# 基础

equals和==的区别

为什么重写equals要重写hashcode

StringBuilder和StringBuffer的区别

String是如何实现它的不可变的，为什么要设置String为不可变？

接口和抽象类的区别

重写和重载的区别

深拷贝和浅拷贝的区别

java三大特性

object的方法

有些object的方法已经比较过时了，为什么还要保留

static关键字，什么作用，放在哪里

如何高效的进行数组拷贝（System.arraycopy()的原理）

面向对象有哪些特征

# 集合

## List

### 【2】arraylist原理

### 扩容底层原理，访问速度，和linkedlist的区别

### 【2】linkedlist原理

## 【4】hashmap

### 你会如何设计一个hashmap

### 为什么size是2的幂

### resize的过程

### 效率受到哪些因素的影响

### 扰动函数的作用

### hash值的计算

### 为什么使用红黑树，红黑树的特点

### 【2】1.7和1.8的改动

### put数据的过程

### 链表很长会怎样，什么情况转红黑树

### 对hashmap的resize进行优化

### concurrenthashmap

### 【2】和hashmap的比较

### 线程安全的问题

## hashtable

### 和hashmap的区别

### 理想的查找效率是多少

### 怎么解决hash碰撞

# 并发

### 多线程

### 【2】线程池核心参数

### 【2】线程池执行过程

### 常用线程池

### 拒绝策略有哪些

### 使用executor的弊端

### 【2】sleep和wait的区别

### 为什么sleep是Thread的方法，不是Object的

### 【2】volatile的作用

## ThreadLocal

### 是什么

### 底层原理

## AQS

### 框架理解

### 源码理解

## 锁

### 【2】synchronized

### 性能为什么提高了

### 锁升级过程

### 【2】和lock的区别

### 使用场景

### 常见的锁机制

### ReentrantLock的原理

### CAS底层原理

# JVM

## 【2】内存

### 每个区域的介绍

### 版本变化

### 堆和栈

### 栈的默认大小

### 字符串常量池是在堆吗？

### OOM和StackOverFlow

### 内存泄漏

### 引用类型（强软弱虚四种）

## 【2】GC算法

### 优缺点

### 使用场景

### 对应的垃圾收集器

### 分区

### 判断方法（GC触发条件）

## GC器

### 判断对象死亡的方法

### GCroot有哪些，为什么选择他们作为GCroot

### 什么时候STW

### 引用计数的循环引用的情况

## 类加载过程

### 双亲委派机制

### 原理

### 作用

# 设计模式

## 【3】单例

### 为什么双检测

### 为什么用volatile修饰

### 手写懒汉式，DLC懒汉式

### 装饰模式和代理模式的区别

## 生产者消费者

## 工厂方法模式

### 手写一个工厂方法模式