# 常见面试题及答案

## 周振伟提出

### Exception和Error的区别？

Exception和Error都继承了Throwable类，在java中只有继承了Trowable类型的实例才可以被抛出throw或者捕获catch,它是异常处理机制的基本组成类型 。

Exception是程序运行中可以预料到的情况，可以被捕获并作出相应处理。

Error是指在正常情况下不可能出现的情况 ，无法预料或者被捕获。比如jvm虚拟机本身出错，不便也不需要被捕获，如OutOfMemoryError之类。

### final ,finally,finalize有什么不同？

他们三个的关系就像 雷锋和雷锋塔的关系

final：final可以修饰变量，方法 和类。被final修饰的变量只能被赋值一次，不可被修改。被final修饰的方法不可以被重写。被final修饰的类不可以被继承。

finally:finally和try catch一起使用，try finally 或者 try catch finally 用来处理异常，finally中的代码块中在程序不中止，进程 不被杀死的情况下，一定会被执行，一般用来释放资源。

finalize:Object类的一个方法 ，为了保证对象在被垃圾回收之前，完成特定的资源回收，java9已经不推荐使用这个方法。用cleaner代替。

### Integer和int的区别？

int 是我们常说的整形数字，是 Java 的 8 个原始数据类型（Primitive Types，boolean、byte 、short、char、int、float、double、long）之一。Java 语言虽然号称一切都是对象，但原始数据类型是例外。

Integer 是 int 对应的包装类，它有一个 int 类型的字段存储数据，并且提供了基本操作，比如数学运算、int 和字符串之间转换等。在 Java 5 中，引入了自动装箱和自动拆箱功能（boxing/unboxing），Java 可以根据上下文，自动进行转换，极大地简化了相关编程。

关于 Integer 的值缓存，这涉及 Java 5 中另一个改进。构建 Integer 对象的传统方式是直接调用构造器，直接 new 一个对象。但是根据实践，我们发现大部分数据操作都是集中在有限的、较小的数值范围，因而，在 Java 5 中新增了静态工厂方法 valueOf，在调用它的时候会利用一个缓存机制，带来了明显的性能改进

### 强引用、软引用、弱引用、幻象引用有什么区别？

强引用：

强引用就是指在程序代码中普遍存在的，类似 Object obj = new Objcet()这类的引用，只要强引用还存在，垃圾 收集器永远不会回收掉被引用的对象。

软引用：

软引用是用来描述一些还有用但 非必须的对象 ，对于软引用关联着的对象，在系统将要 发生内存溢出之前，将会把这些对象列进回收范围之中进行第二次回收。如果这次回收还没有足够的内存，才会抛出内存溢出的异常。

弱引用：

弱引用也是用来描述非必须对象的，但是他的强度比软引用更弱一些，被弱引用关联的对象只能生存到下一次垃圾收集发生之前。当垃圾收集器工作时，无论当前内存是否足够，都会回收掉只被弱引用关联的对象 。

虚引用：

虚引用也称为幽灵引用或者幻影引用，它是最弱的一种引用 关系。一个对象是否有虚引用的存在，完全不会对其生存时间构成影响。也无法通过虚引用来取的一个对象的示例。为一个对象设置虚引用的唯一目的就是能在这个对象被收集器回收时收到一个系统 通知。

## 于鑫提出

### **1.当数据表中A、B字段做了组合索引，那么单独使用A或单独使用B会有索引效果吗？（使用like查询如何有索引效果）**

答：看A、B两字段做组合索引的时候，谁在前面，谁在后面，如果A在前，那么单独使用A会有索引效果，单独使用B则没有，反之亦然。同理，使用like模糊查询时，如果只是使用前面%，那么有索引效果，如果使用双%号匹配，那么则无索引效果

### **2.常用排序算法**

选择排序

插入排序

冒泡排序

快速排序

堆排序

### **3.Java中Wait、Sleep和Yield方法的区别**

wait()是用于线程间通信的，而sleep()是用于短时间暂停当前线程。更加明显的一个区别在于，当一个线程调用wait()方法的时候，会释放它锁持有的对象的管程和锁，但是调用sleep()方法的时候，不会释放他所持有的管程。

回到yield()方法上来，与wait()和sleep()方法有一些区别，它仅仅释放线程所占有的CPU资源，从而让其他线程有机会运行，但是并不能保证某个特定的线程能够获得CPU资源。谁能获得CPU完全取决于调度器，在有些情况下调用yield方法的线程甚至会再次得到CPU资源。所以，依赖于yield方法是不可靠的，它只能尽力而为。

 wait只能在同步（synchronize）环境中被调用，而sleep不需要。

 进入wait状态的线程能够被notify和notifyAll线程唤醒，但是进入sleeping状态的线程不能被notify方法唤醒。

 wait通常有条件地执行，线程会一直处于wait状态，直到某个条件变为真。但是sleep仅仅让你的线程进入睡眠状态。

 wait方法在进入wait状态的时候会释放对象的锁，但是sleep方法不会。

 wait方法是针对一个被同步代码块加锁的对象，而sleep是针对一个线程。

### 4.现在有T1、T2、T3三个线程，你怎样保证T2在T1执行完后执行，T3在T2执行完后执行？

thread.Join把指定的线程加入到当前线程，可以将两个交替执行的线程合并为顺序执行的线程。 比如在线程B中调用了线程A的Join()方法，直到线程A执行完毕后，才会继续执行线程B。

## 迟佳梁提出

### 1.springMVC和spring boot 的区别

**Spring MVC的功能**

Spring MVC提供了一种轻度耦合的方式来开发web应用。

Spring MVC是Spring的一个模块，式一个web框架。通过Dispatcher Servlet, ModelAndView 和 View Resolver，开发web应用变得很容易。解决的问题领域是网站应用程序或者服务开发——URL路由、Session、模板引擎、静态Web资源等等。

**Spring Boot的功能**

Spring Boot实现了自动配置，降低了项目搭建的复杂度。

众所周知Spring框架需要进行大量的配置，Spring Boot引入自动配置的概念，让项目设置变得很容易。Spring Boot本身并不提供Spring框架的核心特性以及扩展功能，只是用于快速、敏捷地开发新一代基于Spring框架的应用程序。也就是说，它并不是用来替代Spring的解决方案，而是和Spring框架紧密结合用于提升Spring开发者体验的工具。同时它集成了大量常用的第三方库配置(例如Jackson, JDBC, Mongo, Redis, Mail等等)，Spring Boot应用中这些第三方库几乎可以零配置的开箱即用(out-of-the-box)，大部分的Spring Boot应用都只需要非常少量的配置代码，开发者能够更加专注于业务逻辑。

Spring Boot只是承载者，辅助你简化项目搭建过程的。如果承载的是WEB项目，使用Spring MVC作为MVC框架，那么工作流程和你上面描述的是完全一样的，因为这部分工作是Spring MVC做的而不是Spring Boot。

对使用者来说，换用Spring Boot以后，项目初始化方法变了，配置文件变了，另外就是不需要单独安装Tomcat这类容器服务器了，maven打出jar包直接跑起来就是个网站，但你最核心的业务逻辑实现与业务流程实现没有任何变化。

所以，用最简练的语言概括就是：

Spring 是一个“引擎”；

Spring MVC 是基于Spring的一个 MVC 框架 ；

Spring Boot 是基于Spring4的条件注册的一套快速开发整合包。

throw和throws的区别

## 李加薪提出

### 1.throws和throw的区别？

1.throws出现在方法函数头；而throw出现在函数体。

2.throws表示出现异常的一种可能性，并不一定会发生这些异常；throw则是抛出了异常，执行throw则一定抛出了某种异常对象。

3.两者都是消极处理异常的方式（这里的消极并不是说这种方式不好），只是抛出或者可能抛出异常，但是不会由函数去处理异常，真正的处理异常由函数的上层调用处理。

### **2.sendRedirect() 和forward()的区别？**

HttpServletResponse.sendRedirect与RequestDispatcher.forward方法都可以实现获取相应URL资源。

sendRedirect实现请求重定向，forward实现的是请求转发

在web服务器内部的处理机制也是不一样的。

1.跳转方式,forward方法只能重定向到同一个Web应用程序中的一个资源,sendRedirect方法可以让你重定向到任何URL。

2.forward重定向后，浏览器url地址不变，sendRedirect转发后，浏览器url地址变为目的url地址。

3.使用forward重定向的过程，是浏览器先向目的Servlet发送一次Request请求，然后在服务器端由Servlet再将请求发送到目的url，再由服务器端Servlet返回Response到浏览器端。浏览器和服务器一次请求响应。

使用sendRedirect转发的过程，浏览器先向目的Servlet发送一次请求，Servlet看到sendRedirect将目的url返回到浏览器，浏览器再去请求目的url,目的url再返回response到浏览器。浏览器和服务器两次请求响应。

4.forward方法的调用者与被调用者之间共享Request和Response，sendRedirect方法由于两次浏览器服务器请求，所以有两个Request和Response。