

异常：

● 基本概念

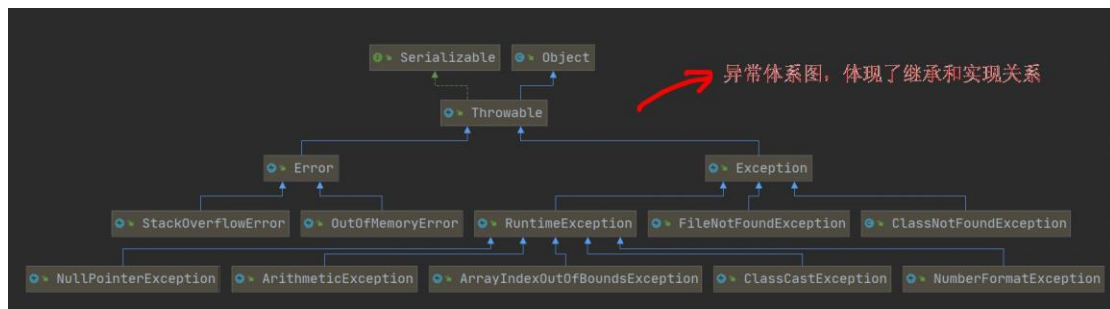
Java语言中，将程序执行中发生的不正常情况称为“异常”。(开发过程中的语法错误和逻辑错误不是异常)

● 执行过程中所发生的异常事件可分为两大类

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、资源耗尽等严重情况。比如: `StackOverflowError`[栈溢出]和`OOM(out of memory)`, Error 是严重错误, 程序会崩溃。

2) Exception: 其它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。例如空指针访问, 试图读取不存在的文件, 网络连接中断等等, Exception 分为两大类: **运行时异常**[程序运行时, 发生的异常]和**编译时异常**[编程时, 编译器检查出的异常]。

异常体系：



异常体系图小结

1. 异常分为两大类，运行时异常和编译时异常。
2. 运行时异常，编译器检查不出来。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
3. 对于运行时异常，可以不作处理，因为这类异常很普遍，若全处理可能会对程序的可读性和运行效率产生影响
4. 编译时异常，是编译器要求必须处置的异常。

运行时异常：

1) `NullPointerException` 空指针异常

当应用程序试图在需要对象的地方使用 `null` 时，抛出该异常

2) `ArithmeticException` 数学运算异常

当出现异常的运算条件时，抛出此异常。

3) `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大小，则该索引为非法索引。

4) 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) 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。

编译异常：

编译期间就必须处理的异常，否则代码不能通过编译。

常见类型：

```
✓ 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
```

异常处理

方法一：try-catch-finally

程序员在代码中捕获发生的异常，自行处理

```
1) Java提供try和catch块来处理异常。try块用于包含可能出错的代码。catch块用于处理try块中发生的异常。可以根据需要在程序中有多个try...catch块。

2) 基本语法

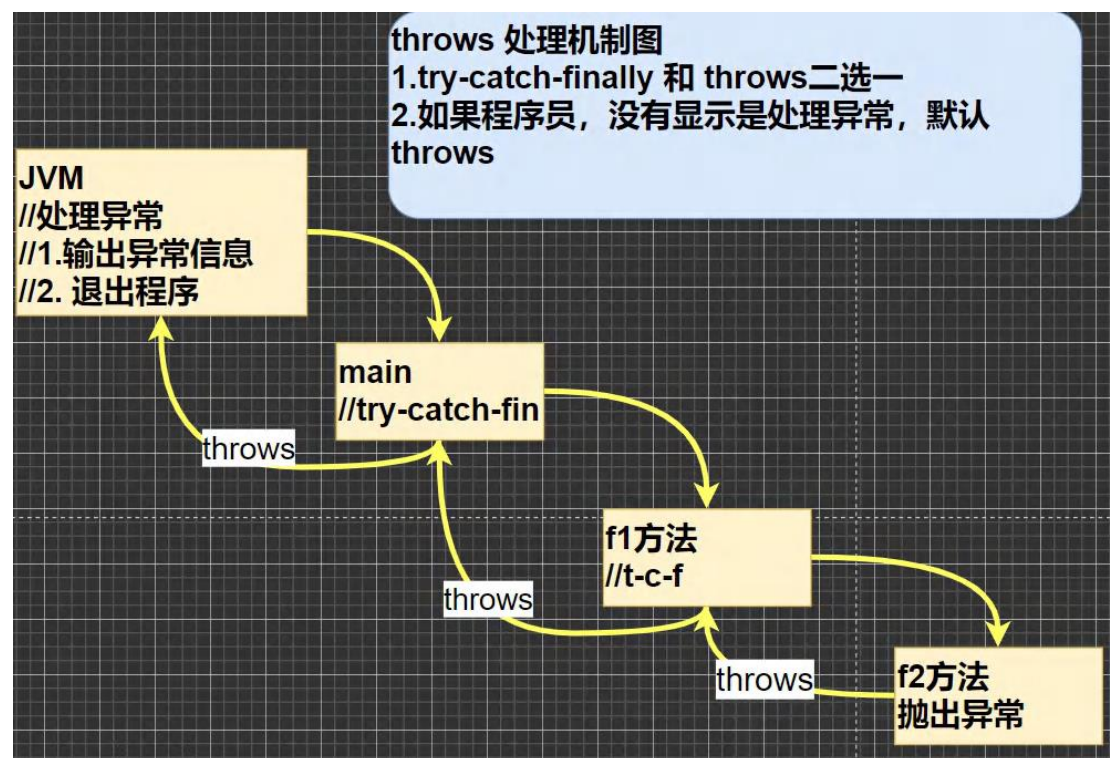
try {
    //可疑代码
    //将异常生成对应的异常对象，传递给catch块
} catch (异常){
    //对异常的处理
}
//如果没有finally，语法是可以通过
```

1. 如果异常发生了，则异常发生后面的代码不会执行，直接进入到了 catch 块
2. 如果异常没有发生，则顺序执行 try 的代码块，不会进入到 catch
3. 如果希望不管是否发生异常，都执行某段代码(比如关闭连接，释放资源等)则使用如下代码- finally
4. 如果 try 代码块有可能有多个异常可以使用多个 catch 分别捕获不同的异常，相应处理要求子类异常写在前面，父类异常写在后面。
5. 可以进行 try-finally 配合使用，这种用法相当于没有捕获异常，因此程序会直接崩

掉/退出。应用场景，就是执行一段代码，不管是否发生异常，都必须执行某个业务逻辑。

方法二：throws

将发生的异常抛出，交给调用者即方法处理，最顶级的处理者就是 JVM。



- 1) 如果一个方法(中的语句执行时)可能生成某种异常，但是并不能确定如何处理这种异常，则此方法应显示地声明抛出异常，表明该方法将不对这些异常进行处理，而由该方法的调用者负责处理。
- 2) 在方法声明中用throws语句可以声明抛出异常的列表，throws后面的异常类型可以是方法中产生的异常类型，也可以是它的父类。

1. 对于编译异常，程序中必须处理，比如 try-catch 或者 throws
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
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类型的子类型。
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。

自定义异常：

当程序中出现了某些“错误”，但是该错误信息没有在 Throwable 子类中描述处理，这个时候可以自己设计异常类，用于描述该错误信息。

- 1) 定义类: 自定义异常类名(程序员自己写) 继承Exception或RuntimeException
- 2) 如果继承Exception, 属于编译异常
- 3) 如果继承RuntimeException, 属于运行异常(一般来说, 继承RuntimeException)

一般情况下,我们自定义异常是继承 RuntimeException,即把自定义异常做成 运行时异常,好处时,我们可以使用默认的处理机制即比较方便

Throw 与 throws

	意义	位置	后面跟的东西
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