# Java 学习笔记

Jdk1.8后hashmap换成数组-链表-红黑树，防止哈希碰撞，因子为0.75，初始容量16，这种方法除了插入变慢，其余操作都加速。

方法存放在堆里面的元空间（直接使用物理内存）

通常定义一个基本的数据类型的变量，一个对象的引用函数调用都放在JVM的栈空间，而通过new关键字和构造器创建的对象则放在堆空间，通常大量对象都放在堆空间。堆是垃圾收集器管理的主要区域。栈和堆的大小都可以通过JVM启动参数调整。

栈操作快但是容量小。

Lambda 表达式需要函数式接口的支持（接口中只有一个抽象方法的接口）

1. 进程与线程的区别

主要区别在于不同的操作系统资源管理方式，进程有独立的地址空间，一个进程崩溃之后不会影响别的进程，而线程是进程内的不同的执行路径，线程拥有自己的堆栈变量，无独立的地址空间。一个线程崩溃就是整个进程的崩溃。切换进程时，资源耗费比较大。线程执行开销小，但是不利于资源的管理和保护，线程相反。

1. 数据库索引，为什么使用b+树

[索引](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95/5716853" \t "/Users/linyue/Documents\\x/_blank)是对[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "/Users/linyue/Documents\\x/_blank)表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息，每一索引都有一个相对应的搜索码。

二叉搜索树，平衡二叉树，B树，B+树，层层递进，能够更快的查找到特定数据，节点可以存储更多数据。

1. spring与springboot管理bean的方式

通常spring使用的是ApplicationContext负责读取bean配置文档，管理bean的加载，实例化，维护bean之间的依赖关系，负责bean的声明周期，国际化支持、aop、事务。它在初始化的时候将配置文件中bean以及相对应关系的配置都加入到ApplicationContext,通过一系列的转换将这些bean实例化，bean被它进行了管理，所以ApplicationContext就扮演了一个容器的角色。

springboot没有xml文件管理，所以所有的Bean管理都放入在一个配置类中实现。  
配置类就是类上具有@Configuration的类。这个类就相当于之前的applicationContext.xml

1. 反射创建对象与正常创建对象的区别

Java中使用new，判断内存中是否已经有那个唯一的Class，如果没有对象Java首先把,java文件编译成，class的字节码，类加载器把这个加载到jvm内存中，一个Class对象生成，并放入方法区中。如果有，则根据Class对象获取有关的构造器进行实例化。

反射创建对象是动态创建，在编译的时候没有加载进去，在需要调用的时候进行加载。

1. mysql主键是怎么自增的（其实是说底层的Auto-INC锁）
2. Msql里面的锁

## 共享锁与排他锁

共享锁（读锁）：其他事务可以读，但不能写。

排他锁（写锁） ：其他事务不能读取，也不能写。

死锁：

死锁是指两个或多个事务在同一资源上相互占用，并请求锁定对方占用的资源，从而导致恶性循环。

当事务试图以不同的顺序锁定资源时，就可能产生死锁。多个事务同时锁定同一个资源时也可能会产生死锁。检测死锁，数据库实现了死锁检测机制，INNODB存储引擎能够检测到死锁循环依赖并立即返回一个错误。当死锁发生后，只有其中一个事务或者多个事务回滚，才能打破死锁，处理方法可以是将目前拥有最少行级排它锁的事物进行回滚。当出现外部锁或者表锁的死锁时，设置等待超时参数来解决。

Java 多态三个条件：继承，重写，向上转型

Java集合类：

list集合，包含ArrayList和linkedlist，一个底层是数组一个是链表，数组遍历速度快，链表增删快，工作中遍历的需求较多，增加元素往往都是从尾部插入，时间复杂度也为O(1).

set集合，包含hashset，treeset，linkedHashset，保证数据的唯一性，treeset是可以排序的set。

map集合，包含hashmap，linkedhashmap，treemap，考虑到线程安全，可以用concurrenthasmap。

queue队列。

HashMap为什么要用红黑树：

在jdk1.8后，在链表长度达于8的时候，将后面的数据存在红黑树，加快检索速度（数组+链表-》数组+树），时间复杂度从On到logn。