

（源代码已上传至文件夹 Java07）

# Task1

几个错误和异常报告的栗子

**Exception 类：**

**ArithmeticException：** 算术运算异常，比如计算一个数除以零时。

**ClassNotFoundException：** 加载类时如果类不存在会报错。

**IOException：** 输入输出操作失败，比如在文件读写时格式错误。

**Error 类：**

**OutOfMemoryError：** 内存溢出，在为程序分配的内存不够时会报错。

**NoSuchFieldError：** 访问不存在的文件时会报错。

处理态度：

对于 **Exception** 的处理态度：

首先是捕获和处理：**Exception** 是可控制的异常，应该在可能的地方捕获并处理。这让程序在遇到问题时能恢复并继续执行。

其次资源清理：在捕获异常后，应该确保释放或清理所有相关资源。

然后错误恢复：尝试提供错误恢复选项，比如重试操作、撤销更改或恢复到稳定状态。

向用户的反馈：向用户提供错误信息，比如 **IDEA** 的错误更改建议。

对于 **Error** 的处理态度：

记录和分析：**Error** 指示的是严重问题，如系统崩溃等。应该记录尽可能多的信息，并进行详细分析。

不尝试捕获：正因为 **Error** 表示的是非常严重的问题，通常不在其内部捕获它们。捕获 **Error** 可能会隐藏严重的问题，导致无法正确识别和修复。

终止运行：**Error** 通常会导致程序终止。在一些情况下，需要紧急停止程序的执行。

保存状态：在程序终止前，尝试保存当前的程序的状态或数据，以便后续分析或恢复。

**Checked Exception：**

一般可预见，应该被捕获处理。一般与外部资源有关比如 **IOException**。

发生的原因：读取的外部文件不存在。

**Unchecked Exception：**

一般不可预见，与在编程过程中发生的逻辑错误有关，比如 **NullPointerException**。

发生的原因：调用未初始化的变量。

区别：

检查异常：必须在编译时处理，给我们提供了更严格的错误处理机制，在编译前就能提示我们修正错误。

非检查异常：更不起眼，我们的处理也更为灵活，但也可能因为未被捕获而在运行时导致程序崩溃。

# Task2

程序运行图如下：



# Task3

各种运行结果的截图



