

# ♥Java IO知识体系详解♥

## 知识体系



- Java IO/NIO/AIO - Overview
  - 主要梳理Java IO/NIO/AIO的知识体系
- Java IO - 分类(传输, 操作)
  - 主要从传输方式和数据操作两个方面分析Java IO的分类
- Java IO - 设计模式(装饰者模式)
  - Java I/O 使用了装饰者模式来实现
- Java IO - 源码: InputStream

- 主要从JDK源码角度分析InputStream
- Java IO - 源码: OutputStream
  - 主要从JDK源码角度分析 OutputStream
- Java IO - 常见类使用
  - 主要介绍Java IO常见类的使用, 包括: 磁盘操作, 字节操作, 字符操作, 对象操作和网络操作
- IO 模型 - Unix IO 模型
  - 主要简要介绍 Unix I/O 5种模型, 并对5大模型比较
- Java IO - BIO 详解
  - BIO就是: blocking IO。最容易理解、最容易实现的IO工作方式, 应用程序向操作系统请求网络IO操作, 这时应用程序会一直等待; 另一方面, 操作系统收到请求后, 也会等待, 直到网络上有数据传到监听端口; 操作系统在收集数据后, 会把数据发送给应用程序; 最后应用程序受到数据, 并解除等待状态
- Java NIO - 基础详解
  - 新的输入/输出 (NIO) 库是在 JDK 1.4 中引入的, 弥补了原来的 I/O 的不足, 提供了高速的、面向块的 I/O
- Java NIO - IO多路复用详解
  - 主要对IO多路复用, Reactor模型以及Java NIO对其的支持
- Java AIO - 异步IO详解
  - 主要对异步IO和Java中对AIO的支持详解。
- 源码着手分析了 Java NIO 对零拷贝的实现, 主要包括基于内存映射 (mmap) 方式的 MappedByteBuffer 以及基于 sendfile 方式的 FileChannel。最后在篇末简单的阐述了一下 Netty 中的零拷贝机制, 以及 RocketMQ 和 Kafka 两种消息队列在零拷贝实现方式上的区别。
- [Java N\(A\)IO - 框架: Netty](#)
  - Netty是一个高性能、异步事件驱动的NIO框架, 提供了对TCP、UDP和文件传输的支持。作为当前最流行的NIO框架, Netty在互联网领域、大数据分布式计算领域、游戏行业、通信行业等获得了广泛的应用, 一些业界著名的开源组件也基于Netty构建, 比如RPC框架、zookeeper等