



# NIO的概述以及应用

Mic



# 什么是NIO

NIO 从JDK1.4 提出的，本意是New IO，它的出现为了弥补IO的不足，提供了更高效的方式

# NIO的新特性

新特性	具体描述
基于 通道 (Channel) & 缓冲区 (Buffer) 操作	<ul style="list-style-type: none"><li>• 通道 (Channel) : 1个新的原始 I/O 抽象</li><li>• 缓冲支持 (Buffer) : 为所有的原始类型提供 缓冲支持</li><li>• 具体操作: 1. 数据 从 通道 读取到 缓冲区、2. 数据 从 缓冲区 写入 通道</li></ul>
非阻塞 (Non - Bloking)	提供 多路 非阻塞式的 I/O 操作 (即 当线程 从通道读取数据到缓冲区时, 线程依然可以进行其他事情)
选择器 (Selectors)	<ul style="list-style-type: none"><li>• 用于 监听 多个通道的事件, 如: 连接打开、数据到达</li><li>• 单个线程可监听多个数据通道</li></ul>
其他	<ul style="list-style-type: none"><li>• 提供 字符集编码、解码的解决方案: Java.nio.charset</li><li>• 支持锁 &amp; 内存映射文件的文件访问接口</li></ul>

## 核心组件

通道（`Channel`）：Java NIO数据来源，可以是网络，也可以是本地磁盘

缓冲区（`Buffer`）：数据读写的中转区，后续会单独讲

选择器（`Selectors`）：异步IO的核心类，可以实现异步非阻塞IO，一个  
selectors可以管理多个通道Channel

# IO和NIO的区别

类型	面向 操作区域	处理数据 (字节流 & 字符流)	IO阻塞 / 非阻塞
Java IO	直接 面向 最初的数据源	<ul style="list-style-type: none"><li>• 每次读取时 = 读取所有字节 / 字符，无缓存</li><li>• 无法前后移动读取流中的数据</li></ul>	<p>阻塞</p> <ul style="list-style-type: none"><li>• 当1个线程在读 / 写时：当 数据被完全读取 / 写入完毕前 &amp; 数据未准备好时，线程不能做其他任务，只能一直等待，直到数据准备好后继续读取 / 写入，即阻塞</li><li>• 当线程处于活跃状态时 &amp; 外部未准备好时，则阻塞</li></ul>
Java NIO	面向缓冲区	<ul style="list-style-type: none"><li>• 先将数据读取 到 缓存区</li><li>• 可在缓冲区中前后移动流数据</li></ul>	<p>非阻塞</p> <ul style="list-style-type: none"><li>• 当1个线程向某通道发送请求 要求 读 / 写时，当 数据被完全读取 / 写入完毕前 &amp; 数据未准备好时，线程可以做其他任务（控制其他通道），直到数据准备好后再 切换回该通道，继续读取 / 写入，即选择器（Selector）的使用</li><li>• 外部准备好时才唤醒线程，则不会阻塞</li></ul>