

ByteBuffer简述及注意点

NIO since Java 1.4

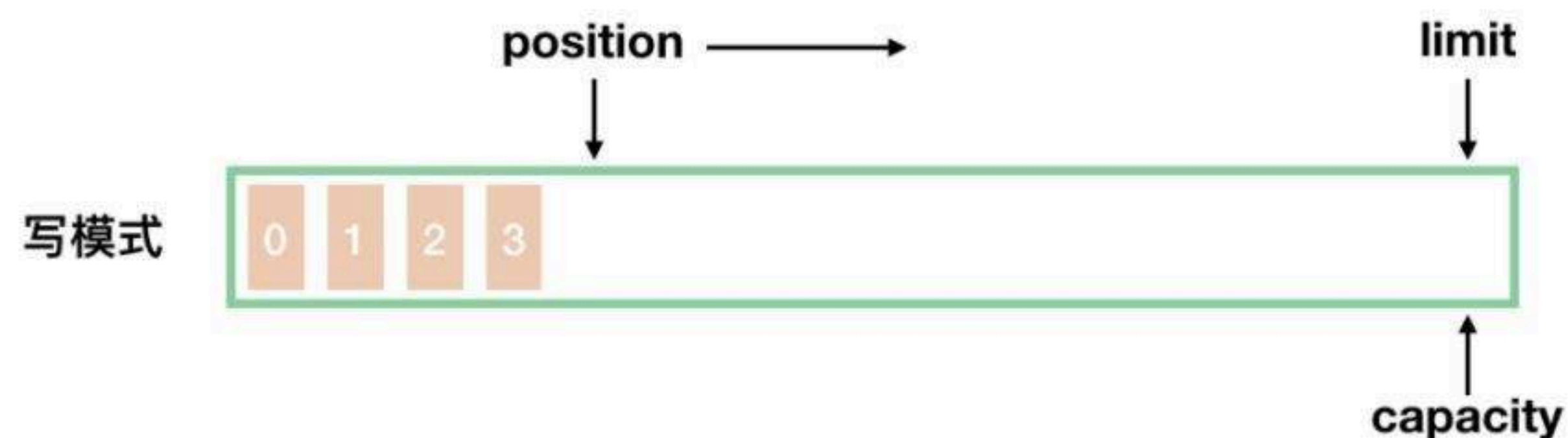
IO	NIO
Stream oriented	Buffer oriented
Blocking IO in socket	Non blocking IO in socketchannel
	Selectors

Buffer类体系

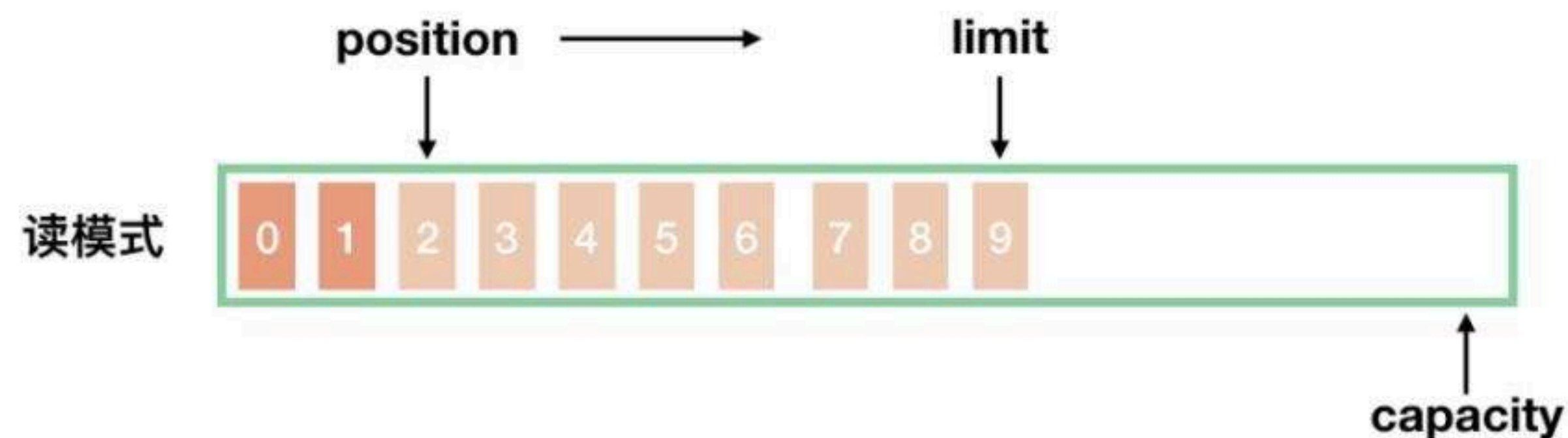
Channel类体系

ByteBuffer operation

- relative put get
- absolute put get
- clear flip rewind



写模式下, limit 等于 capacity, 此时 position 为 4



读模式下, limit 等于 Buffer 的实际数据大小, 此时 limit 为 10

- 注意点
 - 写完之后进行flip操作
 - absolute op 不影响position指针

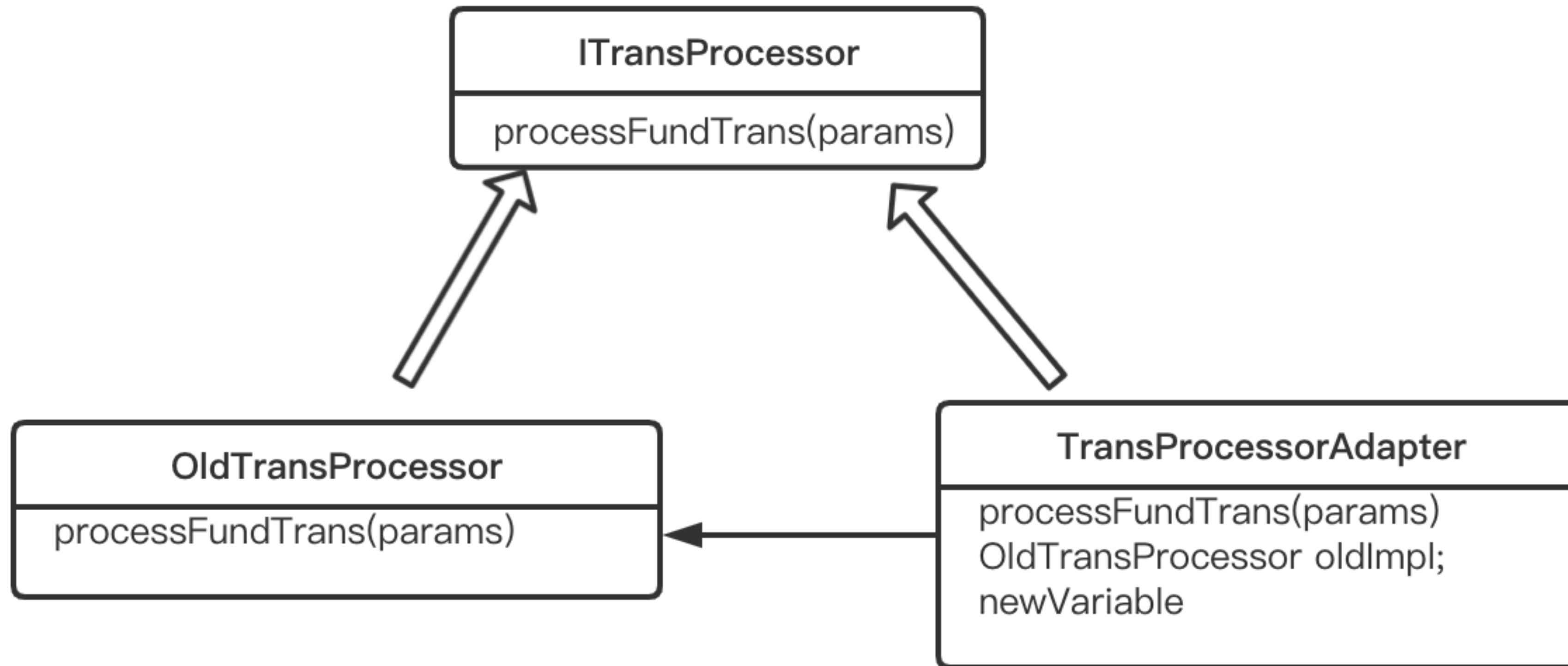
ByteBuffer vs DirectByteBuffer

- DirectByteBuffer
 - 读写访问速度较快
 - 创建和销毁花费较高
- DirectBuffer适用场景
 - 缓冲区复用
 - 自己管理内存，免受gc影响
 - 频繁文件读写、网络通信

Resource

- <https://docs.oracle.com/javase/8/docs/technotes/guides/io/index.html>
- <http://www.java-performance.com/>
- <https://docs.oracle.com/javase/8/docs/api/java/nio/ByteBuffer.html>

Adapter pattern in trans



Design Pattern —Adapter

- Adapter pattern
 - 定义适配器模式（有时候也称包装样式或者包装）将一个类的接口适配成用户所期待的。一个适配允许通常因为接口不兼容而不能在一起工作的类工作在一起，做法是将类自己的接口包裹在一个已存在的类中
- 优点：
 - 提高了类的复用
 - 增加了类的透明度
 - 灵活性好
- 缺点：
 - 过多地使用适配器，会让系统非常零乱，不易整体进行把握

总结

- ByteBuffer —》 DirectByteBuffer —》 ByteBuf
- Adapter Design Pattern