



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0
公众号后台回复“面试手册”

1.什么叫对象？什么叫类？什么面向对象（OOP）？

答案：类的概念：类是具有相同属性和服务的一组对象的集合。它为属于该类的所有对象提供了统一的抽象描述，

其内部包括属性和服务两个主要部分。在面向对象的编程语言中，类是一个独立的程序单位，它应该有一个类名并包括属性说明和服务说明两个主要部分。

对象的概念：对象是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务组成。从更抽象的角度来说，对象是问题域或实现域中某些事物的一个抽象，它反映该事物在系统中需要保存的信息和发挥的作用；它是一组属性和有权对这些属性进行操作的一组服务的封装体。客观世界是由对象和对象之间的联系组成的。

类与对象的关系就如模具和铸件的关系，类的实例化结果就是对象，而对一类对象的抽象就是类。类描述了一组有相同特性（属性）和相同行为（方法）的对象。上面大概就是它们的定义吧，也许你是刚接触面向对象的朋友，不要被概念的东西搞晕了，给你举个例子吧，如果你去中关村想买几台组装的PC机，到了那里你第一步要干什么，是不是装机的工程师和你坐在一起，按你提供的信息和你一起完成一个装机的配置单呀，这个配置单就可以想像成是类，它就是一张纸，但是它上面记录了你要买的PC机的信息，如果用这个配置单买10台机器，那么这10台机器，都是按这个配置单组成的，所以说这10台机器是一个类型的，也可以说是一类的。那么什么是对象呢，类的实例化结果就是对象，用这个配置单配置出来（实例化出来）的机器就是对象，是我们可以操作的实体，10台机器，10个对象。每台机器都是独立的，只能说明他们是同一类的，对其中一个机做任何动作都不会影响其它9台机器，但是我对类修改，也就是在这个配置单上加一个或少一个配件，那么装出来的9个机器都改变了，这是类和对象的关系(类的实例化结果就是对象)。

2.相对于JDK1.4，JDK1.5有哪些新特性？

答案：泛型（Generics）

增强的“for”循环（Enhanced For loop）

自动装箱/自动拆箱（Autoboxing/unboxing）

类型安全的枚举（Type safe enums）

静态导入（Static import）

可变参数（Var args）

3.JAVA 中使用 final 修饰符，对程序有哪些影响？

答案：

1、修饰类

当用 final 修饰一个类时，表明这个类不能被继承。也就是说，如果一个类你永远不会让他被继承，就可以用 final

进行修饰。final 类中的成员变量可以根据需要设为 final，但是要注意 final 类中的所有成员方法都会被隐式地指定为 final 方法。

在使用 final 修饰类的时候，要注意谨慎选择，除非这个类真的在以后不会用来继承或者出于安全的考虑，尽量不

要将类设计为 final 类

2、修饰方法

被 final 修饰的方法将不能被子类覆盖，主要用于 1，把方法锁定，以防任何继承类修改它的含。2，在早期的 Java 实现版本中，会将 final 方法转为内嵌调用，所以效率能够提升

3、修饰变量

对于一个 final 变量，如果是基本数据类型的变量，则其数值一旦在初始化之后便不能更改；如果是引用类型的变

量，则在对其实初始化之后便不能再让其指向另一个对象。

当用 final 作用于类的成员变量时，成员变量（注意是类的成员变量，局部变量只需要保证在使用之前被初始化赋值即可）必须在定义时或者构造器中进行初始化赋值，而且 final 变量一旦被初始化赋值之后，就不能再被赋值了。

4. Java 环境变量 Unix/Linux 下如何配置？

答案：修改/etc/profile 文件当本机仅作为开发使用时推荐使用这种方法，因为此种配置时所有用户的 shell 都

有权使用这些环境变量，可能会给系统带来安全性问题。用文本编辑器打开/etc/profile，在 profile 文件末尾加入：

```
JAVA_HOME=/usr/share/jdk1.5.0_05
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export JAVA_HOME
```

```
export PATH
```

```
export CLASSPATH 重新登录即可
```

5. 写出 5 个你在 JAVA 开发中常用的包含（全名），并简述其作用。

常用的五个：

1) java.lang.*

提供利用 Java 编程语言进行程序设计的基础类。最重要的类是 Object（它是类层次结构的根）和 Class（它的实例表示正在运行的应用程序中的类）。

2) java.util.*

包含集合框架、遗留的 collection 类、事件模型、日期和时间设施、国际化和各种实用工具类（字符串标记生成

器、随机数生成器和位数组、日期 Date 类、堆栈 Stack 类、向量 Vector 类等）。集合类、时间处理模式、日期时间工具等各类常用工具包

3) java.io.*

Java 的核心库 java.io 提供了全面的 IO 接口。包括：文件读写、标准设备输出等。Java 中 IO 是以流为基础进行

输入输出的，所有数据被串行化写入输出流，或者从输入流读入。

4) java.net.*

并非所有系统都支持 IPv6 协议，而当 Java 网络连接堆栈尝试检测它并在可用时透明地使用它时，还可以利用

系统属性禁用它。在 IPv6 不可用或被显式禁用的情况下，InetAddress 对大多数网络连接操作都不再是有效参数。虽然可以保证在查找主机名时 java.net.InetAddress.getByName 之类的方法不返回 InetAddress，但仍然可能通过传递字面值来创建此类对象。在此情况下，大多数方法在使用 InetAddress 调用时都将抛出异常。

5) java.sql.*

提供使用 JavaTM 编程语言访问并处理存储在数据源（通常是一个关系数据库）中的数据的 API。此 API 包括

一个框架，凭借此框架可以动态地安装不同驱动程序来访问不同数据源。

6. 写出 5 个常见的运行时异常 (RuntimeException)

- 1) ClassCastException(类转换异常)
- 2) IndexOutOfBoundsException(数组越界)
- 3) NullPointerException(空指针)
- 4) ArrayStoreException(数据存储异常，操作数组时类型不一致)
- 5) IO 操作的 BufferOverflowException 异常

7. 方法重载 (overload) 需要满足什么条件，方法覆盖/方法重写 (override) 需要满足什么条件？ (二选一)

答案：重载需要满足的条件：在同一类中定义的方法，方法名必须相同，返回类型必须相同，参数一定不同。

发生覆盖的条件：

“三同一不低”，子类和方法名称，参数列表，返回类型必须完全相同，而且子类方法的访问修饰符的权限

不能比父类低。

子类方法不能抛出比父类方法更多的异常。即子类方法所抛出的异常必须和父类方法所抛出的异常一致，或者是

其子类，或者什么也不抛出；

被覆盖的方法不能是 final 类型的。因为 final 修饰的方法是无法覆盖的。

被覆盖的方法不能为 private。否则在其子类中只是新定义了一个方法，并没有对其进行覆盖。

被覆盖的方法不能为 static。所以如果父类中的方法为静态的，而子类中的方法不是静态的，但是两个方法除了这

一点外其他都满足覆盖条件，那么会发生编译错误。反之亦然。即使父类和子类中的方法都是静态的，并且满足覆盖条件，但是仍然不会发生覆盖，因为静态方法是在编译的时候把静态方法和类的引用类型进行匹配。

重写规则：重写方法不能比被重写方法限制有更严格的访问级别。（但是可以更广泛，比如父类方法是包访问权

限，子类的重写方法是 public 访问权限。）比如：Object 类有个 toString() 方法，开始重写这个方法的时候我们总容易忘记 public 修饰符，编译器当然不会放过任何教训我们的机会。出错的原因就是：没有加任何访问修饰符的方法具有包访问权限，包访问权限比 public 当然要严格了，所以编译器会报错的。参数列表必须与被重写方法的相同。重写有个孪生的弟弟叫重载，也就是后面要出场的。如果子类方法的参数与父类对应的方法不同，那么就是你认错人了，那是重载，不是重写。返回类型必须与被重写方法的返回类型相同。

父类方法 A：void eat(){} 子类方法 B：int eat(){} 两者虽然参数相同，可是返回类型不同，所以不是重写。

父类方法 A：int eat(){} 子类方法 B：long eat(){} 返回类型虽然兼容父类，但是不同就是不同，所以不是重写。

8.继承 (inheritance) 的优缺点是什么？

优点：

新的实现很容易，因为大部分是继承而来的。很容易修改和扩展已有的实现

缺点：

打破了封装，因为基类向子类暴露了实现细节，白盒重用，因为基类的内部细节通常对子类是可见的，当父类的

实现改变时可能要相应的对子类做出改变，不能在运行时改变由父类继承来的实现。由此可见，组合比继承具有更大的灵活性和更稳定的结构，一般情况下应该优先考虑组合。只有当下列条件满足时才考虑使用继承：子类是一种特殊的类型，而不只是父类的一个角色，子类的实例不需要变成另一个类的对象子类扩展，而不是覆盖或者使父类的功能失效。

9.为什么要使用接口和抽象类？

Java 接口和 Java 抽象类代表的就是抽象类型，就是我们需要提出的抽象层的具体表现。OOP 面向对象的编程，

如果要提高程序的复用率，增加程序的可维护性，可扩展性，就必须是面向接口的编程，面向抽象的编程，正确地使用

接口、抽象类这些太有用的抽象类型做为你结构层次上的顶层。

1、Java 接口和 Java 抽象类最大的一个区别，就在于 Java 抽象类可以提供某些方法的部分实现，而 Java 接口不

可以，这大概就是 Java 抽象类唯一的优点吧，但这个优点非常有用。如果向一个抽象类里加入一个新的具体方法时，那么它所有的子类都一下子都得到了这个新方法，而 Java 接口做不到这一点，如果向一个 Java 接口里加入一个新方法，所有实现这个接口的类就无法成功通过编译了，因为你必须让每一个类都再实现这个方法才行。

2、一个抽象类的实现只能由这个抽象类的子类给出，也就是说，这个实现处在抽象类所定义出的继承的等级结构

中，而由于 Java 语言的单继承性，所以抽象类作为类型定义工具的效能大打折扣。在这一点上，Java 接口的优势就出来了，任何一个实现了一个 Java 接口所规定的方法的类都可以具有这个接口的类型，而一个类可以实现任意多个 Java 接口，从而这个类就有了多种类型。

3、从第 2 点不难看出，Java 接口是定义混合类型的理想工具，混合类表明一个类不仅仅具有某个主类型的行为，而且具有其他的次要行为。

4、结合 1、2 点中抽象类和 Java 接口的各自优势，具精典的设计模式就出来了：声明类型的工作仍然由 Java 接

口承担，但是同时给出一个 Java 抽象类，且实现了这个接口，而其他同属于这个抽象类型的具体类可以选择实现这个 Java 接口，也可以选择继承这个抽象类，也就是说在层次结构中，Java 接口在最上面，然后紧跟着抽象类，哈，这下两个的最大优点都能发挥到极至了。这个模式就是“缺省适配模式”。在 Java 语言 API 中用了这种模式，而且全都遵循一定的命名规范：Abstract + 接口名。

Java 接口和 Java 抽象类的存在就是为了用于具体类的实现和继承的，如果你准备写一个具体类去继承另一个具

体类的话，那你的设计就有很大的问题了。Java 抽象类就是为了继承而存在的，它的抽象方法就是为了强制子类必须去实现的。

使用 Java 接口和抽象 Java 类进行变量的类型声明、参数是类型声明、方法的返还类型说明，以及数据类型的转

换等。而不要用具体 Java 类进行变量的类型声明、参数是类型声明、方法的返回类型说明，以及数据类型的转换等。

我想，如果你编的代码里面连一个接口和抽象类都没有的话，也许我可以说你根本没有用到任何设计模式，任何一个设计模式都是和抽象分不开的，而抽象与 Java 接口和抽象 Java 类又是分不开的。

接口的作用，一言以蔽之，就是标志类的类别。把不同类型的类归于不同的接口，可以更好的管理他们。把一组看

如不相关的类归为一个接口去调用。可以用一个接口型的变量来引用一个对象，这是接口我认为最大的作用。

10.什么是自定义异常？如何自定义异常？

参考：<https://www.cnblogs.com/AlanLee/p/6104492.html>

11.Set，List，Map 有什么区别？

结构特点

List 和 Set 是存储单列数据的集合，Map 是存储键和值这样的双列数据的集合；List 中存储的数据是有顺序，并

且允许重复；Map 中存储的数据是没有顺序的，其键是不能重复的，它的值是可以有重复的，Set 中存储的数据是无序的，且不允许有重复，但元素在集合中的位置由元素的 hashCode 决定，位置是固定的（Set 集合根据 hashCode 来进行数据的存储，所以位置是固定的，但是位置不是用户可以控制的，所以对于用户来说 set 中的元素还是无序的）；

实现类

List 接口有三个实现类（LinkedList：基于链表实现，链表内存是散乱的，每一个元素存储本身内存地址的同时还

存储下一个元素的地址。链表增删快，查找慢；ArrayList：基于数组实现，非线程安全的，效率高，便于索引，但不便于插入删除；Vector：基于数组实现，线程安全的，效率低）。

Map 接口有三个实现类（HashMap：基于 hash 表的 Map 接口实现，非线程安全，高效，支持 null 值和 null

键；HashTable：线程安全，低效，不支持 null 值和 null 键；LinkedHashMap：是 HashMap 的一个子类，保存了记录的插入顺序；SortMap 接口：TreeMap，能够把它保存的记录根据键排序，默认是键值的升序排序）。

Set 接口有两个实现类（HashSet：底层是由 HashMap 实现，不允许集合中有重复的值，使用该方式时需要重

写 equals()和 hashCode()方法；LinkedHashSet：继承与 HashSet，同时又基于 LinkedHashMap 来进行实现，底层使用的是 LinkedHashMap）

区别

List 集合中对象按照索引位置排序，可以有重复对象，允许按照对象在集合中的索引位置检索对象，例如通过

list.get(i)方法来获取集合中的元素；Map 中的每一个元素包含一个键和一个值，成对出现，键对象不可以重复，值对象可以重复；Set 集合中的对象不按照特定的方式排序，并且没有重复对象，但它的实现类能对集合中的对象按照特定的方式排序，例如 TreeSet 类，可以按照默认顺序，也可以通过实现 Java.util.Comparator接口来自定义排序方式。

12.什么叫对象持久化（Object PERSistence），为什么要进行对象持久化？

持久化的对象，是已经存储到数据库或保存到本地硬盘中的对象，我们称之为持久化对象。为了保存在内存中的

各种对象的状态（也就是实例变量，不是方法），并且可以把保存的对象状态再读出来。虽然你可以用你自己的各种各样的方法来保存 object states，但是 Java 给你提供一种应该比你自己的保存对象状态的机制，那就是序列化。

简单说就是对象序列化是将对象状态转换为可保持或传输的格式的过程。

什么情况下需要序列化：

- a) 当你想把的内存中的对象状态保存到一个文件中或者数据库中时候；
- b) 当你想用套接字在网络上传送对象的时候；
- c) 当你想通过 RMI 传输对象的时候；

对象要实现序列化，是非常简单的，只需要实现 Serializable 接口就可以了。

```
public class Test implements Serializable
```

13.JavaScript 有哪些优缺点？

(1).javascript 的优点：

javascript 减少网络传输。

在 javascript 这样的用户端脚本语言出现之前，传统的数据提交和验证工作均由用户端浏览器通过网络传输到服

务器开发上进行。如果数据量很大，这对于网络和服务器开发的资源来说实在是一种无形的浪费。而使用 javascript 就可以在客户端进行数据验证。

javascript 方便操纵 html 对象。

javascript 可以方便地操纵各种页面中的对象，用户可以使用 javascript 来控制页面中各个元素的外观、状态甚至

运行方式，javascript 可以根据用户的需要“定制”浏览器，从而使网页更加友好。

avascript 支持分布式应用运算。

javascript 可以使多种任务仅在用户端就可以完成，而不需要网络和服务器开发的参与，从而支持分布式应用的运算和处理。

（2）javascript 的局限性：

各浏览器厂商对 javascript 支持程度不同。

目前在互联网上有很多浏览器，如 firefox、internet explorer、opera 等，但每种浏览器支持 javascript 的程度

是不一样的，不同的浏览器在浏览一个带有 javascript 脚本的主页时，由于对 javascript 的支持稍有不同，其效果会有一定的差距，有时甚至会显示不出来。

“web 安全性”对 javascript 一些功能牺牲。

当把 javascript 的一个设计目标设定为“web 安全性”时，就需要牺牲 javascript 的一些功能。因此，纯粹的

javascript 将不能打开、读写和保存用户计算机上的文件。其有权访问的唯一信息就是该 javascript 所嵌入开发的那个 web 主页中的信息，简言之，javascript 将只存在于它自己的小小世界—web 主页里

14.Jsp 有什么特点？

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。

JSP 技术是用 JAVA 语言作为脚本语言的，JSP 网页为整个服务器端的 JAVA 库单元提供了一个接口来服务于 HTTP 的应用程序。

在传统的网页 HTML 文件(.htm,.html)中加入 Java 程序片段(Scriptlet)和 JSP 标记(tag)，就构成了 JSP 网页

(*jsp)。Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定向网页以及发送 email 等等，这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的结果，对客户浏览器的要求最低，可以实现无 Plugin，无 ActiveX，无 Java Applet，甚至无 Frame

JSP 的优点：

- 1)对于用户界面的更新，其实就是由 Web Server 进行的，所以给人的感觉更新很快。
- 2)所有的应用都是基于服务器的，所以它们可以时刻保持最新版本。
- 3)客户端的接口不是很繁琐，对于各种应用易于部署、维护和修改

15.什么叫脏数据，什么叫脏读 (Dirty Read)

脏数据在临时更新（脏读）中产生。事务 A 更新了某个数据项 X，但是由于某种原因，事务 A 出现了问题，于是

要把 A 回滚。但是在回滚之前，另一个事务 B 读取了数据项 X 的值(A 更新后)，A 回滚了事务，数据项恢复了原值。事务 B 读取的就是数据项 X 的就是一个“临时”的值，就是脏数据。

脏读就是指当一个事务正在访问数据，并且对数据进行了修改，而这种修改还没有提交到数据库中，这时，另外一

个事务也访问这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据，那么另外一个事务读到的这个数据是脏数据，依据脏数据所做的操作可能是不正确的



持续更新中

回复“1024”获取Java架构师资源



微信搜一搜

Java研发军团

Java研发军团《Java面试手册》V1.0
公众号后台回复“面试手册”