Java

2.hashMap原理，java8做的改变

从结构实现来讲，HashMap是数组+链表+红黑树（JDK1.8增加了红黑树部分）实现的。HashMap最多只允许一条记录的键为null，允许多条记录的值为null。HashMap非线程安全。ConcurrentHashMap线程安全。解决碰撞：当出现冲突时，运用拉链法，将关键词为同义词的结点链接在一个单链表中，散列表长m，则定义一个由m个头指针组成的指针数组T，地址为i的结点插入以T(i)为头指针的单链表中。Java8中，冲突的元素超过限制（8），用红黑树替换链表。

3.String 和 StringBuilder 的区别

1）可变与不可变：String不可变，每一次执行“+”都会新生成一个新对象，所以频繁改变字符串的情况中不用String，以节省内存。

2）是否多线程安全：StringBuilder并没有对方法进行加同步锁，所以是非线程安全的。StringBuffer和String均线程安全。

7.ArrayList与 LinkedList 的区别？

最明显的区别是

ArrrayList 底层的数据结构是数组，支持随机访问，而 LinkedList 的底层数据结构书链表，不支持随机访问。使用下标访问一个元素，ArrayList 的时间复杂度是 O(1)，而 LinkedList 是 O(n)。LinkedList是双向链表

9.抽象类是什么？它与接口有什么区别？你为什么要使用过抽象类？

抽象类是指不允许被实例化的类；一个类只能使用一次继承关系。但是，一个类却可以实现多个interface。

abstract class和interface所反映出的设计理念不同。其实abstract class表示的是"is-a"关系，interface表示的是"like-a"关系

实现抽象类和接口的类必须实现其中的所有方法。抽象类中可以有非抽象方法。接口中则不能有实现方法。但在Java8中允许接口中有静态默认的方法。

接口中定义的变量默认是public static final 型，且必须给其初值，所以实现类中不能重新定义，也不能改变其值。抽象类中的变量默认是 friendly 型，其值可以在子类中重新定义，也可以重新赋值。

子类中实现父类中的抽象方法时，可见性可以大于等于父类中的；而接口实现类中的接口 方法的可见性只能与接口中相同（public）。

用抽象类是为了重用。减少编码量，降低耦合性。

13.object中定义了哪些方法？

clone(), equals(), hashCode(), toString(), notify(), notifyAll(),

wait(), finalize(), getClass()

14.Java泛型和类型擦除？

泛型即参数化类型，在创建集合时，指定集合元素的类型，此集合只能传入该类型的参数。类型擦除：java编译器生成的字节码不包含泛型信息，所以在编译时擦除：1.泛型用最顶级父类替换；2.移除。

15.说出 5 个 JDK 1.8 引入的新特性？

Java 8 在 Java 历史上是一个开创新的版本，下面 JDK 8 中 5 个主要的特性：

Lambda 表达式；允许像对象一样传递匿名函数 Stream API，充分利用现代多核 CPU，可以写出很简洁的代码 ；Date 与 Time API，最终，有一个稳定、简单的日期和时间库可供你使用 扩展方法，现在，接口中可以有静态、默认方法； 重复注解，现在你可以将相同的注解在同一类型上使用多次。

线程有哪些状态？线程池的参数有哪些？

ConcurrentHashMap的原理

乐观锁、分布式锁

第十章 Lock和synchronized的区别

　　1）Lock是一个接口，而synchronized是Java中的关键字，synchronized是内置的语言实现，synchronized是在JVM层面上实现的，不但可以通过一些监控工具监控synchronized的锁定，而且在代码执行时出现异常，JVM会自动释放锁定，但是使用Lock则不行，lock是通过代码实现的，要保证锁定一定会被释放，就必须将 unLock()放到finally{} 中；

　　2）synchronized在发生异常时，会自动释放线程占有的锁，因此不会导致死锁现象发生；而Lock在发生异常时，如果没有主动通过unLock()去释放锁，则很可能造成死锁现象，因此使用Lock时需要在finally块中释放锁；

　　3）Lock可以让等待锁的线程响应中断，线程可以中断去干别的事务，而synchronized却不行，使用synchronized时，等待的线程会一直等待下去，不能够响应中断；

　　4）通过Lock可以知道有没有成功获取锁，而synchronized却无法办到。

　　5）Lock可以提高多个线程进行读操作的效率。

　　在性能上来说，如果竞争资源不激烈，两者的性能是差不多的，而当竞争资源非常激烈时（即有大量线程同时竞争），此时Lock的性能要远远优于synchronized。所以说，在具体使用时要根据适当情况选择。

13. 实现线程的几种方法

(1)继承Thread类，重写run函数

(2)实现Runnable接口，重写run函数

(3)实现Callable接口，重写call函数

JVM

1.JVM如何加载一个类的过程，双亲委派模型中有哪些方法

类加载过程：加载、验证（验证阶段作用是保证Class文件的字节流包含的信息符合JVM规范，不会给JVM造成危害）、准备（准备阶段为变量分配内存并设置类变量的初始化）、解析（解析过程是将常量池内的符号引用替换成直接引用）、初始化。

双亲委派模型中方法：双亲委派是指如果一个类收到了类加载的请求，不会自己先尝试加载，先找父类加载器去完成。当顶层启动类加载器表示无法加载这个类的时候，子类才会尝试自己去加载。当回到最开的发起者加载器还无法加载时，并不会向下找，而是抛出ClassNotFound异常。

方法：启动（Bootstrap）类加载器，标准扩展（Extension）类加载器，应用程序类加载器（Application ），上下文(Custom)类加载器。意义是防止内存中出现多份同样的字节码 。

2.GC算法（什么样的对象算是可回收对象，可达性分析），CMS收集器

jvm是如何判断一个对象已经变成了可回收的“垃圾”，一般是两个方法：引用记数法和根搜索算法。引用记数法没办法解决循环引用的问题，所以用根搜索。从一系列的”GC Roots“对象开始向下搜索，搜索走过的路径称为引用链。当一个对象到”GC Roots“之间没有引用链时，被称为引用不可达。引用不可到的对象被认为是可回收的对象。

几种垃圾收集器：1，Serial New/Serial Old(串行)，2，Parrallel New (并行)，3，Parrallel Scavenge，4，Parrallel Old，5，CMS（CMS收集器是一个以获得最短回收停顿时间为目标的收集器，它是一种并发收集器，采用的是Mark-sweep算法。），6，G1（是一款并行与并发收集器，并且可建立可预测的停顿时间模型，整体上是基于标记清理，局部采用复制）

内存模型和类加载机制

Spring

什么是IOC

AOP的原理和应用

Spring事务的传播机制

Spring MVC的工作原理是怎样的？

@Controller和@RestController的区别

MyBatis中使用#和$书写占位符有什么区别？

Mybatis的缓存机制

mysql常用函数

MYSQL 有哪些存储引擎

Mysql中的myisam与innodb的区别

Mysql的默认隔离级别

sql优化、联合索引、分区、分表分库

Redis的数据结构

Redis的应用场景

Redis分布式锁

Redis 的持久化方式

Redis部署方式

Redis的过期策略和内存淘汰策略

什么是缓存穿透，缓存击穿，缓存雪崩

说一些你比较常用linux指令

编辑文件、查看文件最后10行内容

运行 Spring Boot 有哪几种方式？

用过Spring Cloud哪些组件

用过哪些设计模式