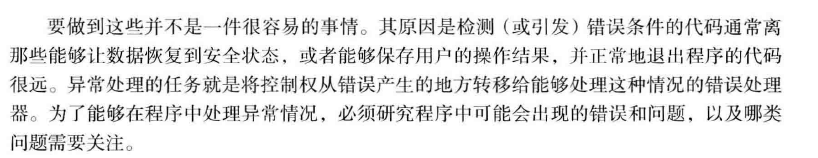
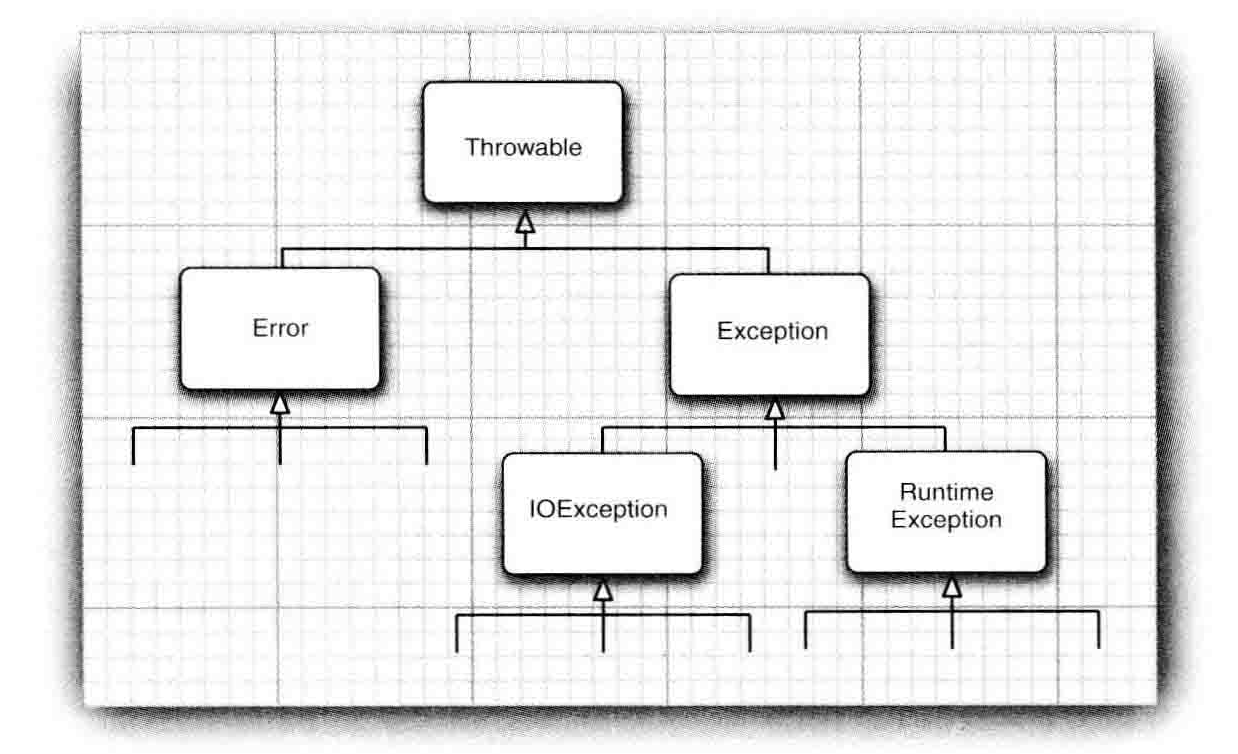
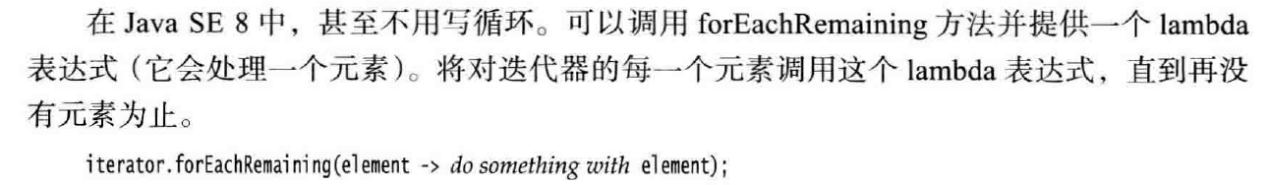
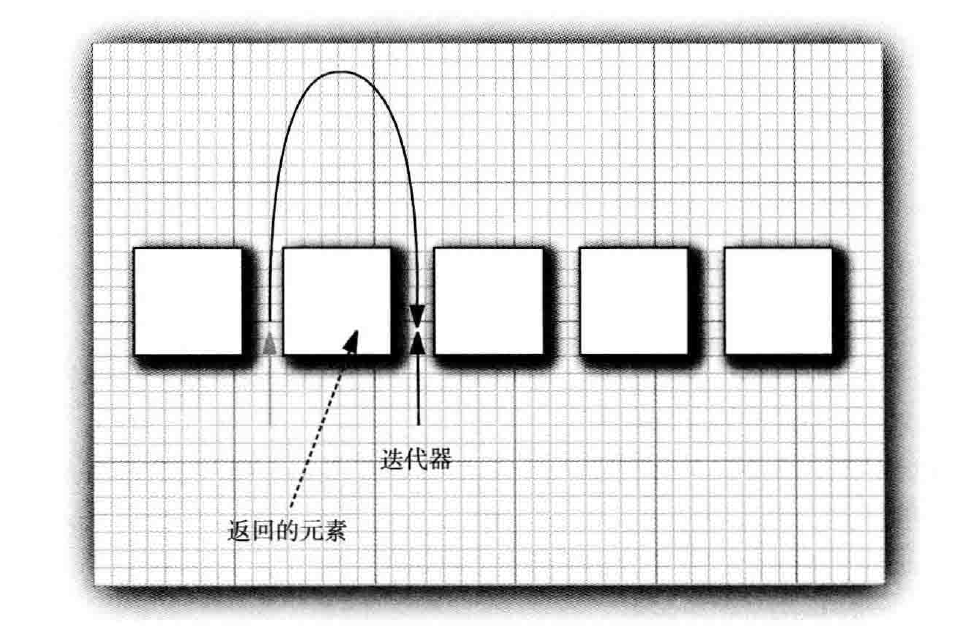
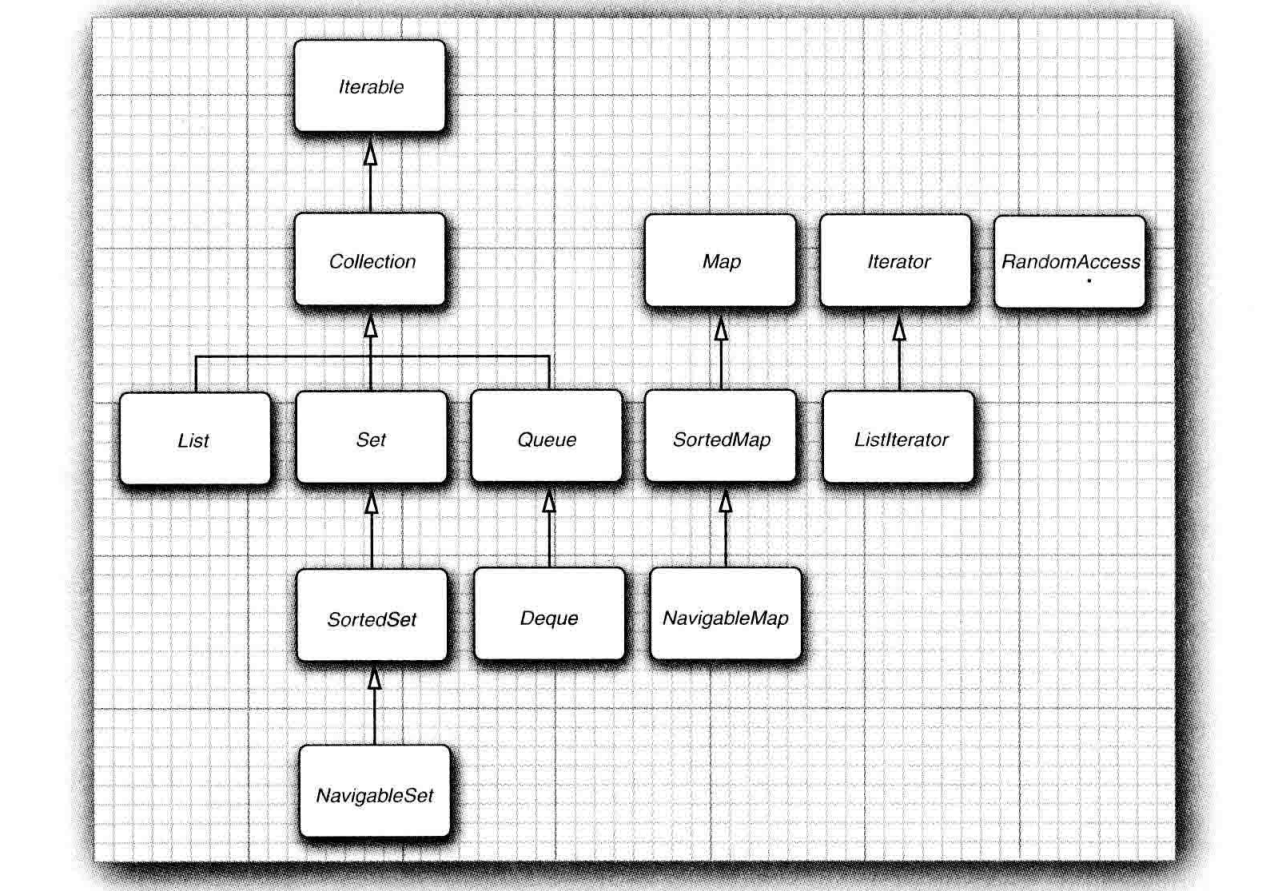
1. 为什么不用抽象类来代替接口 answer:一个类可以实现多个接口 却不能同时继承多个类
2. 接口在java8中会存在静态方法 比如Paths和 collections
3. Clone的浅拷贝 子对象不能被完全复制过来 还是会产生共享 深拷贝所有的域包括子对象都被拷贝过来 遇到有对象 但是要全部拷贝过来 就用深拷贝
4. 动态代理
5. 处理异常的缘由和意义
6. 
7. 像可能由系统自身产生的异常一般都抛出去 自己写的代码造成的问题 可以捕获异常并处理
8. 创建自己的自定义异常 一般继承异常父类 然后写两个构造方法
9. 当finally子句中存在return 程序最终返回的值根据finally字句得来
10. 在进行异常处理时 要将异常和异常处理分开来 最好不要一个子句处理一次异常 要把可能发生的异常的程序代码放在一起 然后通过不同的异常进行捕获
11. 日志级别:severe warning info config fine finer finest
12. 泛型让类的传入参数不限一种 使得以对象为传入参数的方法非常灵活
13. **集合**
14. 
15. 迭代器 在内存中是要跨越过一个元素之后再进行相关的操作 如如果要删除相邻元素必须调用两次next方法

iterator.remove();

iterator.next(); iterator.remove();



Collection:List set queue

Map:

各个集合的作用:

ArrayList:一种可以动态增长和缩减的索引序列

LinkedList:一种可以在任何位置进行高效的插入和删除操作的有序序列

ArrayDeque：一种用循环数组实现的双端队列

HashSet:一种没有重复元素的无序集合

TreeSet:一种有序集

EnumSet：一种包含枚举类型值的集

PriorityQueue：一种允许高效删除最小元素的集合

HashMap:一种存储键/值关联的数据结构

TreeMap:一种键值有序排列的映射表

EnumMap:一种键值属于枚举类型的映射表

LinkedHashMap:一种可以记住键/值添加次序的映射表

WeakHashMap:一种其值没用的可以进行垃圾回收的映射表

IdentityHashMap:一种用==而不是用equals比较键值的映射表

散列表的原理：set是一种散列表的实现 set的contains方法是被重新定义过的，它并不是查找所有的列表来判断是否有该元素 而是通过先确定桶的位置 再进行查找