**异常 Throwable**

**1.异常的概述**

**异常就是程序出现了不正常的情况**

**异常体系:**

**Throwable:Error和Exception**

**Exception:RuntimeException和非RuntimeException**

**Error: 严重问题,不需要处理.**

**Exception: 称为异常类,它表示程序本身可以处理的问题.**

**RuntimeException: 在编译期间不会报错,也就是编译通过,在编译期间是不检查的,出现问题后,才需要回来修改代码.**

**非RuntimeException: 编译不通过,所以也不可能运行.**

**2.JVM的默认处理方案 RuntimeException**

**如果程序出现了问题,我们没有预见也没有去处理的话,JVM会有默认处理方案.**

**方案:把异常的名称,异常的原因和异常在代码中出现的行号返回并打印在控制台**

**停止程序运行,不会在执行接下来的代码.**

**3.异常处理**

**如果程序出现了问题,我们需要自己来处理,有两种方案:**

**try...catch... 抓取**

**throws 抛出**

**try...catch...**

**格式:**

**try{**

**可能出现异常的代码;**

**}catch(异常类名 变量名){**

**异常处理的代码;**

**}**

**执行流程:**

**1.程序从try里面的代码开始执行;**

**2.出现异常,会自动生成一个异常类对象,该异常对象将被提交给Java运行时系统;**

**3.当Java运行时系统接收到异常处理对象,会到catch中去找匹配的异常类,找到后进行异常的处理;**

**4.执行完毕后,还可以继续执行程序**

**Throws**

**try...catch...和throws是两种处理异常的方法,**

**格式:**

**Throws 异常类名 这个格式是跟在方法的形参括号后面的**

**编译时异常是非RuntimeException, 两种处理方案,try或者throws都可以.但如果使用throws,将来谁调用,还是需要处理这个异常.**

**运行时异常是RuntimeException, 可以暂时不处理,等运行报错了再回来改代码**

**理解总结一下:**

**遇到异常JVM会终止程序运行,为了一个异常不会导致整个程序甚至整个项目停止,需要抛出(throws 程序不会停,也不会出现这个异常了),或者抓取(****try...catch... 会把异常通过Exception的方法把异常元婴,出现位置,打印在对应程序位置的控制台)异常,使得程序可以继续运行.因为考虑到一段不确定是否会存在异常的代码 try...catch... ,这是代码的规范书写格式.在格式中异常直接用Exception,当然,确定不会出问题的代码,比如变量定义等自己可以确定不会出现异常的代码则不需要try...catch... ,而编译报错要么解决,要么抛出异常,也是可以的.**

**4.** **RuntimeException extends Exception extends Throwable**

**5.Throwable的成员方法**

**public class Throwable extends Object implements Serializable{}**

**Throwable类是Java语言中所有错误(Error)和异常(Exception)的超类**

**Throwable的成员方法:**

**e.getMessage() 返回此throwable的详细消息字符串**

**e.toString() 返回此可抛出的简短描述**

**e.printStackTrace​() 将此throwable和其追溯打印到标准错误流。**

**6.编译时异常和运行时异常的区别**

**Java中的异常被分为两大类,编译时异常和运行时异常,也被称为受检异常和非受检异常,所有的 RuntimeException 类及其子类的实例被称为运行时异常,其他的异常都是编译时异常.**

**编译时异常:必须显示处理,否则程序就会发生错误,我发通过编译;**

**运行时异常:无需监视出来,也可以和编译时异常一样处理.**

**7.自定义异常**

**就是自定义一个异常类,继承Exception 类中要包含无参和带参构造方法,带参构造传入的是异常.**

**throws和throw的区别**

**throws:用在方法声明后面,跟的是异常类名**

**表示抛出异常,由该方法的调用者来处理**

**表示出现异常的一种可能性,并不一定会发生这些异常**

**throw:用在方法体内,跟的是异常对象名**

**表示抛出异常,由方法体内的语句处理**

**执行throw一定抛出了某种异常**