异常：指的就是程序在 **运行时** 出现错误通知调用者的一种机制.

**运行时异常（非受查异常）**：派生于 Error 类或 RuntimeException 类的所有异常称为

**编译时异常（受查异常）**：除非受查异常之外的其他异常，需要进行显示处理，否则编译不通过

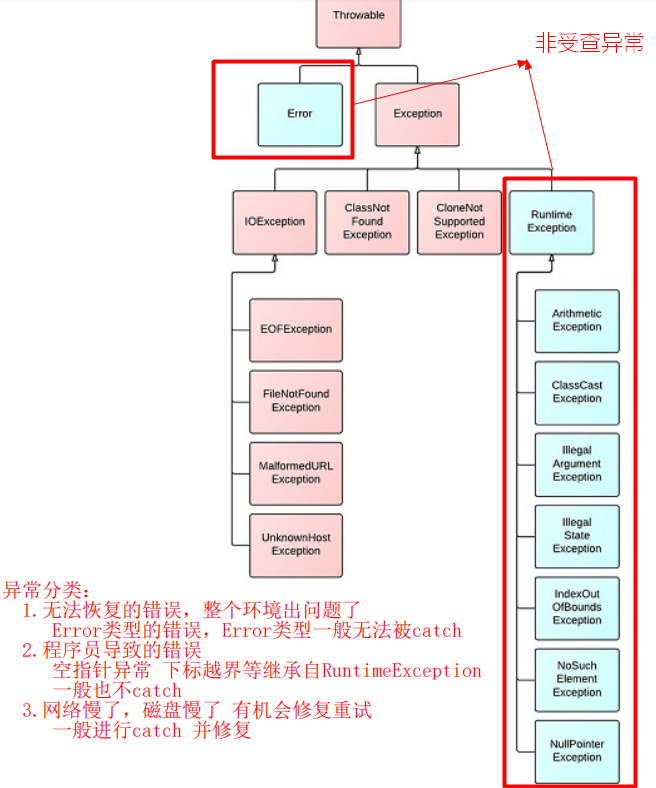
**两者区别：**

非受查异常：java编译器不要求你一定要把它捕获或者一定要继续抛出

受查异常：要求你必须要在方法里面或者捕获或者继续抛出。

非受查：NullPointerException,ClassCastException,ArrayIndexsOutOfBoundsException,ArithmeticException(算术异常，除0溢出)

受查：Exception,FileNotFoundException,IOException,SQLException.



**异常捕获：**try-catch-finally

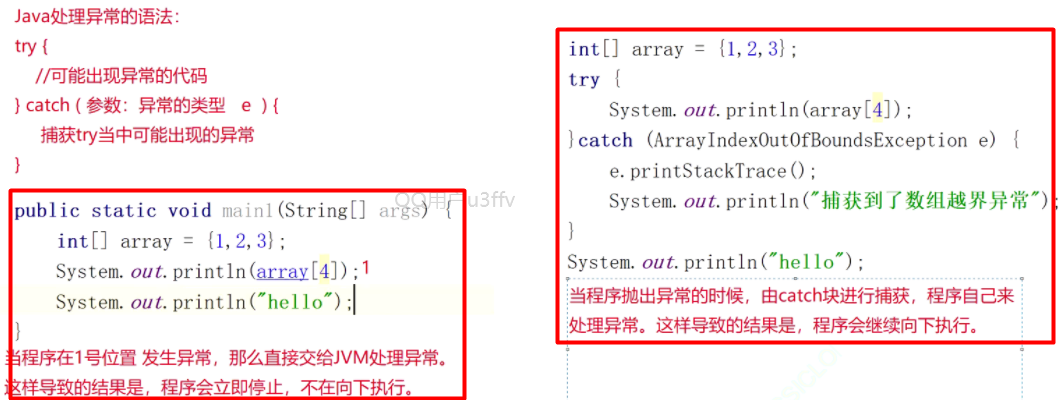
try 放的是可能出现异常的代码.

catch 放的是出现异常后的处理行为.(只有异常类型匹配才能处理）

finally 无论有没有异常都会执行

其中 catch 和 fifinally 都可以根据情况选择加或者不加.

catch可以多个，也可以用一个处理所有异常



**异常处理步骤：**

先执行 try 中的代码

try 中的代码出现异常, 就会结束 try 中的代码, 与catch 中的异常类型是否匹配.

找到匹配的异常类型, 就会执行 catch 中的代码

没有找到匹配的异常类型, 就会将异常向上传递到上层调用者.

无论是否找到匹配的异常类型, finally 中的代码都会被执行到

如果上层调用者也没有处理的了异常, 就继续向上传递.一直到 main 方法也没有合适的代码处理异常, 就会交给 JVM 来进行处理, 此时程序就会异常终止

注意：finally 中也存在 return 语句, 那么就会执行 finally 中的 return, 从而不会执行到 try 中原有的 return.

**抛出异常:throw**

**Error与Exception区别：**

Error 指的是 Java 运行时内部错误和资源耗尽错误，应用程序不抛出此类异常. 告知用户并使程序终止。

Exception 是我们程序猿所使用的异常类的父类.

**自定义异常**

需要继承一个父类异常（Exception 或者 RuntimeException）

需要构造方法

继承自 Exception 的异常默认是受查异常

继承自 RuntimeException 的异常默认是非受查异常.（只有在运行时才会抛出）

所以最好处理异常使用try catch包裹