

什么是final及final的应用

- final：是最终的意思，可修饰类，变量，方法。
- 特点：
 - 修饰类：类不可以被继承
 - 修饰方法：方法不被重写
 - 修饰变量：该变量变成常量，且不可以被修改
- 当final修饰局部变量的时候：
 - 该局部变量在方法内部，那么他的值不能被改变。
 - 该变量在方法声明上，那么当他修饰基本类型时是值不被修改，如果是引用类型则是地址不被修改。

异常

- 程序在正常执行的情况下发生了不正当的事件，阻止了程序的正常执行。
- 异常的关键字：
 - try、catch、finally、throw、throws。

异常的应用

- 正常,不执行catch块，catch块之后剩余的代码

```
try { // 代码段(此处不会产生异常)
    } catch (异常类型 ex) {
        // 对异常进行处理的代码段
    }    //代码段
```

- 代码段产生异常时，进行异常类型匹配，如果异常类型相同，则执行catch块内容。

```
try { // 代码段(产生异常)
    } catch (异常类型 ex) {
        // 对异常进行处理的代码段
    }    //代码段
```

- 代码段产生异常，异常类型匹配失败，则终止程序运行，即异常后的所有代码都不能够正常执行。

```
try { 代码段1;
        代码段2(产生异常)
    代码段3
    } catch (异常类型 ex) {
        // 对异常进行处理的代码段4
    }//代码段5
```

异常的体系结构

- Java语言要求java程序中（无论是谁写的代码）所有抛出（throw）的异常都必须是从Throwable派生而来。java程序员一般是不需要再重新定义自己的异常对象。而且即便是需要扩展自定义的异常对象，也往往会从Exception派生而来。所以，对于java程序员而言，它一般只需要在它的顶级函数中catch(Exception ex)就可以捕获出所有的异常对象。所有异常对象的根基类是 Throwable，Throwable从Object直接继承而来（这是java系统所强制要求的），并且它实现了 Serializable接口（这为所有的异常对象都能够轻松跨越Java组件系统做好了最充分的物质准备）
- Throwable是java.lang包中一个专门用来处理异常的类。它有两个子类，即Error 和 Exception，它们分别用来处理两组异常。
 - Error用来处理程序运行环境方面的异常
 - Exception是Throwable的一个主要子类。Exception下面还有子类，其中一部分子类分别对应于Java程序运行时常常遇到的各种异常的处理，其中包括隐式异常(比如，程序中除数为0引起的错误、数组下标越界错误等，这类异常也称为运行时异常，因为它们虽然是由程序本身引起的异常，但不是程序主动抛出的，而是在程序运行中产生的)
 - Exception 子类下面的另一部分子类对应于Java程序中的非运行时异常的处理,这些异常也称为显式异常,它们都是在程序中用语句抛出、并且也是用语句进行捕获的，比如，文件没找到引起的异常、类没找到引起的异常等。
- 从Throwable直接派生出的异常类有Exception和Error
 - **Exception:** 它一般代表了真正实际意义上的异常对象的根基类,是应用程序能够catch到的，种异常是设计中早已考虑到很常见的错误，并且可以进行异常错误恢复处理，以及如何处理等。
 - **Error:** 表示Java系统中出现了一个非常严重的异常错误，并且这个错误可能是应用程序所不能恢复的，也就是你再试一次依然会出错，比如 OutOfMemoryError，试多少次很大概率出错的。
 - RuntimeException: 表示这种异常是否出现经常不可预料，出现了的话，就中止处理
- RuntimeException 更接近我们日常生活中的“意外”。
- Exception 这个更接近我们日常生活中所说的“常见错误”。
- Error 更像是“见鬼，看来周末国庆不加班是搞不定的”。

Java异常Throwable、Error、Exception、RuntimeException的区别

- Throwable类是 Java 语言中所有错误或异常的超类。它的两个子类是Error和Exception；
- Error是Throwable 的子类，用于指示合理的应用程序不应该试图捕获的严重问题(如内存溢出、虚拟机错误、栈溢出等。这类错误一般与硬件有关，与程序本身无关，通常由系统进行处理，程序本身无法捕获和处理。)
 - OutOfMemoryError内存溢出
 - java.lang.StackOverflowError 堆栈溢出错误(当一个应用递归调用的层次太深而导致堆栈溢出时抛出该错误。)
- Exception类及其子类是 Throwable 的一种形式，合理的应用程序想要捕获的条件(++有些异常在编写程序时无法预料的，如中断异常、非法存取异常等。为了保证程序的健壮性，Java要求必须对这些可能出现的异常进行捕获，并对其进行处理++)
- RuntimeException类是Exception类的子类(++可能在 Java 虚拟机正常运行期间抛出的异常的超类++)

- IOException类是Exception类的子类。