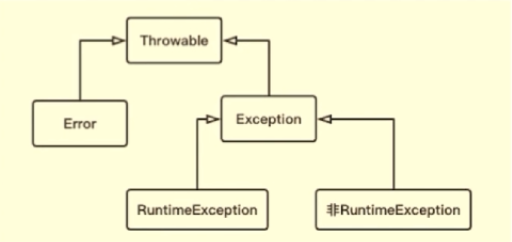
**10 Java常用类库与技巧**

一个基础问题：String、StringBuffer（线程安全且节省空间）、StringBuilder（高效但线程不安全）三者的区别？

答：暂略

Java异常处理机制主要回答了三个问题：

* What：异常类型回答了什么被抛出
* Where：异常堆栈跟踪（StackTrack）回答了在哪被抛出
* Why：异常信息回答为什么被抛出



Java异常体系

* RuntimeException：运行时异常。如数组下标越界、访问空指针等。不可预知，程序应该自行避免，如设置数组访问边界检查
* 非RuntimeException：可预知的，从编译器校验的异常。如IOException

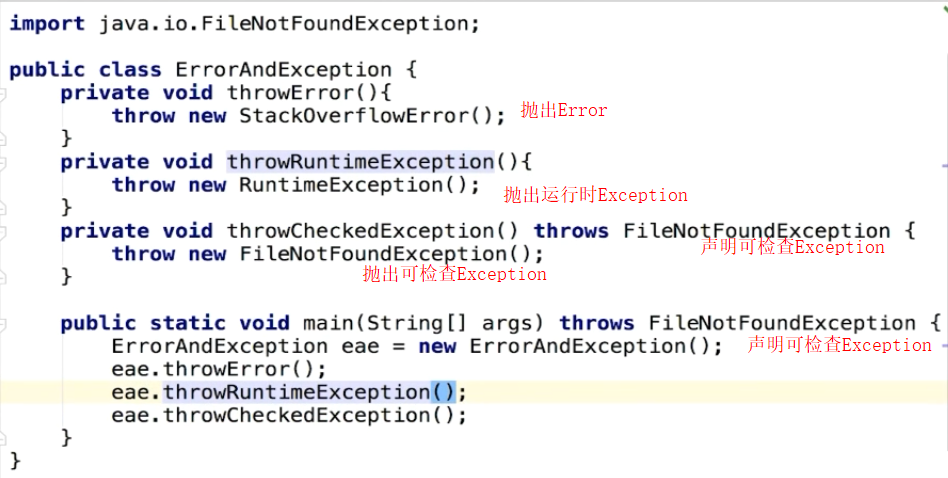
**Error和Exception的区别？**

答：体现了Java平台设计者对不同异常情况的分类——**前者是程序无法处理的错误，后者是可以处理的异常**

* Error：程序无法处理的系统错误。一般是与JVM相关的问题，如：系统崩溃、虚拟机错误、内存空间不足（OutOfMemoryError）、方法调用栈溢出（StackOverflowError）等。编译器对这类错误不做检查，遇到这类错误建议程序终止
* Exception：程序可以处理的异常，可以被捕获并可能被处理恢复

从责任角度看：

1. Error是JVM需要承担的责任
2. RuntimeException是程序应该负担的责任
3. Checked Exception可检查异常是Java编译器应该负担的责任



使用throw关键字在方法体内部抛出异常，用throw关键字在方法头声明异常

**常见Error和Exception：**

RuntimeException

1. NullPointException-空指针引用异常
2. ClassCastException-类型强制转换异常
3. IllegalArgumentException-传递非法参数异常
4. IndexOutOfBoundsException-下边越界异常
5. NumberFormatException-数字格式异常

非RuntimeException

1. ClassNotFoundException-找不到指定class的异常
2. IOException-IO操作异常，如打开一个不存在的文件

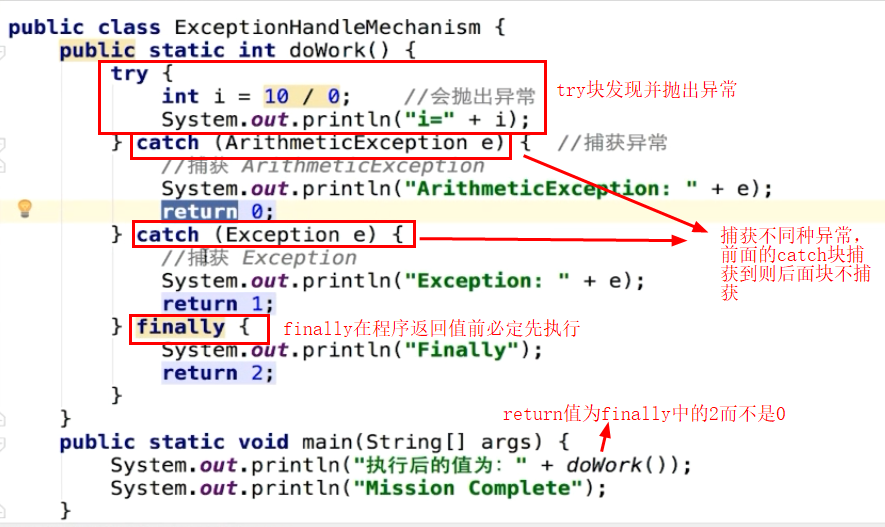
Error

1. NoClassDefFoundError-找不到class定义的异常，如类依赖的jar不存在
2. StackOverflowError-深递归导致JVM栈耗尽而抛出的Error
3. OutOfMemoryError-内存溢出异常

**ClassNotFoundException和NoClassDefFoundError的区别？**

答：

1. ClassNotFoundException:是一个Exception类型的异常。当动态加载Class找不到类时，例如执行Class.forName()、ClassLoader.loadClass()或ClassLoader.findSystemClass()的时候抛出该异常
2. NoClassDefFoundError：是一个Error类型。当编译时正常编译，但执行过程中找不到Class导致该Error。例如编译时new A()可以正常编译，但随后该A.class文件被删除，执行过程中new A()这条语句就会抛出NoClassDefFoundError

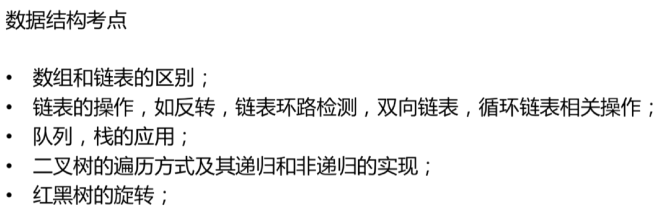


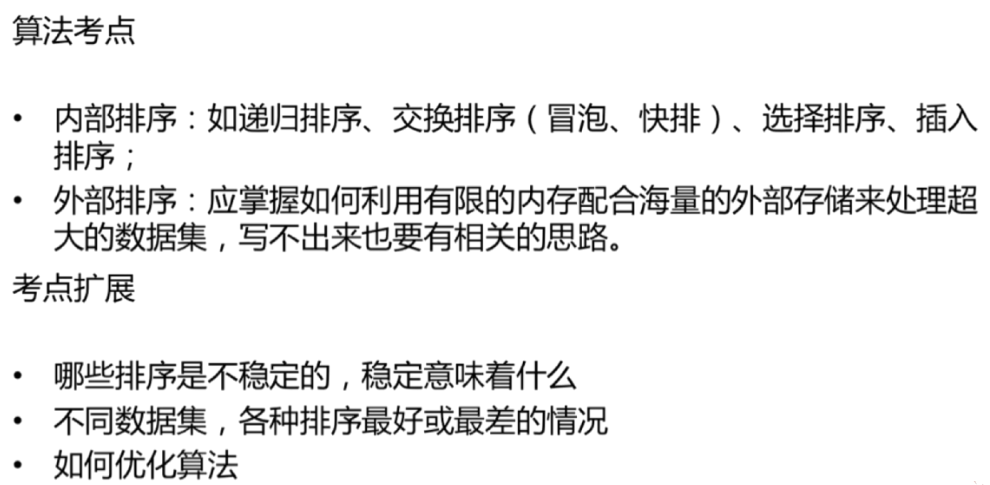
try-catch-finally块捕获发现异常处理方法

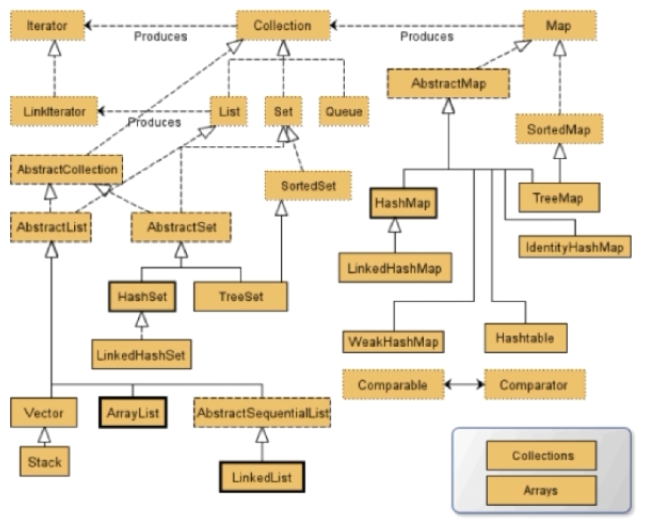
**使用try-catch-finally进行异常捕获及处理示例详见“剑指Java 10-2 Java异常要点分析”**

Java异常处理原则：

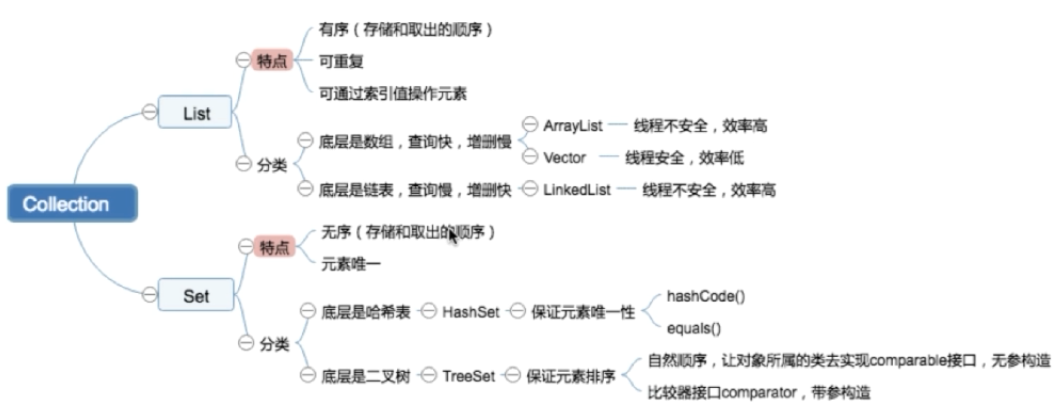
1. 具体明确：抛出的异常应能通过异常类名和message准确说明异常类型和原因
2. 提早抛出：尽可能早的发现并抛出异常，便于精确定位
3. 延迟捕获：异常的捕获和处理尽可能延迟，让掌握更多信息的作用域来处理异常







Java合集框架



List和Set的区别？

答：List是线性表，元素可重复，可以根据下标访问元素，有存储和取出的顺序；Set是集合，元素具有唯一性不可重复，元素没有下标，无存储和取出的顺序

ArrayList、Vector和LinkedList三者的区别？

答：

1. 从底层实现角度，ArrayList和Vector是数组实现的，查询快，增删慢；LinkedList是链表实现的，查询慢，增删快
2. 从线程安全角度，ArrayList和LinkedList是线程不安全的，方法中没有用到任何锁；Vector线程安全，它的public方法几乎都加上了synchronized，但只能在多线程环境中被串行访问，效率低

HashSet和TreeSet分别是HashMap和TreeMap的特殊形式：Set元素值存入Map<K,V>的键K中，V值设为 final Object PRESENT=new Object();

TreeSet的核心功能在于排序，可用自然排序Comparable和比较器Comparator两种方法进行排序（**待完善**）

**Comparable和比较器Comparator详细介绍待完善**