

一、异常的概念

1、简介

Java语言中，将程序执行中发生的不正常情况称为“异常”。（开发过程中的语法错误和逻辑错误不是异常）

2、异常事件分类

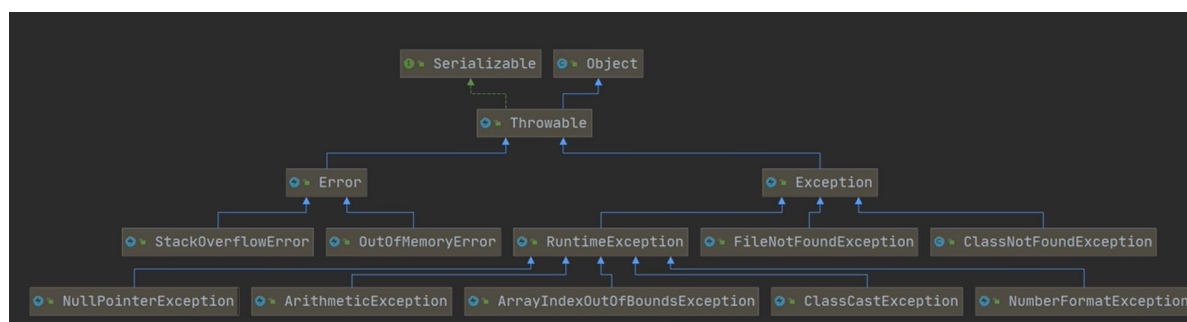
2.1、Error(错误)：Java虚拟机无法解决的严重问题。如：JVM系统内部错误、资源耗尽等

2.2、Exception：其他因编程错误或偶然的外在因素导致的一般性问题，可以使用针对性的代码进行处理。如：空指针访问、试图读取不存在的文件、网络连接终端等等。Exception分为两大类：

- 运行时异常：程序运行时发生的异常
- 编译时异常：编程时，编译器检查出的异常

二、***异常体系图

1、异常体系图



2、异常体系图总结

2.1、异常分为两大类，运行时异常和编译时异常

2.2、运行时异常，编译器不要求强制处置的异常。一般是指编程时的逻辑错误，是程序员应该避免其出现的异常。`java.lang.RuntimeException`类及它的子类都是运行时异常

2.3、对于运行时异常，可以不做处理，因为这类异常很普遍，若全处理可能对程序的可读性和运行效率产生影响

2.4、编译时异常，是编译器要求必须处置的异常

三、常见异常

1、运行异常-NullPointerException

空指针异常：当应用程序使用的对象为`null`时，抛出该异常。

2、运行异常 -ArithmeticException

数学运算异常：当出现异常的运算条件时，抛出此异常。

3、运行异常 -ArrayIndexOutOfBoundsException

数组下标越界异常：用非法索引访问数组时抛出的异常。如果索引为负或大于等于数组的大小，则该索引为非法索引。

4、运行异常 -ClassCastException

类型转换异常：当试图将对象强制转换为不是实例的子类时，抛出该异常。

5、运行异常 -NumberFormatException

数字格式不正确异常：当应用程序试图将字符串转换成一种数值类型，但该字符串不能转换为适当格式时，抛出该异常。

6、编译异常-常见汇总

SQLException	操作数据库时，查询表可能发生异常
IOException	操作文件时，发生的异常
FileNotFoundException	当操作一个不存在的文件时，发生异常
ClassNotFoundException	加载类，而该类不存在时，异常
EOFException	操作文件，到文件末尾，发生异常
IllegalArgumentException	参数异常

四、***异常处理概念

1、简介

异常处理就是当异常发生时，对异常处理的方式。

2、异常处理方式

- 2.1、try-catch-finally: 程序员在代码中捕获发生的异常，自行处理
- 2.2、throws: 将发生的异常抛出，交给调用者（方法）来处理，最顶级的处理者就是JVM

五、***异常处理分类

1、try-catch-finally 处理机制

```
try {  
    // 代码可能有异常  
} catch (Exception e) {  
    // 捕获到异常  
    // 1、当异常发生时  
    // 2、系统将异常封装成Exception对象e，传递给catch  
    // 3、得到异常对象后，程序员自己处理  
    // 4、如果没有发生异常，catch代码块不执行  
} finally {  
    // 1、不管try代码块是否有异常发生，始终要执行finally  
    // 2、所以通常将释放资源的代码，放在finally  
}
```

2、throws 处理机制

- 2.1、throws方法会将异常从底层依次向上传递，直至JVM
- 2.2、如果都没有处理，JVM则会输出异常，并退出程序
- 2.3、如果程序员没有显式处理异常，默认使用throws

3、try-catch 细节

- 3.1、如果发生异常，则异常发生位置后面的代码不会执行，直接进入catch块
- 3.2、如果异常没有发生，则顺序执行try的代码块，不会进入catch
- 3.3、如果希望不管是否发生异常，都执行某段代码（如关闭连接、释放资源等），则使用 finally
{ }
- 3.4、可以有多个catch语句，捕获不同的异常（进行不同的业务处理），要求父类异常在后，子类异常在前
- 3.5、可以进行try-finally配合使用，相当于没有捕获异常，因此程序会直接崩掉。应用场景：执行一段代码，不管是否发生异常，都必须执行某个业务逻辑

4、throws 细节

- 4.1、对于编译异常，程序中必须处理，try-catch 或者 throws都可以
- 4.2、对于运行时异常，程序中如果没有处理，默认就是throws的方式处理
- 4.3、子类重写父类的方法时，对抛出异常的规定：子类重写的方法，所抛出的异常类型要么和父类抛出的异常一致，要么为父类抛出的异常的子类型
- 4.4、在throws过程中，如果有方法try-catch，就相当于处理异常，就可以不必throws

六、自定义异常

1、简介

当程序中出现了某种“错误”，但该错误信息并没有在Throwable子类中描述错误，这个时候可以自己设计异常类，用于描述该错误信息。

2、自定义异常步骤

- 2.1、定义类：自定义异常类名，继承Exception（编译异常）或RuntimeException（运行异常）
- 2.2、构造器：创建构造器，形参传入message
- 2.3、调用：调用时使用 `throw new` [自定义异常(实参)]

七、throw和throws对比

	意义	位置	后面跟的内容
throws	异常处理的一种方式	方法声明处	异常类型
throw	手动生成异常对象的关键字	方法体中	异常对象