



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、Java中异常分为哪两种？

编译时异常 运行时异常

## 2、异常的处理机制有几种？

异常捕捉：try...catch...finally，异常抛出：throws。

## 3、如何自定义一个异常

继承一个异常类，通常是RuntimeException或者Exception

## 4、try catch finally，try里有return，finally还执行么？

执行，并且finally的执行早于try里面的return

结论：

- 1、不管有没有出现异常，finally块中代码都会执行；
- 2、当try和catch中有return时，finally仍然会执行；
- 3、finally是在return后面的表达式运算后执行的（此时并没有返回运算后的值，而是先把要返回的值保存起来，管finally中的代码怎么样，返回的值都不会改变，任然是之前保存的值），所以函数返回值是在finally执行前确定的；
- 4、finally中最好不要包含return，否则程序会提前退出，返回值不是try或catch中保存的返回值。

## 5、Exception与Error包结构

Java可抛出(Throwable)的结构分为三种类型：被检查的异常(CheckedException)，运行时异常(RuntimeException)，错误(Error)。

## 1、运行时异常

定义:RuntimeException及其子类都被称为运行时异常。

特点:Java编译器不会检查它。也就是说,当程序中可能出现这类异常时,倘若既"没有通过throws声明抛出它",也"没有用try-catch语句捕获它",还是会编译通过。例如,除数为零时产生的ArithmeticException异常,数组越界时产生的IndexOutOfBoundsException异常,fail-fast机制产生的ConcurrentModificationException异常(java.util包下面的所有的集合类都是快速失败的,"快速失败"也就是fail-fast,它是Java集合的一种错误检测机制。当多个线程对集合进行结构上的改变的操作时,有可能会产生fail-fast机制。记住是有可能,而不是一定。例如:假设存在两个线程(线程1、线程2),线程1通过Iterator在遍历集合A中的元素,在某个时候线程2修改了集合A的结构(是结构上面的修改,而不是简单的修改集合元素的内容),那么这个时候程序就会抛出ConcurrentModificationException异常,从而产生fail-fast机制,这个错叫并发修改异常。Fail-safe,java.util.concurrent包下面的所有的类都是安全失败的,在遍历过程中,如果已经遍历的数组上的内容变化了,迭代器不会抛出ConcurrentModificationException异常。如果未遍历的数组上的内容发生了变化,则有可能反映到迭代过程中。这就是ConcurrentHashMap迭代器弱一致的表现。ConcurrentHashMap的弱一致性主要是为了提升效率,是一致性与效率之间的一种权衡。要成为强一致性,就得到处使用锁,甚至是全局锁,这就与Hashtable和同步的HashMap一样了。)等,都属于运行时异常。

常见的五种运行时异常:

ClassCastException (类转换异常)

IndexOutOfBoundsException (数组越界)

NullPointerException (空指针异常)

ArrayStoreException (数据存储异常,操作数组是类型不一致)

BufferOverflowException

## 2、被检查异常

定义:Exception类本身,以及Exception的子类中除了"运行时异常"之外的其它子类都属于被检查异常。特点:Java编译器会检查它。此类异常,要么通过throws进行声明抛出,要么通过try-catch进行捕获处理,否则不能通过编译。例如,CloneNotSupportedException就属于被检查异常。当通过clone()接口去克隆一个对象,而该对象对应的类没有实现Cloneable接口,就会抛出CloneNotSupportedException异常。被检查异常通常都是可以恢复的。

如:

IOException

FileNotFoundException

SQLException

被检查的异常适用于那些不是因程序引起的错误情况,比如:读取文件时文件不存在引发的FileNotFoundException。然而,不被检查的异常通常都是由于糟糕的编程引起的,比如:在对象引用时没有确保对象非空而引起的NullPointerException。

## 3、错误

定义:Error类及其子类。

特点：和运行时异常一样，编译器也不会对错误进行检查。当资源不足、约束失败、或是其它程序无法继续运行的条件发生时，就产生错误。程序本身无法修复这些错误的。例如，VirtualMachineError就属于错误。出现这种错误会导致程序终止运行。OutOfMemoryError、ThreadDeath。

Java虚拟机规范规定JVM的内存分为了好几块，比如堆，栈，程序计数器，方法区等

## 6、Throw与throws区别

### 位置不同

1. throws 用在函数上，后面跟的是异常类，可以跟多个；而 throw 用在函数内，后面跟的是异常对象。

### 功能不同：

1. throws 用来声明异常，让调用者只知道该功能可能出现的问题，可以给出预先的处理方式；throw 抛出具体的问题对象，执行到 throw，功能就已经结束了，跳转到调用者，并将具体的问题对象抛给调用者。也就是说 throw 语句独立存在时，下面不要定义其他语句，因为执行不到。

1. 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。

1. 两者都是消极处理异常的方式，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。

## 7、Error与Exception区别？

Error和Exception都是java错误处理机制的一部分，都继承了Throwable类。

Exception表示的异常，异常可以通过程序来捕捉，或者优化程序来避免。

Error表示的是系统错误，不能通过程序来进行错误处理。

## 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