

---

## 概括的解释下线程的几种状态

1. **就绪(Runnable)**: 线程准备运行, 不一定立马就能开始执行。
2. **运行中(Running)**: 进程正在执行线程的代码。
3. **等待中(Waiting)**: 线程处于阻塞的状态, 等待外部的处理结束。
4. **睡眠中(Sleeping)**: 线程被强制睡眠。
5. **I/O 阻塞(Blocked on I/O)**: 等待 I/O 操作完成。
6. **同步阻塞(Blocked on Synchronization)**: 等待获取锁。
7. **死亡(Dead)**: 线程完成了执行。

## Java 集合类里面基本的接口有哪些

- **Collection**: 代表一组对象, 每一个对象都是它的子元素。
- **Set**: 不包含重复元素的 Collection。
- **List**: 有顺序的 collection, 并且可以包含重复元素。
- **Map**: 可以把键(key)映射到值(value)的对象, 键不能重复。

## Iterator 和 ListIterator 的区别

- Iterator 可用来遍历 Set 和 List 集合, 但是 ListIterator 只能用来遍历 List。
- Iterator 对集合只能是前向遍历, ListIterator 既可以前向遍历也可以后向遍历。
- ListIterator 实现了 Iterator 接口, 并包含其他的功能, 比如: 增加、替换元素, 获取前一个和后一个元素的索引等等。

## Enumeration 和 Iterator 的区别

- java 中的集合类都提供了返回 Iterator 的方法, 就是迭代器, 它和 Enumeration (枚举) 的主要区别其实就是 Iterator 可以删除元素, 但是 Enumeration 却不能。
- 使用 Iterator 来遍历集合时, 应使用 Iterator 的 remove() 方法来删除集合中的元素, 使用集合的 remove() 方法将抛出 ConcurrentModificationException 异常。
- Enumeration 接口的功能和 Iterator 接口的功能是重复的。此外, Iterator 接口添加了一个可选的移除操作, 并使用较短的方法名。新的实现应该优先考虑使用 Iterator 接口而不是 Enumeration 接口。
- Enumeration 速度是 Iterator 的 2 倍, 同时占用更少的内存。但是, Iterator 远远比 Enumeration 安全, 因为其他线程不能够修改正在被 iterator 遍历的集合里面的对象。

## 介绍一下 HTTP 的响应结构

### 请求报文的组成:

- 请求方法。
- 请求的资源的 URI。
- 协议版本。
- 可选的请求首部字段。
- 内容实体。

### 响应报文的组成:

- 协议版本。
- 状态码。
- 用于解释状态码的原因短语。

- 可选的响应首部字段。
- 实体主体。

#### 响应：

- 响应码：回应客户端此次响应是否成功。如：404(找不到请求的资源)，500(服务器内部错误)，200(成功响应)等。
- 消息头：服务器与客户端通信的暗码，告诉客户端该怎么执行某些操作。
- 响应正文：传递服务器响应给客户端要显示的内容，可以是下载文件或者显示界面。

### final、finalize 和 finally 各自的作用

final 是一个修饰符，可以修饰变量、方法和类。如果 final 修饰变量，意味着该变量的值在初始化后不能被改变。

finalize 方法是在对象被回收之前调用的方法，给对象自己最后一个复活的机会，但是什么时候调用 finalize 没有保证。

finally 是一个关键字，与 try 和 catch 一起用于异常的处理。finally 块一定会被执行，无论在 try 块中是否有发生异常。

### 说一下你了解的几种进程间的通信方式

- 管道 pipe：管道是一种半双工的通信方式，数据只能单向流动，而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。
- 高级管道 popen：将另一个程序当做一个新的进程在当前程序进程中启动，则它算是当前程序的子进程，这种方式我们成为高级管道方式。
- 有名管道 named pipe：有名管道也是半双工的通信方式，但是它允许无亲缘关系进程间的通信。
- 消息队列 MessageQueue：消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。
- 共享存储 SharedMemory：共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是最快的 IPC 方式，它是针对其他进程间通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号两，配合使用，来实现进程间的同步和通信。
- 信号量 Semaphore：信号量是一个计数器，可以用来控制多个进程对共享资源的访问。它常作为一种锁机制，防止某进程正在访问共享资源时，其他进程也访问该资源。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。
- 套接字 Socket：套接口也是一种进程间通信机制，与其他通信机制不同的是，它可用于不同及其间的进程通信。
- 信号 sinal：信号是一种比较复杂的通信方式，用于通知接收进程某个事件已经发生。