**Java常用类库与技巧**

## java异常

### 异常处理机制主要回答了三个问题

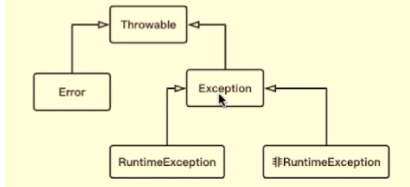
● what :异常类型回到了什么被抛出

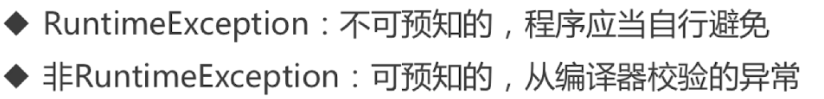
● where：异常堆栈回到了在哪被抛出

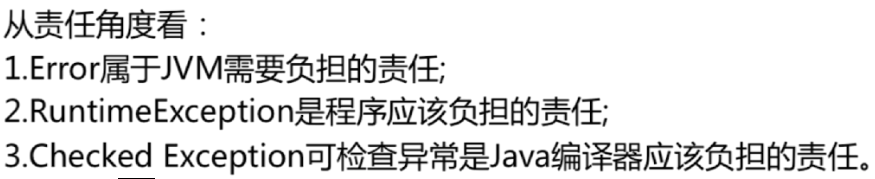
● why： 异常信息回答了为什么被抛出

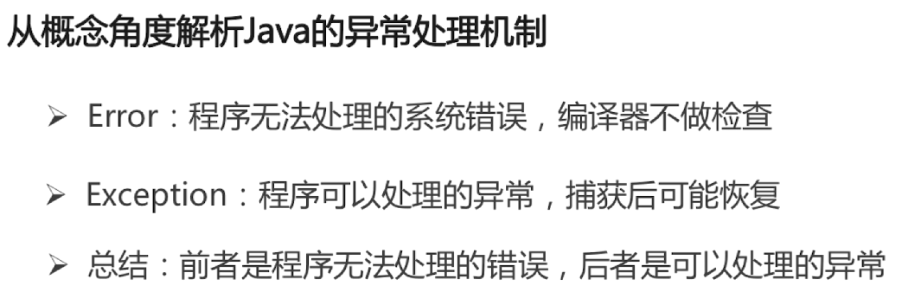
### Error和Exception的区别

Java异常体系：

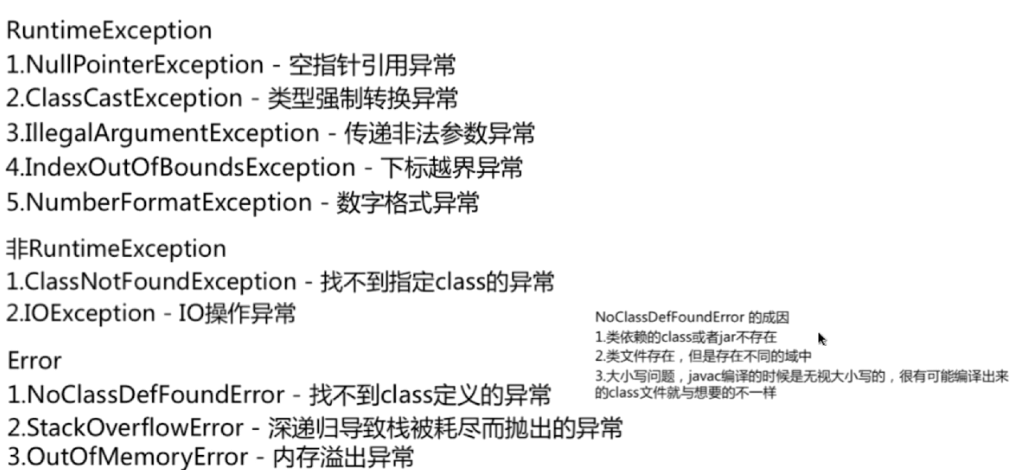




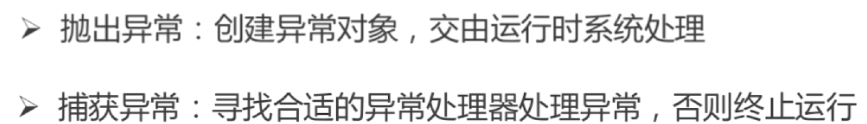




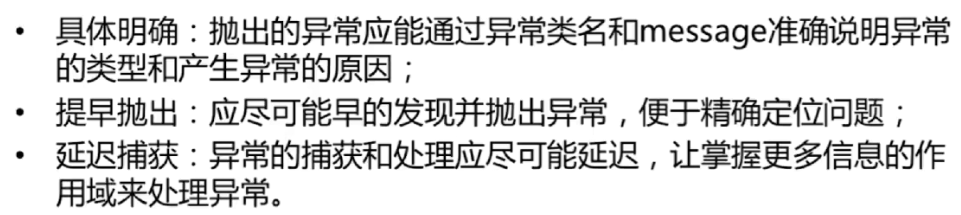
### 常见的error和Exception



### java的异常处理机制

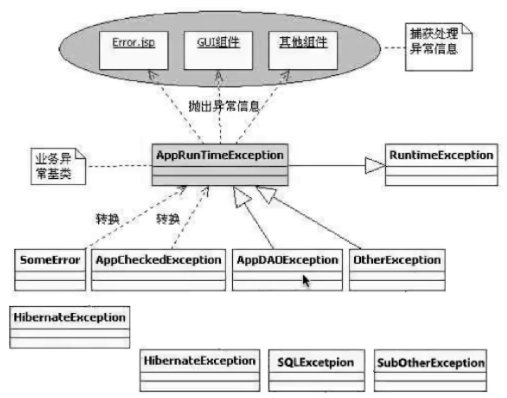


### java异常的处理原则



### 高效主流的异常处理框架



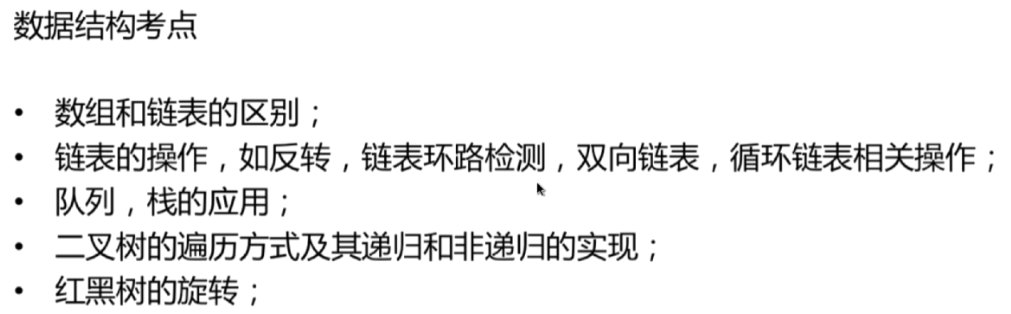


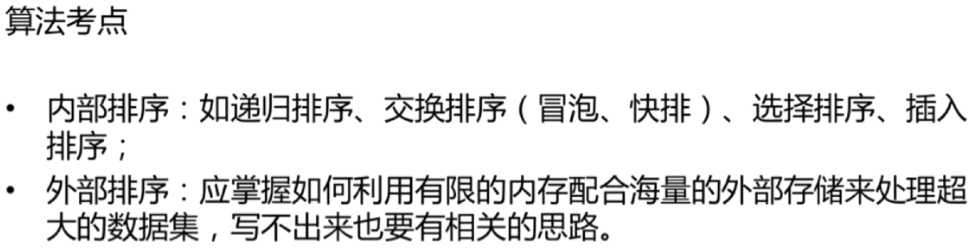
### try-catch的性能

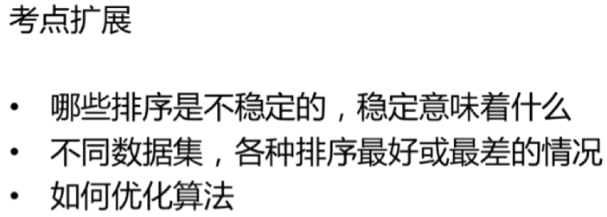
**java异常处理消耗性能的地方：**

1. **try-catch块影响jvm的优化**
2. **异常对象实例需要保存栈快照等信息，开销较大**

## java集合框架







### 集合之list和set

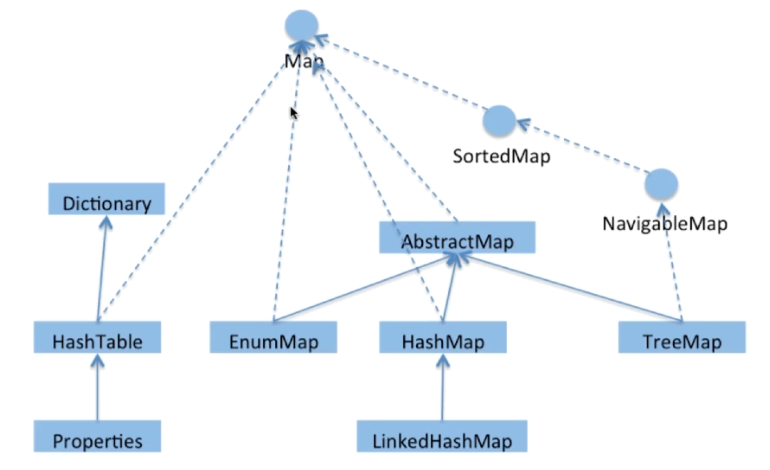


|  |
| --- |
| 备注：  Hashset底层使用hashMap实现 |

### 集合之map

|  |
| --- |
| 备注：  Map中的key是由set组织起来的，所以有去重功能 |

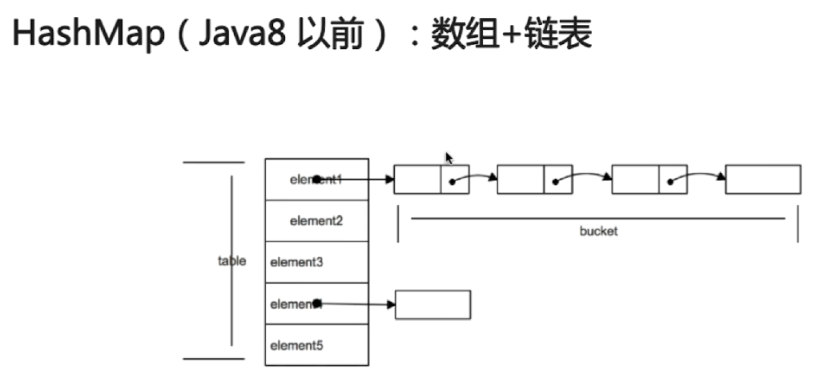
总体结构：

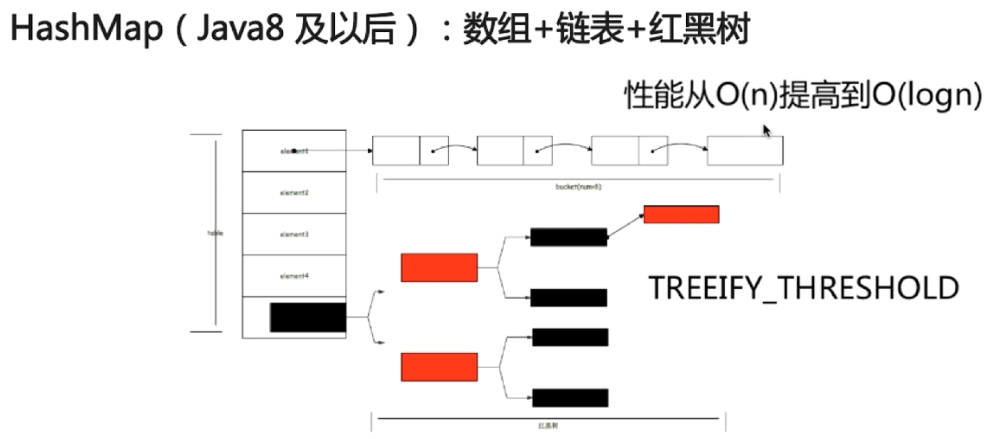


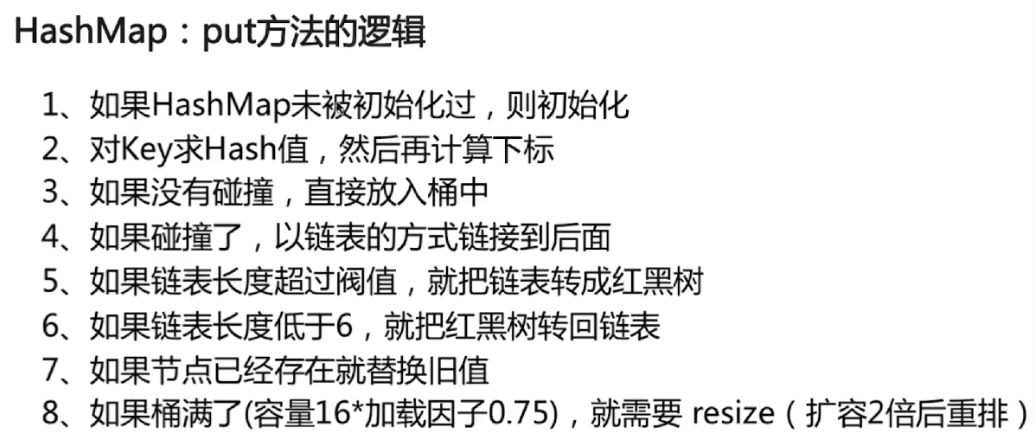
### hashMap，hashTable，conccurentHashMap的区别

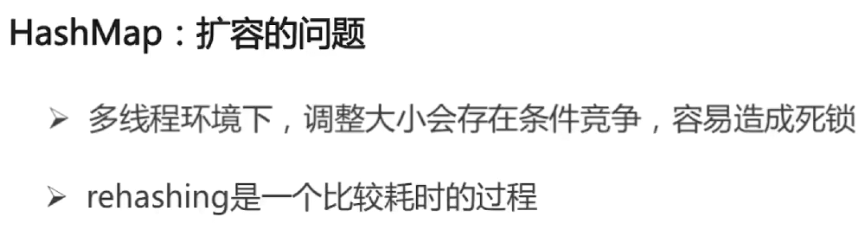
阿里常考hashMap

#### （1）hashMap



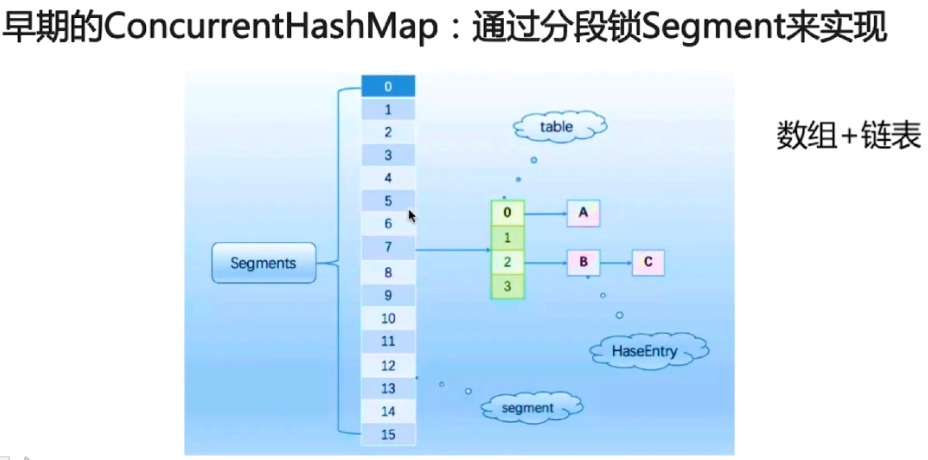


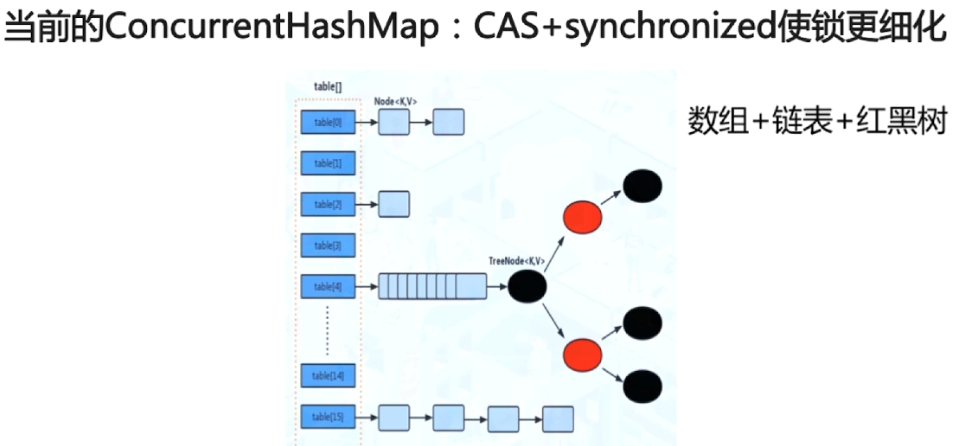


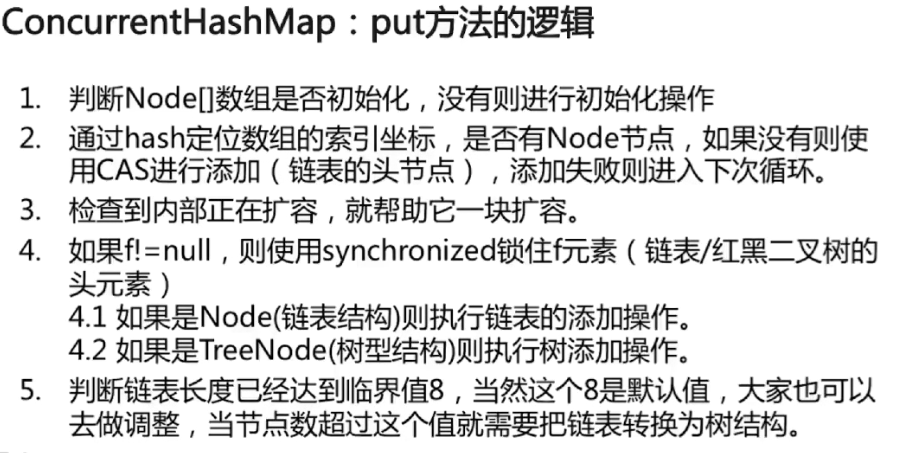


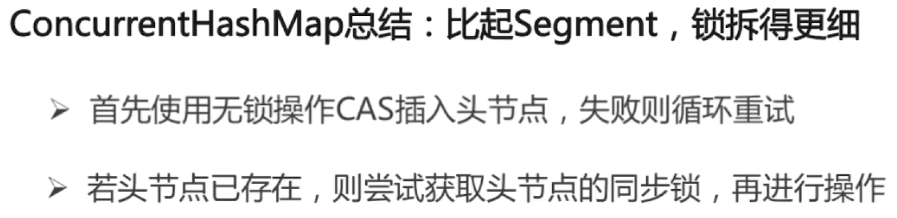
|  |
| --- |
| HashMap默认长度：16 |

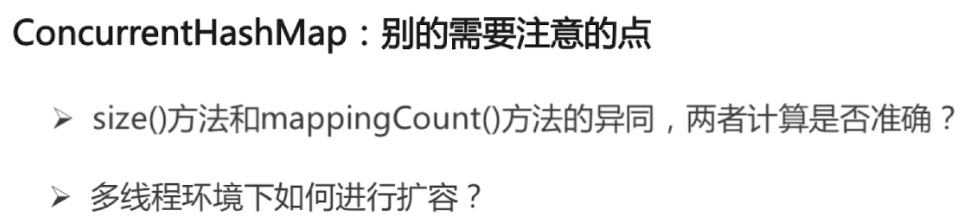
#### （2）concurrentHashMap



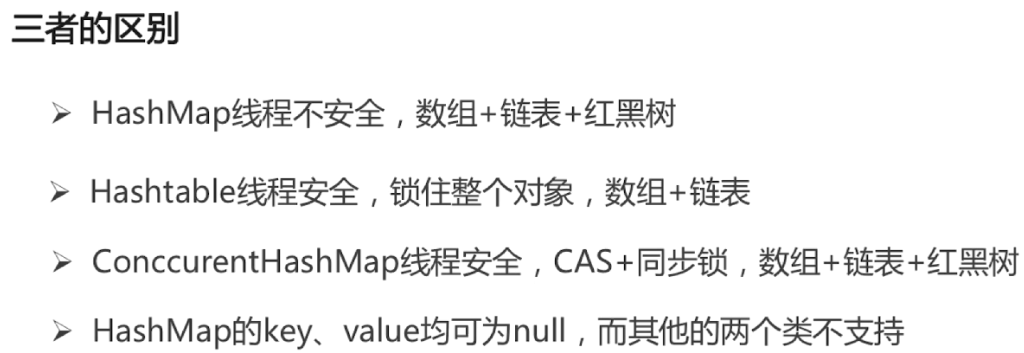








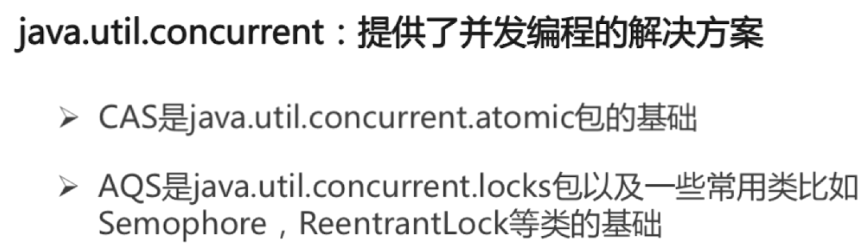
### （3）三者区别



|  |
| --- |
| 备注  ConcurrentHashMap出自juc包。 |

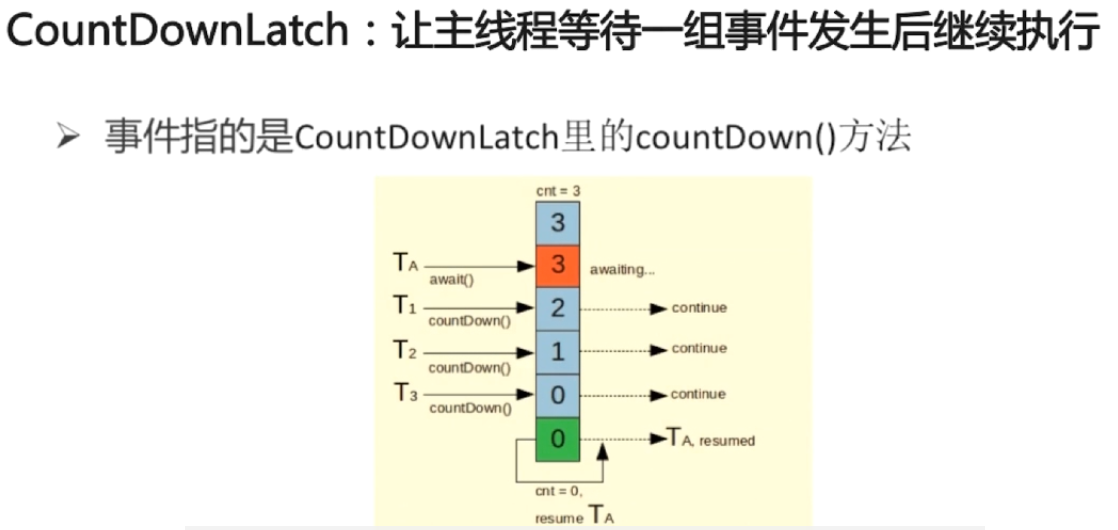
## J.U.C知识点梳理

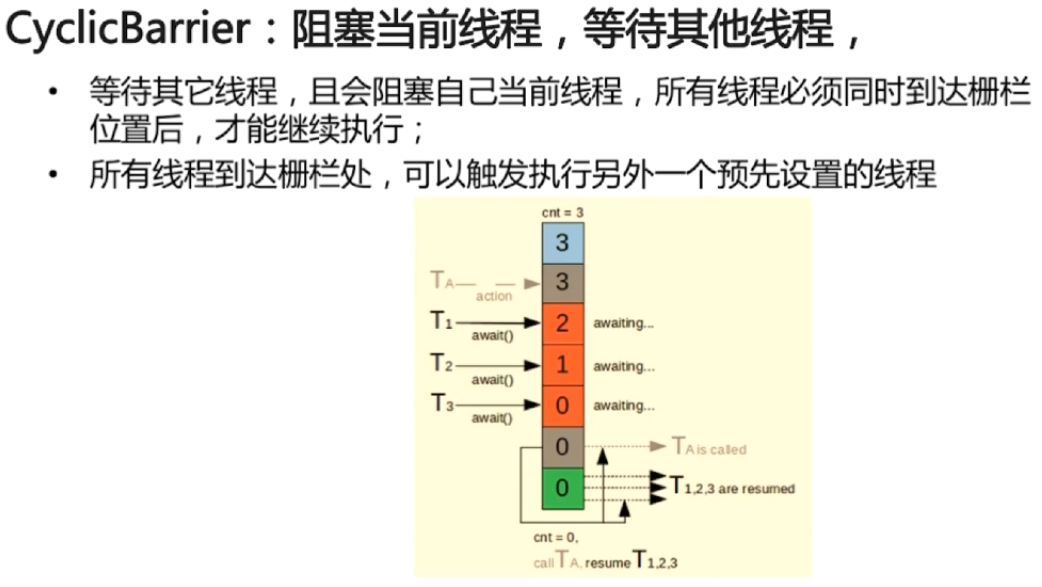
### 1.J.U.C并发包

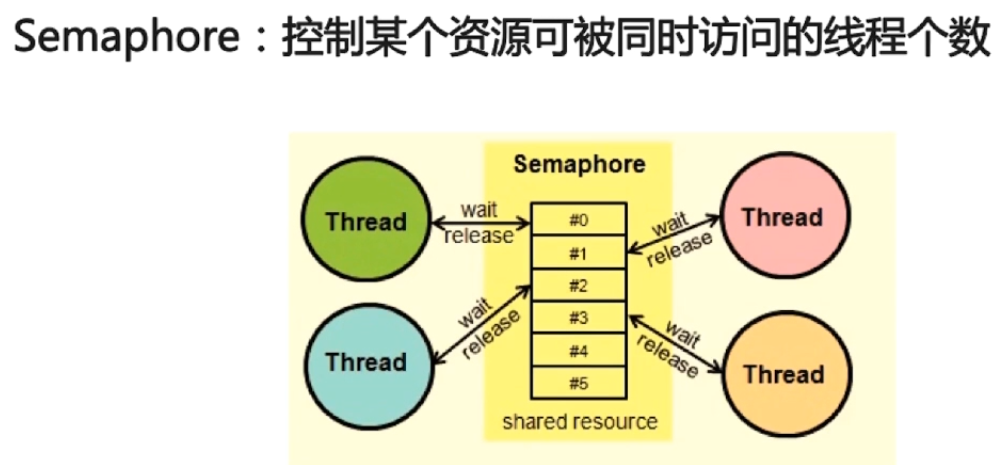


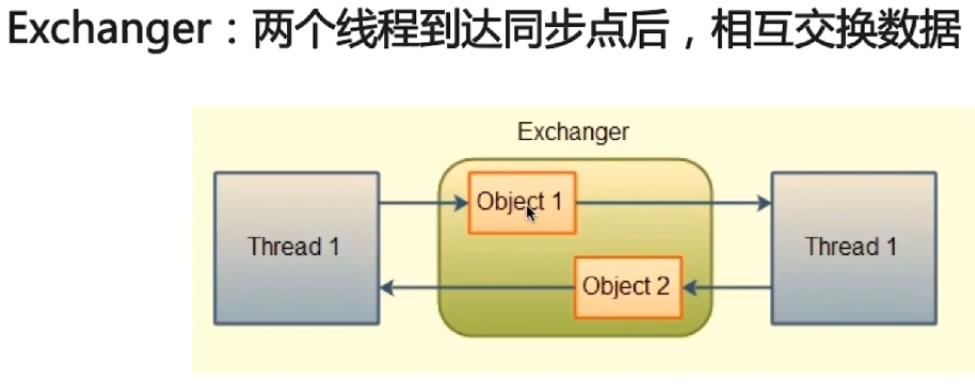


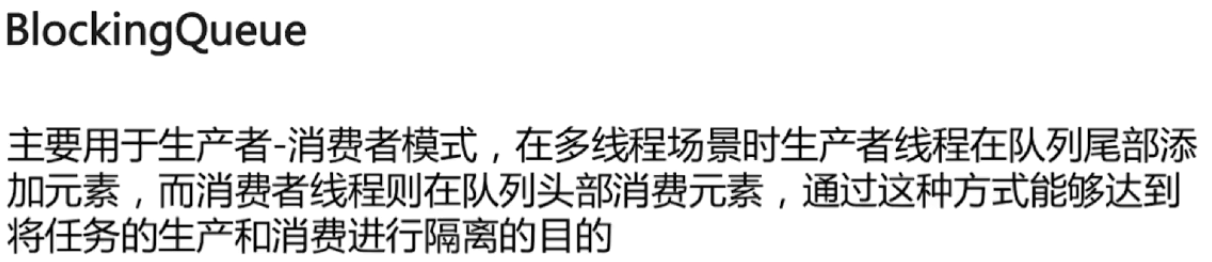


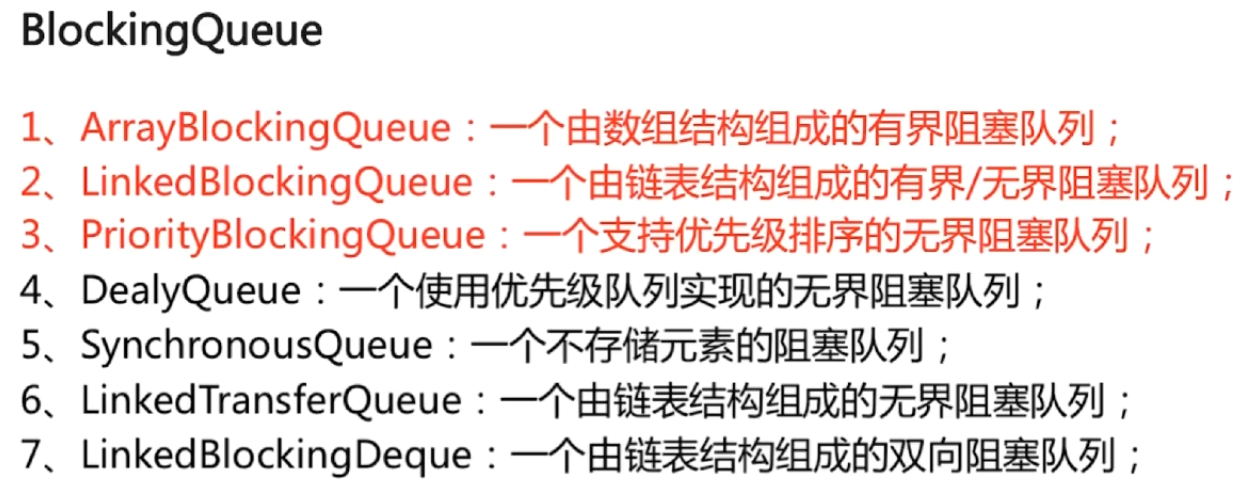












### BIO,NIO,AIO

