**1JAVA基础**

**1 static的作用**

1、修饰函数的局部变量：

特点：有默认值0，只执行一次，运行一开始就开辟了内存，内存放在全局

2、修饰全局函数和全局变量：

特点：只能在本源文件使用

3、修饰类里面的成员变量：

特点：和1差不多，定义多个static y,但只有一个y,不进入类的大小计算，不依赖于类对象的存在而存在（可直接调用，要进行外置声明）

4、修饰类的成员函数

**2 Final 的作用**

1修饰方法是

2修饰成员变量

3修饰参数问题

**1接口与抽象的类的区别**

1抽象类是用来捕捉子类的通用性的，接口则是抽象方法的集合

2 抽象类不能被实例化能能用作超类，是用来创建继承层级里的模板

3接口知识一种形式接口自身不能做任何事情

**2抽象类的关键字是什么**

1子类使用extends继承抽象类如果子类不是抽象类的话他需要声明方法

2 子类使用关键字implements来实现是接口

3 public protected default 接口默认修饰public

3**线程有几种方法实现**

1 继承Thread类 重写run方法（每次创建一个Threadz子类对象启动new Thread子类（）.start()）

2实现Runnable接口重写run(不论创建多少歌线程只需要创建一个Runnable接口实现的对象)

3实现Callable接口重写call方法（有返回值）麻烦每次启动都要创建一个

4 线程池 Executor 类（Executor 负责现成的使用和调度的根接口 ExecutorService 线程池的主要接口 ThreadPoolExecutor 线程池的实现类 -ScheduledExecutorService 接口，负责线程的调度ScheduledThreadPoolExecutor (extends ThreadPoolExecutor implements ScheduledExecutorService)

**4 为什么要用使用线程池**

诸如 Web 服务器、数据库服务器、文件服务器或邮件服务器之类的许多服务器应用程序都面向处理来自某些远程来源的大量短小的任务。请求以某种方式到达服务器，这种方式可能是通过网络协议（例如 HTTP、FTP 或 POP）、通过 JMS 队列或者可能通过轮询数据库。不管请求如何到达，服务器应用程序中经常出现的情况是：单个任务处理的时间很短而请求的数目却是巨大的

**5 线程池有哪几种锁 这些锁的作用是什么**

公平锁/非公平锁 可重入锁独享锁/共享锁 互斥锁/读写锁 乐观锁/悲观锁 分段锁 偏向锁/轻量级锁/重量级锁 自旋锁

**6 常用的几种线程池**

**1** newCachedThreadPool

创建一个可缓存线程池，如果线程池长度超过处理需要，可灵活回收空闲线程，若无可回收，则新建线程

**2 newFixedTheadPool**

创建一个指定工作线程数量的线程池。每当提交一个任务就创建一个工作线程，如果工作线程数量达到线程池初始的最大数，则将提交的任务存入到池队列中。

**3 newSingleThreadExecutor**

创建一个单线程化的Executor，即只创建唯一的工作者线程来执行任务，它只会用唯一的工作线程来执行任务，保证所有任务按照指定顺序(FIFO, LIFO, 优先级)执行。如果这个线程异常结束，会有另一个取代它，保证顺序执行。单工作线程最大的特点是可保证顺序地执行各个任务，并且在任意给定的时间不会有多个线程是活动的

**4 newScheduleThreadPool**

创建一个定长的线程池，而且支持定时的以及周期性的任务执行，支持定时及周期性任务执行

**1 线程是用什么参数停止的**

1 停止一个线程可以用Thread.stop()方法

2使用interrupt方法中断线程停止不了的线程interrupt()方法的使用效果并不像for+break语句那样，马上就停止循环。调用interrupt方法是在当前线程中打了一个停止标志，并不是真的停止线程

**7说说怎么运行线程中的方法**

使该线程开始执行；Java 虚拟机调用该线程的 run 方法结果是两个线程并发地运行；当前线程（从调用返回给 start 方法）和另一个线程（执行其 run  
方法）

**8 线程安全你是什么的做的**

synchronized(任意的对象（锁） )

{

写要被同步的代码

**9 Synchronized和volatile区别**

1 Volatile是变量修饰符 synchronized是修饰类的方法代码段

**10 并发三要素**

1 原子性（指的是一个或者多个操作一致性）

**2 可见性**

可见性指多个线程操作一个共享变量其中一个线程对变量进行修改

**3 有序性**

先后顺序执行

**线程五种基本状态**

1 新建状态（New）Thread t = new MyThead();

2 就绪状态（Runnable）用start方法

3 运行状态（Running）

4 阻塞状态（Blocked）

5 死亡状态 （Dead）退出run()

**10 Java中有几种类型的流**

**InputStream和outputStream继承inputStramReader和StreamWriter**

**11 介绍Collection框架和实现什么接口**

结构list Set queue的父级接口

Set接口有两个常用的实现类：HashSet和TreeSet List接口的常用接口有Arraylist和Vector

Map 接口有两个常用的实现类 Hashtable和HashMap

**1 Comparable**

**2 Comparator**

**12 list和map 的区别**

list 是存储单列数据集合 Map存储的是key value

**13 list map set 三个接口的存储元素时候各有什么特点**

List 和set有一个共同的接口的是父接口叫Collection set里不容许有相同元素

List 表示有先后顺序的集合 注意 不是按年龄调用add（）

Map 是双列集合 一般用put key value 结构来存储数据

**14 ArrayList与Linkedlist的区别是什么**

1. 二者实现结构不同arraylist是基于数组，linkedlist是基于链表，他们的特性也是由其数据结构决定的

2、随机遍历访问时linkedlist的性能要低于arraylist

3、 Arrylist的初始话时默认10的容量二而linkedlist默认初始化为空

4、linkedlist的增删要优于arraylist

**15 、arraylist底层实现原理**

基于数组实现【object】

**16、 equals和==的区别**

Int a =2；

Int b=2;

System.out.println（a==b）输出true 比较栈内存的值是否相等

Equals用来比较两个对象地址是否相等 所有类都是继承object所以没办法去比较唯一有效的是一个变量一个是对象

17、**面向对象的三个特征**

封装,继承,多态.这个应该是人人皆知.有时候也会加上抽象

18、**多态的好处**

**1**可替换性2可扩充性3接口性4简化性

**19 、int和Integer的区别?**

Integer是int的包装类型在拆箱和装箱二者自动转换

**20、String,StringBuffer和StringBuilder区别?**

String是字符串常量,final修饰;StringBuffer字符串变量(线程安全);StringBuilder 字符串变量(线程不安全).此外StringBuilder和StringBuffer实现原理一样,都是基于数组扩容来实现的

**21、了解泛型么?简述泛型的上界和下界?**

有时候希望传入的类型有一个指定的范围（自己想一些集合）

<?>: 无限制通配符

<? extends E>: extends 关键字声明了类型的上界,表示参数化的类型可能是所指定的类型,或者是此类型的子类

<? super E>: super关键字声明了类型的下界,表示参数化的类型可能是指定的类型,或者是此类型的父类

**22说说jvm垃圾回收实现原理**

1分代回收->内存划分不同世代->3个世代 年轻，年老，永久代->所有新生的对象都是优先存储年轻代的->大对象直接分配在老年代->年轻按照统计规律分成三个区->一个Eden区，两个Survivor区N次垃圾回收后仍然存活对象就会老年代所以他都是存储都是生命一些生命周期较长对象

**23 java当中的四种引用类型?他们之间的区别是什么?**

强引用:如果一个对象具有强引用,它就不会被垃圾回收器回收.即使当前内存空间不足,JVM也不会回收它.而是抛出 OutOfMemoryError 错误.使程序异常终止.如果想中断强引用和某个对象之间的关联.可以显式地将引用赋值为null,这样一来的话.JVM在合适的时间就会回收该对象.

软引用:在使用软引用时,如果内存的空间足够,软引用就能继续被使用而不会被垃圾回收器回收.只有在内存不足时,软引用才会被垃圾回收器回收

弱引用:具有弱引用的对象拥有的生命周期更短暂.因为当 JVM 进行垃圾回收,一旦发现弱引用对象,无论当前内存空间是否充足,都会将弱引用回收.不过由于垃圾回收器是一个优先级较低的线程,所以并不一定能迅速发现弱引用对象.

虚引用:如果一个对象仅持有虚引用,那么它相当于没有引用,在任何时候都可能被垃圾回收器回收.

**24 FutureTask是什么**

FutureTask表示一个异步运算的任务。FutureTask里面可以传入一个Callable的具体实现类，可以对这个异步运算的任务的结果进行等待获取、判断是否已经完成、取消任务等操作。当然，由于FutureTask也是Runnable接口的实现类，所以FutureTask也可以放入线程池中。

**2 Mysql面试题**

**1 说说mysql怎么调优**

1 Sql语句优化索引的优化

2 数据库表结构的优化

3 系统配置的优化

4 硬件优化

**2 数据库三大方式**

第一方式：每个咧都不可在拆分

第二方式：在第一范式的基础上，非主键列完全依赖于主键，而不能是依赖于主键的一部

第三方式：在第二方式的基础，非主键列只依赖于主键不依赖于其他非主键

**3你是如何理解三范式**

1 第一方式：1NF是对属性的原子性不可再分解

2 第二方式：2nf是对记录唯一性约束要求记录有唯一标识既实体的唯一性

3 第三方式：3NF是对字段用于约束

**4 数据库里有几种引擎**

****1 I****nnodb引擎：Innodb引擎提供了对数据库ACID事务的支持

2 MyIASM引擎(原本Mysql的默认引擎)：不提供事务的支持，也不支持行级锁和外键

3 MEMORY引擎：所有的数据都在内存中，数据的处理速度快，但是安全性不高

**5什么是脏读？幻读？不可重复读？**

1 脏读(Drity Read)：某个事务已更新一份数据，另一个事务在此时读取了同一份数据，由于某些原因，前一个RollBack了操作，则后一个事务所读取的数据就会是不正确的

2 不可重复读(Non-repeatable read):在一个事务的两次查询之中数据不一致，这可能是两次查询过程中间插入了一个事务更新的原有的数据

3 幻读(Phantom Read):在一个事务的两次查询中数据笔数不一致，例如有一个事务查询了几列(Row)数据，而另一个事务却在此时插入了新的几列数据，先前的事务在接下来的查询中，就会发现有几列数据是它先前所没有的。

**6 Sql标准定义四个隔离级别**

读取未提交 读取以提交 可重复读 可串行化

**7 怎么让索引失效**

**1前导模糊查询不能利用索引（）**

2 组合是组合索引的话

3 条件中有or

4.索引无法存储null值，所以where的判断条件如果对字段进行了null值判断，将导致数据库放弃索引而进行全表查询

**3 Redis面试题**

**1 Redis 中的数据类型有哪些**

有5种 数据结构分别是 string list set hash zset

**2 什么时候用redis**

1 不需要实时更新数据

2 需要实时更新但是更新频率不高的数据

3 谋个访问量极大而且更新很频繁

**3 为什么使用缓存**

**数据多的时候且很少变动的时候才使用加强系统效率**

**4 spring的面试题**

**1说说你对IOC理解**

控制反转是一种通过描述XML并且通过第三方式取生产获取特定的对象方式在spring中实现控制反转是IOC容器 其实现方法是依赖注入 通过IOC 管理POJO对象他们之间耦合关系

是一种设计思想就是 将原本在程序中手动创建控制权 交由Springle框架来管理IOC 就像一个是工厂一样 当我们 需要创建一个对象的时候只需要配置文件注解即可完全不用考虑是如何被创建出来

Spring IOC 的实现原理就是工厂模式 实现松耦合

**2 说说Spring中的设计模式？**

工厂模式：单例模式：代理模式：

**3 谈谈你对AOP理解**

AOP是OOP的延续是软件开发中的一个热点也是Spring框架中的一个重要内容是函数编程的一种衍生泛型

利用AOP可以对业务逻辑的各个部分进行隔离从而使得业务了逻辑各部分之间的耦合度降低提高高程序的可重用性同时提高开发的效率从不原有结构下增强功能

通过AOP以动态非侵入方式增强服务

4 AOP有哪些注解

前置通知 @Before 后置通知@After@返回通知@AfterReturning@异常通知@afterThowing@Around

@EnableAspectJAutoProxy开启基于注解的aop模式

**5 Springboot面试题**