charCharacter

byteByte

shortShort

intInteger

longLong

floatFloat

doubleDouble

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。

7. &和&&的区别。

&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

8. error 和 exception 有什么区别?

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

9. HashMap 和 Hashtable 的区别。

HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key）,由于非线程安全，效率上可能高于 Hashtable。

HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。

HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。

Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Mapinterface 的一个实现。

最大的不同是，Hashtable 的方法是 Synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供外同步。

Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。

10. forward 和 redirect 的区别

forward 是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。

redirect 就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，所以 session,request 参数都可以获取。

11. String 和 StringBuffer 的区别

JAVA 平台提供了两个类：String 和 StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个 String 类提供了数值不可改变的字符串。而这个 StringBuffer

类提供的字符串进行修改。当你知道字符数据要改变的时候你就可以使用 StringBuffer。典型地，你可以使用 StringBuffers 来动态构造字符数据。

12. StaticNestedClass 和 InnerClass 的不同。

StaticNestedClass 是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。

13. Collection 和 Collections 的区别。

Collection 是集合类的上级接口，继承与他的接口主要有 Set 和 List.

Collections 是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

14. java 中实现多态的机制是什么?

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。

15. List,Set,Map 是否继承自 Collection 接口?

List，Set 是，Map 不是。

16. 构造器 Constructor 是否可被 override?

构造器 Constructor 不能被继承，因此不能重写 Overriding，但可以被重载 Overloading。

17. 数组有没有 length()这个方法?String 有没有 length()这个方法？

数组没有 length()这个方法，有 length 的属性。String 有有 length()这个方法。

18. Java 中的异常处理机制的简单原理和应用。

当 Java 程序违反了 Java 的语义规则时，Java 虚拟机就会将发生的错误表示为一个异常。违反语义规则包括 2 种情况。一种是 Java 类库内置的语义检查。例如数组下标越界,会引发 IndexOutOfBoundsException;访问 null 的对象时会引发 NullPointerException。另一种情况就是 Java 允许程序员扩展这种语义检查，程序员可以创建自己的异常，并自由选择在何时用 throw 关键字引发异常。所有的异常都是 java.lang.Thowable 的子类。

19. 在 java 中一个类被声明为 final 类型，表示了什么意思？

表示该类不能被继承，是顶级类。

20. 运行时异常与一般异常有何异同？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java 编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

21. abstractclass 和 interface 有什么区别?

声明方法的存在而不去实现它的类被叫做抽象类（abstractclass），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 staticfinal 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。

22. String 是最基本的数据类型吗?

基本数据类型包括 byte、int、char、long、float、double、boolean 和 short。

java.lang.String 类是 final 类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用 StringBuffer 类

23. Java 有没有 goto?

java 中的保留字，现在没有在 java 中使用。

24. 在 JAVA 中，如何跳出当前的多重嵌套循环?

用 break;return 方法。

25. Strings=newString("xyz");创建了几个 StringObject?

两个。

26. Java 的接口和 C++的虚类的相同和不同处。

由于 Java 不支持多继承，而有可能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制就不能满足要求。与继承相比，接口有更高的灵活性，因为接口中没有任何实现代码。当一个类实现了接口以后，该类要实现接口里面所有的方法和属性，并且接口里面的属性在默认状态下面都是 publicstatic,所有方法默认情况下是 public.一个类可以实现多个接口。

27. List、Map、Set 三个接口，存取元素时，各有什么特点?

List 以特定次序来持有元素，可有重复元素。Set 无法拥有重复元素,内部排序。Map 保存 key-value 值，value 可多值。

28. GC 是什么?为什么要有 GC?

GC 是垃圾收集的意思（GabageCollection）,内存处理是编程人员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java 提供的 GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，Java 语言没有提供释放已分配内存的显示操作方法。

29. Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型?

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。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。

30. 说出 Servlet 的生命周期，并说出 Servlet 和 CGI 的区别。

Servlet被服务器实例化后，容器运行其init方法，请求到达时运行其service方法，service方法自动派遣运行与请求对应的 doXXX 方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其 destroy 方法。

与 cgi 的区别在于 servlet 处于服务器进程中，它通过多线程方式运行其 service 方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而 CGI 对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于 servlet。

31. 线程同步的方法。

wait():使一个线程处于等待状态，并且释放所持有的对象的 lock。

sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 InterruptedException 异常。

notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切地唤醒某一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级。

Allnotity():唤醒所有处入等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

32. 垃圾回收器的基本原理是什么?垃圾回收器可以马上回收内存吗?有什么办法主动通知虚拟机进行垃圾回收?

对于 GC 来说，当程序员创建对象时，GC 就开始监控这个对象的地址、大小以及使用情况。通常，GC 采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是"可达的"，哪些对象是"不可达的"。当 GC 确定一些对象为"不可达"时，GC 就有责任回收这些内存空间。可以。程序员可以手动执行 System.gc()，通知 GC 运行，但是 Java 语言规范并不保证 GC 一定会执行。

33. 什么是 java 序列化，如何实现 java 序列化?

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

序列化的实现：将需要被序列化的类实现 Serializable 接口，该接口没有需要实现的方法，implementsSerializable 只是为了标注该对象是可被序列化的，然后使用一个输出流(如：FileOutputStream) 来构造一个 ObjectOutputStream(对象流)对象，接着，使用 ObjectOutputStream 对象的 writeObject(Objectobj)方法就可以将参数为 obj 的对象写出(即保存其状态)，要恢复的话则用输入流。

2018

大数据从入门到就业

一场接地气的公开分享课

大数据新手三步走 = 学习技巧 + 求职就业 + 工作提升

王培

大数据开发工程师

参与过中国电信等多个企业的大型大数据项目，项目经验丰富。

张丁全

大数据分析可视化工程师

专注大数据呈现，负责参与过公司多个项目，在大数据可视化上研究颇深

亮点预告

- 零基础学习大数据技巧经验分享
- 专为新人准备的面试求职干货
- 新人进入公司如何快速成长
- 更多干货内容，扫码见公开课介绍

公开直播时间

2018.6.20 晚20:00

线下+线上同步，手机/电脑都能学

长按扫码 了解详情

从入门到就业，一场接地气的公开课，期待你的加入：
http://mk.meeket.com/flyer/991024/914899.html?source=13

34. Math.round(11.5)等於多少?Math.round(-11.5)等於多少?

Math.round(11.5)==12
Math.round(-11.5)==-11
round 方法返回与参数最接近的长整数，参数加 1/2 后求其 floor。

35. 启动一个线程是用 run()还是 start()?

启动一个线程是调用 start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

36. java 中会存在内存泄漏吗，请简单描述。

会。如：intI,i2;return(i-i2);//wheni 为足够大的正数,i2 为足够大的负数。结果会造成溢位，导致错误。

37. try{}里有一个 return 语句，那么紧跟在这个 try 后的 finally{}里的 code 会不会被执行，什么时候被执行，在 return 前还是后?

会执行，在 return 前执行。

38. 接口是否可继承接口?抽象类是否可实现(implements)接口?抽象类是否可继承实体类(concreteclass)?

接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

39. 静态变量和实例变量的区别?

StaticI=10;//常量
classAa;a.i=10;//可变

40. 两个对象值相同(x.equals(y)==true)，但却可有不同的 hashcode，这句话对不对?

不对，有相同的 hashcode。

41. JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？

动态 INCLUDE 用 jsp:include 动作实现 <jsp:include page="included.jsp" flush="true" /> 它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数。

静态 INCLUDE 用 include 伪码实现,定不会检查所含文件的变化，适用于包含静态页面<%@ include file="included.htm" %> 。

42. EJB 与 JAVA BEAN 的区别？

Java Bean 是可复用的组件，对 Java Bean 并没有严格的规范，理论上讲，任何一个 Java 类都可以是一个 Bean。但通常情况下，由于 Java Bean 是被容器所创建（如 Tomcat）的，所以 Java Bean 应具有一个无参的构造器，另外，通常 Java Bean 还要实现 Serializable 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM，即分布式组件。它是基于 Java 的远程方法调用（RMI）技术的，所以 EJB 可以被远程访问（跨进程、跨计算机）。但 EJB 必须被布署在诸如 Webspere、WebLogic 这样的容器中，EJB 客户从不直接访问真正的 EJB 组件，而是通过其容器访问。EJB 容器是 EJB 组件的代理，EJB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

二、编程题

1. 用最有效率的方法算出 2 乘以 8 等于几?

2<<3

2. 用 JAVA 实现一种排序，JAVA 类实现序列化的方法(二种)？如在 COLLECTION 框架中，实现比较要实现什么样的接口？

答:用插入法进行排序代码如下

```
packagetest;
importjava.util.*;
classInsertSort
{
ArrayListal;
publicInsertSort(intnum,intmod)
{
```

```
al=newArrayList(num);
Randomrand=newRandom();
System.out.println("TheArrayListSortBefore:");
for(inti=0;i<num;i++)
{
al.add(newInteger(Math.abs(rand.nextInt())%mod+1));
System.out.println("al["+i+"]="+al.get(i));
}
}
publicvoidSortIt()
{
IntegertempInt;
intMaxSize=1;
for(inti=1;i<al.size();i++)
{
tempInt=(Integer)al.remove(i);
if(tempInt.intValue()>=((Integer)al.get(MaxSize-1)).intValue())
{
al.add(MaxSize,tempInt);
MaxSize++;
System.out.println(al.toString());
}else{
for(intj=0;j<MaxSize;j++)
{
if

(((Integer)al.get(j)).intValue()>=tempInt.intValue())
{
al.add(j,tempInt);
MaxSize++;
System.out.println(al.toString());
break;
}
}
}
}
System.out.println("TheArrayListSortAfter:");
```

```
for(inti=0;i<al.size();i++)
{
System.out.println("al["+i+"]="+al.get(i));
}
}
publicstaticvoidmain(String[]args)
{
InsertSortis=newInsertSort(10,100);
is.SortIt();
}
}
```

3. 现在输入 n 个数字，以逗号，分开；

然后可选择升或者降序排序；
按提交键就在另一页面显示
按什么排序，结果为，
提供 reset

答案：

```
publicstaticString[]splitStringByComma(Stringsource){
    if(source==null||source.trim().equals(""))
        returnnull;
    StringTokenizercommaToker=newStringTokenizer(source,",");
    String[]result=newString[commaToker.countTokens()];
    inti=0;
    while(commaToker.hasMoreTokens()){
        result[i]=commaToker.nextToken();
        i++;
    }
    returnresult;
}
```

循环遍历 String 数组
Integer.parseInt(Strings)变成 int 类型
组成 int 数组
Arrays.sort(int[]a),
a 数组升序
降序可以从尾部开始输出

4. 写一个 Singleton 出来。

Singleton 模式主要作用是保证在 Java 应用程序中，一个类 Class 只有一个实例存在。

一般 Singleton 模式通常有几种种形式：

第一种形式：定义一个类，它的构造函数为 private 的，它有一个 static 的 private 的该类变量，在类初始化时实例话，通过一个 public 的 getInstance 方法获取对它的引用,继而调用其中的方法。

```
publicclassSingleton{
privateSingleton(){}
//在自己内部定义自己一个实例，是不是很奇怪？
//注意这是 private 只供内部调用
privatestaticSingletoninstance=newSingleton();
//这里提供了一个供外部访问本 class 的静态方法，可以直接访问
publicstaticSingletongetInstance(){
returninstance;
}
}
```

第二种形式：

```
publicclassSingleton{
privatestaticSingletoninstance=null;
publicstaticsynchronizedSingletongetInstance(){
//这个方法比上面有所改进，不用每次都进行生成对象，只是第一次
//使用时生成实例，提高了效率！
if(instance==null)
instance＝newSingleton();
returninstance;}
}
```

5. 继承时候类的执行顺序问题,一般都是选择题,问你将会打印出什么?

答:父类：

```
packagetest;
publicclassFatherClass
{
publicFatherClass()
{
System.out.println("FatherClassCreate");
```

```
}
}
```

子类：

```
packagetest;
importtest.FatherClass;
publicclassChildClassextendsFatherClass
{
publicChildClass()
{
System.out.println("ChildClassCreate");
}
publicstaticvoidmain(String[]args)
{
FatherClassfc=newFatherClass();
ChildClasscc=newChildClass();
}
}
```

输出结果：

```
C:>javatest.ChildClass
FatherClassCreate
FatherClassCreate
ChildClassCreate
```

6. Java 的通信编程，用 JAVASOCKET 编程，读服务器几个字符，再写入本地显示？

答：Server 端程序：

```
packagetest;
importjava.net.*;
importjava.io.*;
publicclassServer
{
privateServerSocketss;
privateSocketsocket;
privateBufferedReaderin;
privatePrintWriterout;
publicServer()
{
```

```
try
{
ss=newServerSocket(10000);
while(true)
{
socket=ss.accept();
StringRemoteIP=socket.getInetAddress().getHostAddress();
StringRemotePort=":"+socket.getLocalPort();
System.out.println("Aclientcomein!IP:"+RemoteIP+RemotePort);
in=newBufferedReader(new

InputStreamReader(socket.getInputStream()));
Stringline=in.readLine();
System.out.println("Cleintsendis:"+line);
out=newPrintWriter(socket.getOutputStream(),true);
out.println("YourMessageReceived!");
out.close();
in.close();
socket.close();
}
}catch(IOExceptione)
{
out.println("wrong");
}
}
publicstaticvoidmain(String[]args)
{
newServer();
}
};
```

Client 端程序：

```
packagetest;
importjava.io.*;
importjava.net.*;

publicclassClient
{
```

```
Socketsocket;
BufferedReaderin;
PrintWriterout;
publicClient()
{
try
{
System.out.println("TrytoConnectto127.0.0.1:10000");
socket=newSocket("127.0.0.1",10000);
System.out.println("TheServerConnected!");
System.out.println("PleaseentersomeCharacter:");
BufferedReaderline=newBufferedReader(new

InputStreamReader(System.in));
out=newPrintWriter(socket.getOutputStream(),true);
out.println(line.readLine());
in=newBufferedReader(newInputStreamReader(socket.getInputStream()));
System.out.println(in.readLine());
out.close();
in.close();
socket.close();
}catch(IOExceptione)
{
out.println("Wrong");
}
}
publicstaticvoidmain(String[]args)
{
newClient();
}
};
```

7. 编写一个截取字符串的函数，输入为一个字符串和字节数，输出为按字节截取的字符串。但是要保证汉字不被截半个，如“我 ABC”4，应该截为“我 AB”，输入“我 ABC 汉 DEF”，6，应该输出为“我 ABC”而不是“我 ABC+汉的半个”。

答：代码如下：

```
packagetest;
```

```
classSplitString
{
StringSplitStr;
intSplitByte;
publicSplitString(Stringstr,intbytes)
{
SplitStr=str;
SplitByte=bytes;
System.out.println("TheStringis:′"+SplitStr+"′;SplitBytes="+SplitByte);
}
publicvoidSplitIt()
{
intloopCount;
loopCount=(SplitStr.length()%SplitByte==0)?(SplitStr.length()/SplitByte):(SplitStr.l
ength()/Split
Byte+1);
System.out.println("WillSplitinto"+loopCount);
for(inti=1;i<=loopCount;i++)
{
if(i==loopCount){
System.out.println(SplitStr.substring((i-1)*SplitByte,SplitStr.length()));
}else{
System.out.println(SplitStr.substring((i-1)*SplitByte,(i*SplitByte)));
}
}
}
publicstaticvoidmain(String[]args)
{
SplitStringss=newSplitString("test 中 dd 文 dsaf 中男大 3443n 中国 43 中国人

0ewldfls=103",4);
ss.SplitIt();
}
}
```

2018

大数据从入门到就业

一场接地气的公开分享课

大数据新手三步走 = 学习技巧 + 求职就业 + 工作提升

王培

大数据开发工程师

参与过中国电信等多个企业的大型大数据项目，项目经验丰富。

张丁全

大数据分析可视化工程师

专注大数据呈现，负责参与过公司多个项目，在大数据可视化上研究颇深

亮点预告

- 零基础学习大数据技巧经验分享
- 专为新人准备的面试求职干货
- 新人进入公司如何快速成长
- 更多干货内容，扫码见公开课介绍

公开直播时间

2018.6.20 晚20:00

线下+线上同步，手机/电脑都能学

长按扫码 了解详情

从入门到就业，一场接地气的公开课，期待你的加入：
http://mk.meeket.com/flyer/991024/914899.html?source=13

8. 内部类的实现方式?

答：示例代码如下：

```
packagetest;
publicclassOuterClass
{
privateclassInterClass
{
publicInterClass()
{
System.out.println("InterClassCreate");
}
}
publicOuterClass()
{
InterClassic=newInterClass();
System.out.println("OuterClassCreate");
}
publicstaticvoidmain(String[]args)
{
OuterClassoc=newOuterClass();
}
}
```

输出结果:

```
C:>javatest/OuterClass
InterClassCreate
OuterClassCreate
```

再一个例题：

```
publicclassOuterClass{
privatedoubled1=1.0;
//insertcodehere
}
Youneedtoinsertaninnerclassdeclarationatline3.Whichtwoinnerclassdeclarationsa
re

valid?(Choosetwo.)
A.classInnerOne{
```

```
publicstaticdoublemethoda(){returnd1;}
}
B.publicclassInnerOne{
staticdoublemethoda(){returnd1;}
}
C.privateclassInnerOne{
doublemethoda(){returnd1;}
}
D.staticclassInnerOne{
protecteddoublemethoda(){returnd1;}
}
E.abstractclassInnerOne{
publicabstractdoublemethoda();
}
```