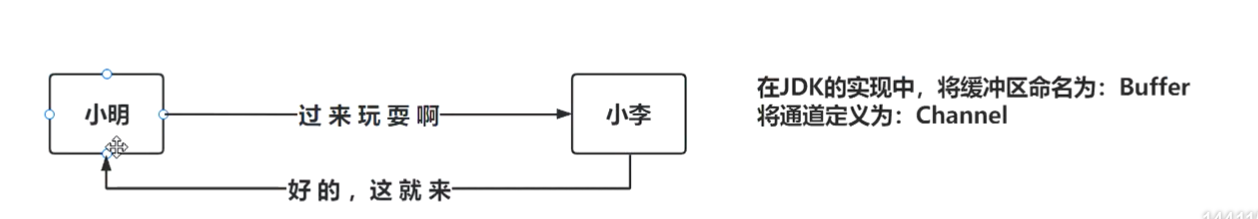


1. BIO
2. 原理：

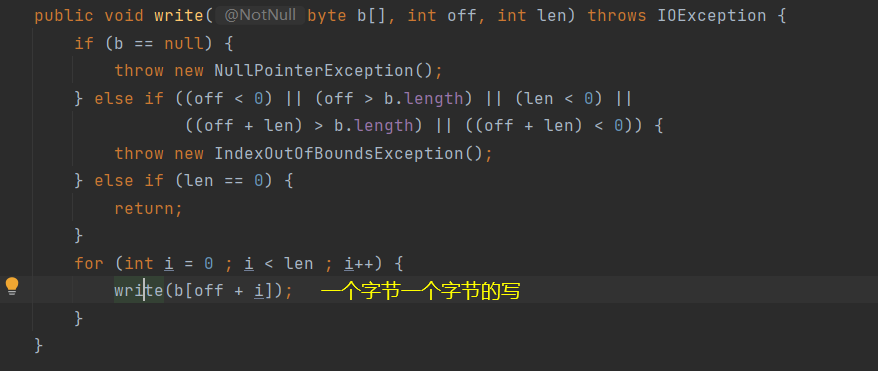
阻塞式IO，基于字节流：InputStream、OutputStream

原生的字节流模型：通信双方之间是基于字节流（InputStream/OutputStream），发送的数据是字节，即一个字节一个字节的发送与接收。

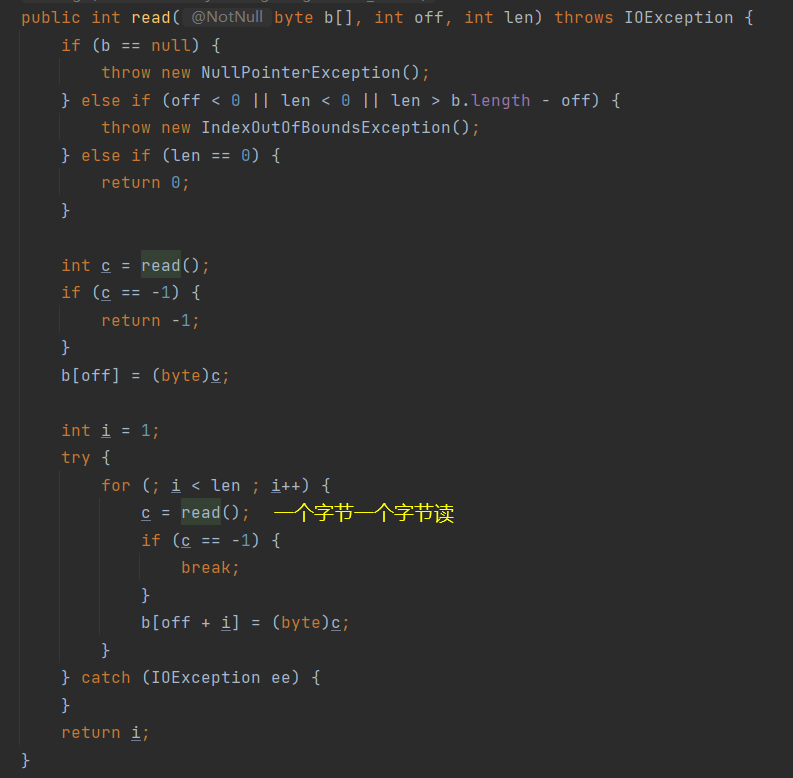


1. 源码

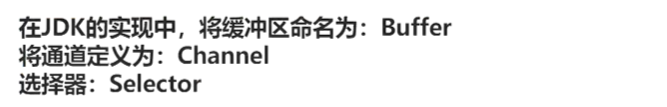
* OutputStream



* InputStream



1. NIO



三大组件：通道、Buffer、Selector

1. 原理：面向缓冲区编程

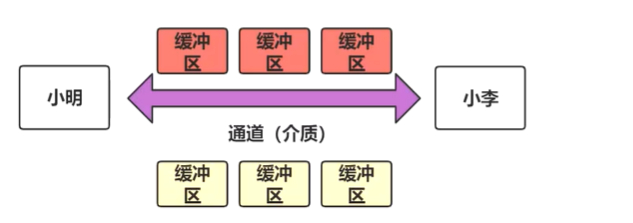
沟通的双方不再是基于字节流，而是面向缓冲区。

NIO是一个模型，模型定义通信双方需要构建通道Channel，通道中传输的是缓冲区Buffer。当只有一个线程处理多个通道的情况下，引入选择器Selector来监听通道。

与BIO区别：

BIO通信需要两个流（输入流、输出流）；

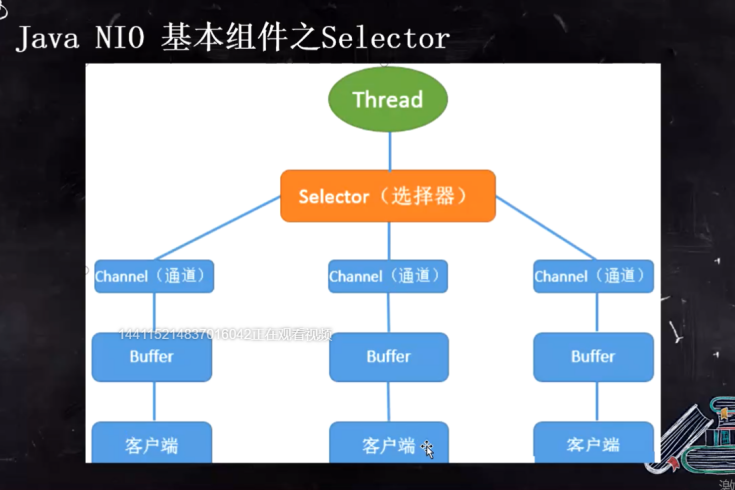
NIO模型构建一个通道就可以完成读写操作。



1. NIO基本组件之选择器Selector：监听通道

没有选择器：一个线程处理一个通道

有选择器：通过查询选择器，一个线程处理所有通道；把通道放到选择器上，选择器会监视每个通道的状态，如果某个通道有buffer数据过来，线程就可以查询到是哪个通道传输数据，然后进行业务处理。



1. 源码

* Selector

多路（通道）复用（单线程）器，并且只能通道只能是SelectableChannel 的子类。

其选择的通道一定是非阻塞的，否则还是会阻塞线程。

1. 编码流程
2. 构建通道（可以设置为阻塞式、也可以设置为非阻塞；不一定要有多路复用器）
3. 构建数据的Buffer
4. 使用Buffer