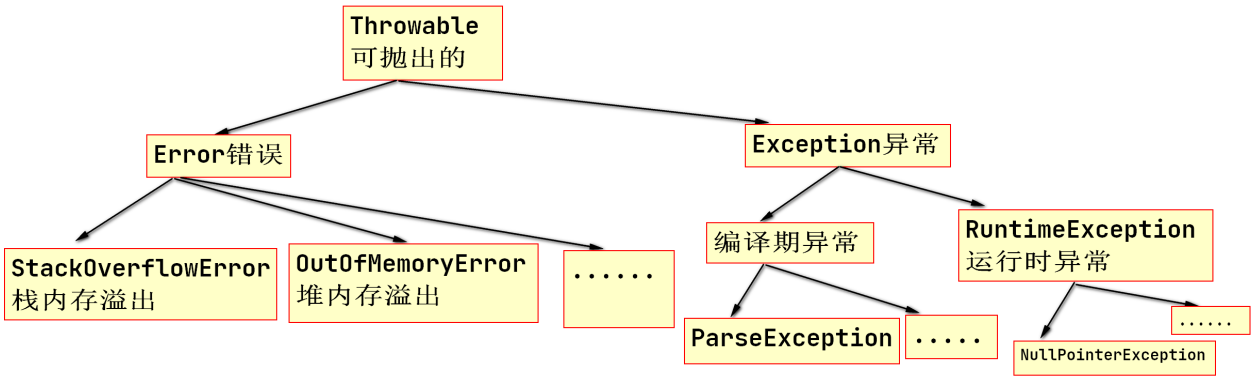
1、什么是异常

我们的程序在运行过程中产生的一些问题，这些问题有的是我们自己编码产生的，有的问题是用户的不当操作引起的，也有可能是外部物理设备引起的。

结构：



2、异常的继承结构

·Throwable：可抛出的

两个子类

·Error：错误：

一般是由外部物理设备引起的故障，无法通过代码来进行解决的。

一般不需要进行处理。

1、StackOverflowError 栈内存溢出

2、OutofMemoryError 堆内存溢出

·Exception:异常

一般是程序编写的引起的问题或者用户不正当操作引起的，是需要通过编码来进行解决的

----->编译时异常（检查异常）

在程序编译期间由编译器检测出来的，在编译期间必须要处理，否则编译就不通过，如：ParseException

----->运行时异常（非检查异常）

运行时在编译期间编译器检测不到，但是在运行期间又可能会报的异常，在编译器期间可以处理，也可以不处理，所有的运行时异常都继承自RuntimeException

如：NullPointerException 空指针异常

ArrayIndexOutOfBoundsException 数组下标越界

ArithmeticException 数字运算异常

ClassCastException 类型转换异常

3、编译期异常的处理

1、抛出（throw）

在方法上面通过throws将异常进行向上抛出。

异常将被抛给方法调用者，谁来调用这个方法异常就抛在给谁。

如果方法调用者也选择将异常进行抛出，呢么最终异常将会被抛给虚拟机，

jvm虚拟机就会打印出异常的堆栈信息。

如果在方法上想要抛出多个异常，这多个异常之间可以用逗号隔开。

或者直接抛出这多个异常的父类异常，异常的最终父类就是Exception。

注意：如果在方法中某一行代码出现异常了，呢么这个方法就结束了。

跟在这一行后面的代码就不再执行了。

2、捕获

我们可以再方法中通过try-catch代码块来对异常进行捕获并处理，

呢么在方法上面就不需要再对异常进行抛出了，

方法调用者也就不需要再去处理异常了

语法：

```
try{
        //有编译期异常的代码，e代表捕获到的异常
}catch(XXException e){
        //捕获异常，并对异常进行处理
}finally{
        //最终一定会执行的代码块
}
```

1>try中代码有异常就会进入catch中来执行

2>finally中代码一定会执行，即使try或者catch中有return，

也会去执行finally中代码，再回去执行return、

3>catch可以有多个，用于捕获多个不同类型的异常，

这些异常之间如果有父子关系，呢么子类型异常必须写在符类型异常上面，

也可以捕获这多个异常的父类型异常

4>如果有一定要执行的代码，可以使用try~finally代码块结构

四、自定义异常

当官方提供的异常不满足我们实际的业务需求时，可以进行自定义异常

自定义异常的步骤：

1>自己编写一个异常类，

自定义运行时异常，需要继承自RuntimeException

自定义编译期异常，需要继承Eeception

2>在异常类中调用父类中的构造器，并传入异常信息字符串

3>在方法内部可以创建自定义异常对象，并通过throw关键字进行抛出

异常面试题：

1>Exception和Error之间有什么区别

2>运行时异常和编译期异常有什么区别

3>如果try或catch中有return，呢么finally中还会不会执行

4>throw和throws有什么区别

5>finally,final,fianllize有什么区别

6>说明5个常见的运行时异常