# 一、线程

1、线程的生命周期：

start（）-线程就绪--》线程运行--》线程结束/线程阻塞。

2、线程的一些关键方法.

Thread.sleep(1000) 当前线程休眠1秒，不释放锁。wait（）释放锁等待。

Join（）等待线程执行完

notify（）用于唤醒等待的对象

# 二、线程池

1、为什么要使用线程池

如果短时间内有大量的线程任务。这时需要大量的创建和销毁。线程池的使用将使得系统使用线程无需重复创建和释放。

2、怎样使用？

executors（线程池）- > 创建一个线程池，设置线程池的参数。

3、线程生命周期

execute（）执行线程--》核心线程处理---核心线程满---》放到工作队列---工作队列满---》创建非核心线程---超过线程最大值---》执行拒绝策略

# 三、threadlocal的使用

使用场景：1、数据库连接

2、实现session共享

threadlocal的原理：

threadlocal的set方法通过当前线程绑定map。

threadlocal的get方法通过当前线程获取set的值。

# 四、jvm

1、jvm的内存模型

方法区，堆，栈，本地方法区，程序计数器

2、类加载机制

Bootrap加载器（根加载器）、extension类加载（拓展加载器）、system类加载器、自定义类。

加载-》验证-》准备-》解析-》初始化（静态资源）-》实例化-》销毁

3、jvm的垃圾回收器（GC）

问题：GC的思维导图

垃圾回收的是内存，--》垃圾的定义—》根据垃圾定义写出算法—》将算法和程序结合，制造回收器—》回收器的的种类—》hotspot vm / vm

1. 标记-清除法 ---》可达性分析--再统一清除
2. 标记-整理法---》可达性分析-》整理归档
3. 标记-复制算法—》将edan数据复制到s0，再复制到s1，删除edan和s0

分代收集器：

年轻代收集器：serail，parallel scavenge，parnew

老年代收集器：serail old，parallel scavenge old，cms

G1和ZGC拥有两种时代都收集

4、jvm内存设置（性能调优）

Jconsole、

堆栈内存相关：

-Xms 设置初始堆的大小

-Xmx 设置最大堆的大小

-Xmn 设置年轻代大小，相当于同时配置

-XX:NewSize和-XX:MaxNewSize为一样的值

-Xss 每个线程的堆栈大小

-XX:NewSize 设置年轻代大小(for 1.3/1.4)

-XX:MaxNewSize 年轻代最大值(for 1.3/1.4)

-XX:NewRatio 年轻代与年老代的比值(除去持久代)

-XX:SurvivorRatio Eden区与Survivor区的的比值

-XX:PretenureSizeThreshold 当创建的对象超过指定大小时，直接把对象分配在老年代。

-XX:MaxTenuringThreshold设定对象在Survivor复制的最大年龄阈值，超过阈值转移到 老年代

垃圾收集器相关：

-XX:+UseParallelGC：选择垃圾收集器为并行收集器。

-XX:ParallelGCThreads=20：配置并行收集器的线程数

-XX:+UseConcMarkSweepGC：设置年老代为并发收集。

-XX:CMSFullGCsBeforeCompaction=5 由于并发收集器不对内存空间进行压缩、整理， 所以运行一段时间以后会产生“碎片”，使得运行效率降低。此值设置运行5次GC以后对内 存空间进行压缩、整理。

-XX:+UseCMSCompactAtFullCollection：打开对年老代的压缩。可能会影响性能，但是 可以消除碎片

辅助信息相关：

-XX:+PrintGCDetails 打印GC详细信息

-XX:+HeapDumpOnOutOfMemoryError让JVM在发生内存溢出的时候自动生成内存快照, 排查问题用

-XX:+DisableExplicitGC禁止系统System.gc()，防止手动误触发FGC造成问题.

-XX:+PrintTLAB 查看TLAB空间的使用情况

5、编译器优化（分析逃逸）

6、执行模式

# 五、注解

1、注解的初始化

**一个注解准确意义上来说，只不过是一种特殊的注释而已，如果没有解析它的代码，它可能连注释都不如。**而解析一个类或者方法的注解往往有两种形式，一种是编译期直接的扫描，一种是运行期反射。

2、自定义注解

元注解：

@Target：注解的作用目标

@Retention：注解的生命周期

@Documented：注解是否应当被包含在 JavaDoc 文档中

@Inherited：是否允许子类继承该注解

# 六、Spring

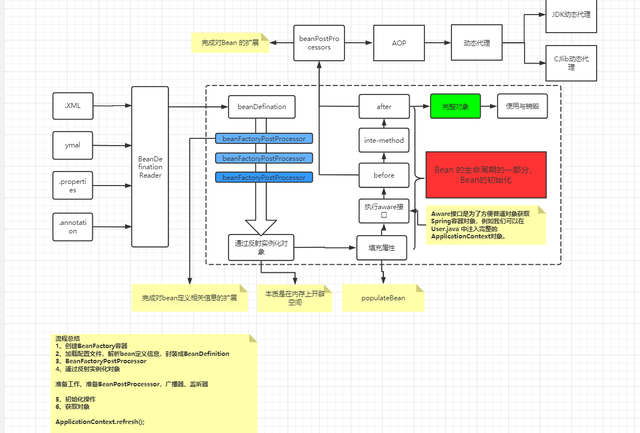
1、讲讲你对spring的理解.

Spring是一个轻量级的java容器框架，它通过IOC，AOP等理念，融入其他第三方框架。是企业级开发更简单，开发更专注于业务代码的开发。

Ioc（**Inversion of Control，控制反转**）：控制反转，Spring帮助我们生产对象、管理对象和对象之间的依赖关系，降低了代码的耦合度，方便后期对项目做维护。

利用了工厂模式将对象交给器管理，你只需要在spring配置文件总配置相应的bean，以及设置相关的属性，让spring器来生成类的实例对象以及管理对象。在spring器启动的时候，spring会把你在配置文件中配置的bean都初始化好，然后在你需要调用的时候，就把它已经初始化好的那些bean分配给你需要调用这些bean的类，

而不需要你在A里面new这些bean了。



AOP（**Aspect Oriented Programming，面向切面编程**）

面向切面（方便）编程，可以为某一类对象进行监督和控制，也就是在调用这类对象的具体方法的前后去调用你指定的功能模块，从而达到对一个模块扩充的功能，可以通过注解或者XML文档进行实现；

**Spring的初始化几种方式：**

Spring boot为我们提供了以下几种方案供我们选择:

1）springboot提供的ApplicationRunner与CommandLineRunner接口

2） Spring Bean初始化的init-method、PostConstruct注解、

3）InitializingBean、BeanPostProcessor接口

4） Spring的事件机制: 实现 ApplicationListener 接口

# 七、定时器qartz

# 八、缓存redis

# 九、队列MQ

# 十、所搜引擎ES