Java异常处理

1. 基本概念：

当程序出现无法控制的外部环境问题（如用户提供的文件不存在，文件内容损坏，网络不可用……）时,JAVA就会调用异常对象来描述

1. Java中异常的分类
2. 检查性异常： java.lang.Exception即在编译时就会发生异常，编译不过去
3. 运行时异常 java.lang.RuntimeExcepiton在运行时才会报错，编译时不会出问题，运行时异常意味着程序存在bug，如数组下标越界，0被除等，Java编译器强制要求处理这类异常。
4. 错误 java.lang.Error

错误一般很少见，也很难通过程序解决，它可能源于程序的bug，但多见于环境问题如内存耗尽，错误在程序中无须处理，一般由环境来处理。

注：Throwable是所有异常的父类，可查看jdk文档

1. Java处理异常的方式
2. 在发生异常的地方直接处理掉
3. 将异常抛给调用者，让调用者处理

直接处理的方法：

**try** {

FileReader fr = **new** FileReader("d:\\aa.txt");

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally**{

//关闭资源

}

抛出异常的方法：

public void aa() throws Exception{

FileReader fr = new FileReader("d:\\aa.txt");

}

Finalize() 垃圾回收方法

Finally 异常处理

Final 修饰符

Java自动装箱与拆箱

Java在5.0版本之后就提供了基本数据类型的自动装箱、拆箱

1.什么是自动装箱与拆箱

一般我们要创建一个类的对象实例的时候，我们会这样：

 Class a = new Class(parameter);

 当我们创建一个Integer对象时，却可以这样：

 Integer i = 100; (注意：不是 int i = 100; )

实际上，执行上面那句代码的时候，系统为我们执行了：

Integer i = Integer.valueOf(100);

此即基本数据类型的自动装箱功能

2、基本数据类型与对象的差别

基本数据类型不是对象，也就是使用int、double、boolean等定义的变量、常量。

基本数据类型没有可调用的方法。

eg：  int t = 1；     t.  后面是没有方法滴。

 Integer t = 1； t.  后面就有很多方法可让你调用了。

### 什么时候自动拆箱

　　自动拆箱(unboxing)，也就是将对象中的基本数据从对象中自动取出。如下可实现自动拆箱：

1 Integer i = 10; //装箱   
2  int t = i; //拆箱，实际上执行了 int t = i.intValue();

在进行运算时，也可以进行拆箱。

1 Integer i = 10;   
2 System.out.println(i++);

1. 基本数据类型的包装类型是什么

byte ------->Byte

short-->Short

int -->Integer

long--> Long

float--> Float

double-->Double

char -->Character

boolean-->Boolean

总结：

1. 装箱：将基本数据类型转换成其对应的包装(引用)类型
2. 拆箱：将包装类型转化为基本数据类型
3. 基本数据类型可以通过装箱转为引用数据类型
4. 每种基本数据类型都有与之对应的包装类型