

# Lab 09 Assignment

学号：71216002

姓名：金俊河

提交要求 ( 以下内容在最终提交的作业中可以删除，仅作参考 )

目录格式

```
-- 班级-学号-姓名-Lab09.zip
  |-- answer.pdf
  |-- solution
    |-- Question1
      |-- 题1的代码文件(如果需要)

    |-- Question2
      |-- 题2的代码文件(如果需要)
    |-- ...
```

## Question1

非检查型异常：包括 `RuntimeException`、`Error`。用户的 `Java` 程序可不做处理，系统将它们交给缺省的异常处理程序。类异常在会通过编译。

检查型异常：包括除了 `RuntimeException`、`Error` 以外的其他 `Exception` 的子类。在编译时能被编译器检测到异常，不能通过编译。采用声明或 `try`、`catch` 方式处理异常。

## Question2

**throws:**

在方法声明处声明抛出特定异常。从当前方法 ( `method` ) 中呼出自己的上位方法产生 `Exception`。

`throws` 用来表明一个方法可能抛出的各种异常。把例外推到自己不处理、把责任推给呼叫自己的方法 ( `method` ) 。

**throw:**

在方法中抛出具体的异常。它既被用于强制产生错误，也处理当前方法的错误后，通过给上游方法提供错误信息，可以感知上游方法也发生了错误。即，`throw` 是开发者强制发生 `exception`，在方法内执行例外处理。

## Question3

运行时异常：`NullPointerException`、`ArithmeticException`、`IndexOutOfBoundsException`

非运行时异常：`IOException`、`AWTException`

## Question4

因为 `RuntimeException` 是 `Exception` 的子类。所以不能进入第二个 `catch` 中。  
代码在 `solution` 文件里

## Question5

1. throws `IOException` -> throws `Exception`
2. throws `FileNotFoundException`

代码在 `solution` 文件里

## Question6

`methodA` 抛出一个异常！  
执行 `methodA` 的 `finally`！  
`methodB` 执行！  
执行 `methodB` 的 `finally`！

## Question7

```
-----  
error  
i in finally block:2  
1  
-----  
error  
i in finally:okfinally  
ok  
-----  
error  
i in finally:okfinally  
okfinally  
-----
```

原因：

`get0()`：进入 `try` 并抛弃异常 -> 输出 `err` -> `return` -> `finally` 输出

`get1()`：进入 `try` 并抛弃异常 -> 输出 `err` -> 不返回，字符串加法，`i = okfinally` -> 输出

`get2()`：前部分一样，但是 `finally` 部分上 `i` 是由 `StringBuilder i=new StringBuilder("ok")` 创造对象创造出来的。

`i` 指同一段内存地址，所以变 `return` 值。

## Question8

代码在solution文件里