Java面试题目

1. 分别说明下空间复杂度和时间复杂度？

答：

时间复杂度：时间复杂度并不是指程序具体运行的时间，而是算法执行语句的次数。通过时间复杂度，可以判断哪一个算法在具体执行时花费的时间更少。

常见的时间复杂度有：

常数阶O(1),

对数阶O(log2 n),

线性阶O(n),

线性对数阶O(n log2 n),

平方阶O(n^2)，

立方阶O(n^3)

k次方阶O(n^K),

指数阶O(2^n)。

随着n的不断增大，时间复杂度不断增大，算法花费时间越多。

空间复杂度：空间复杂度是对一个算法在运行过程中临时占用存储空间大小的量度。

1. 常见的几种排序算法？

答：

1. 冒泡排序：比较相邻的两个元素，如果第一个比第二个大，就交换它们两个，每次对比完，最后的一个元素是最大的。
2. 选择排序：每次都在未排序序列中找到最小（大）元素存放到起始位置。
3. 插入排序：从第一个元素开始，该元素可以任务已经被排序，取出下一个元素，在已经排序的元素序列中从后向前扫描，如果前面已经排序元素大于新元素，将该元素移到下一位置。
4. 希尔排序：先将整个待排序的记录序列分割成为若干子序列分别进行直接插入排序。
5. 归并排序：将序列分成若干个子序列，先使每个子序列有序，再使子系列断间有序。
6. 快速排序：从数列中挑出一个元素，称为“基准”（pivot）,重新排序数列，所有元素比基准小的摆放在基准前面，所有元素比基准值大的排在基准后面，最后递归把两个子序列排序了。
7. 单链表结构如何找到倒数第三个元素？

答：

1. 遍历链表，得到链表的长度n,将倒数的k的序号到顺序排号（n-k+1）,遍历链表，直到找到（n-k+1）个元素。
2. 从第一个元素开始，遍历k个元素，判断是否为空，若为空，则找到第k个元素。（不懂）
3. 设置两个指针，第二个比第一个先往后移动k步（因为包含头结点）。则这两个指针之间的距离刚好为k。那么，当第二个指针指向NULL的时候，第一个指针就刚好指在倒数第k个位置。只需一次遍历。
4. 乐观锁和悲观锁分别是什么？

答：

悲观锁表示在整个数据的处理过程中，将数据处于锁定的状态，悲观锁的实现依靠数据库提供的锁机制，悲观锁分为共享锁和排他锁，共享锁多个事物对于同一个数据可以共享一把锁，只能读不能修改，排他锁，只能由一个事物获取，其他事物就不能获取改行的其他锁，获取排他锁的事物可以对数据行读取和修改。

乐观锁：相对于悲观锁，在对数据库进行处理的时候，乐观锁并不会使用数据库提供的锁机制。一般的实现乐观锁的方式就是记录数据版本。

1. mysql常见的sql优化方式有哪些？
2. 选取最适用的字段属性
3. 使用连接（join）来代替子查询
4. 少用模糊查询，少用比较运算符，少用in，多用精确查询
5. 创建表时字段尽可能设置为not null
6. 使用索引
7. oracle常见的sql优化方式有哪些？

答：

一：sql语句尽量用大写的

因为oracle总是先解析SQL语句，把小写的字母转换成大写的再执行。

二、使用表的别名

当在SQL语句中连接多个表时, 尽量使用表的别名并把别名前缀于每个列上。这样一来,

就可以减少解析的时间并减少那些由列歧义引起的语法错误。

三、WHERE子句中的连接顺序

ORACLE采用自下而上的顺序解析WHERE子句,根据这个原理,表之间的连接必须写在其他

WHERE条件之前, 那些可以过滤掉最大数量记录的条件必须写在WHERE子句的末尾。

1. select子句中避免使用\*

ORACLE在解析的过程中, 会将'\*' 依次转换成所有的列名, 这个工作是通过查询数据字

典完成的, 这意味着将耗费更多的时间。

1. 避免使用耗费资源的操作

带有DISTINCT、UNION、MINUS、INTERSECT、ORDER BY的SQL语句会启动SQL引擎执行

耗费资源的排序(SORT)功能。DISTINCT需要一次排序操作,而其他的至少需要执行两次排序。

通常,带有UNION、MINUS、INTERSECT的SQL语句都可以用其他方式重写。

1. 优化group by

提高GROUP BY 语句的效率,可以通过将不需要的记录在GROUP BY之前过滤掉。

下面两个查询返回相同结果但明显第二个效率更高。

低效:

SELECT JOB,AVG(AGE) FROM TEMP

GROUP BY JOB HAVING JOB = 'STUDENT' OR JOB = 'MANAGER';

高效:

SELECT JOB,AVG(AGE) FROM EMP

WHERE JOB = 'STUDENT' OR JOB = 'MANAGER' GROUP BY JOB;

七、用TRUNCATE替代DELETE：

当删除表中的记录时,在通常情况下, 回滚段(rollback segments)用来存放可以被恢复的信息。如果你没有COMMIT事务,ORACLE会将数据恢复到删除之前的状态(准确地说是恢复到执

行删除命令之前的状况)。而当运用TRUNCATE时,回滚段不再存放任何可被恢复的信息。当命令运行后,数据不能被恢复。因此很少的资源被调用,执行时间也会很短。

注:TRUNCATE只在删除全表适用,TRUNCATE是DDL不是DML

八、使用DECODE函数来减少处理时间：

使用DECODE函数可以避免重复扫描相同记录或重复连接相同的表。

九、用EXISTS替代IN、用NOT EXISTS替代 NOT IN：

在基于基础表的查询中经常需要对另一个表进行联接。在这种情况下, 使用EXISTS(或NOT

EXISTS)通常将提高查询的效率。在子查询中,NOT IN子句将执行一个内部的排序和合并。无

论在哪种情况下,NOT IN都是最低效的(要对子查询中的表执行了一个全表遍历)。所以尽量将

NOT IN改写成外连接(Outer Joins)或NOT EXISTS。

（高效）

SELECT A.\* FROM TEMP(基础表) A WHERE AGE > 0

AND EXISTS(SELECT 1 FROM TEMP1 WHERE A.ID= ID AND NAME='TOM');

(低效)

SELECT A.\* FROM TEMP(基础表) A WHERE AGE > 0

AND A.ID IN(SELECT ID FROM TEMP1 WHERE NAME ='TOM');

1. 用索引提高效率
2. 用>=替代>：

高效:

SELECT \* FROM TEMP WHERE ID >=4;

低效:

SELECT \* FROM TEMP WHERE ID >3;

十二、避免在索引列上使用计算：

WHERE子句中，如果索引列是函数的一部分，优化器将不使用索引而使用全表扫描。

低效：

SELECT … FROM TEMP WHERE SAL \* 12 > 25000;

高效:

SELECT … FROM TEMP WHERE SAL > 25000/12;

十三：用UNION替换OR(适用于索引列):

用UNION替换WHERE子句中的OR将会起到较好的效果。对索引列使用OR将造成全表扫描。注意,以上规则只针对多个索引列有效。如果有column没有被索引, 查询效率可能会因为你没有选择OR而降低。

高效:

SELECT \* FROM USER\_TAB1 WHERE USER\_ID = 10

UNION

SELECT \* FROM USER\_TAB1 WHERE AGE = 20;

低效:

SELECT \* FROM USER\_TAB1 WHERE USER\_ID = 10 OR AGE = 20;

十四：避免在索引列上使用IS NULL和IS NOT NULL

避免在索引中使用任何可以为空的列，ORACLE将无法使用该索引。对于单列索引，如果列包含空值，索引中将不存在此记录。对于复合索引，如果每个列都为空，索引中同样不存在此记录。如果至少有一个列不为空，则记录存在于索引中。

如果唯一性索引建立在表的A列和B列上, 并且表中存在一条记录的A,B值为(123,null),ORACLE将不接受下一条具有相同A,B值（123,null）的记录(插入)。然而如果所有的索引列都为空，ORACLE将认为整个键值为空而空不等于空。因此你可以插入10000条具有相同键值的记录,当然它们都是空! 因为空值不存在于索引列中,所以WHERE子句中对索引列进行空值比较将使ORACLE停用该索引。

低效: (索引失效)

SELECT … FROM DEPARTMENT WHERE DEPT\_CODE IS NOT NULL;

高效: (索引有效)

SELECT … FROM DEPARTMENT WHERE DEPT\_CODE >=0;

十五、某些SELECT 语句中的WHERE子句不使用索引。

(1)'!='不走索引。索引只能告诉我们什么存在于表中, 而不能告诉你什么不在表中。

(2)'||'是字符连接函数。就象其他函数那样, 停用了索引。

(3)'+'是数学函数。和其他数学函数一样, 停用了索引。

(4)相同的索引列不能互相比较,这将会启用全表扫描。

7、java中的空接口有什么作用？

答：空接口的存在是为了起一种标识作用。

8、mybatis模糊查询有几种方式？

答：

1. 使用${}

question like ‘%${question}%’

注意：由于$是参数直接注入的，导致这种写法，大括号里面不能注明jdbcType，不然会报错

1. 使用#{}

question like “%”#{question,jdbcType=VARCHAR}”%”

注意：因为#{...}解析成sql语句时候，会在变量外侧自动加单引号' '，所以这里 % 需要使用双引号" "，不能使用单引号 ' '，不然会查不到任何结果。

1. 使用concat()函数连接参数

answer like concat(‘%’,#{answer,jdbcType=VARCHAR},’%’)

1. 行锁和锁表有什么区别？

答：

行锁：

1. 支持的存储引擎：Innodb;
2. InnoDB行锁是通过给索引上的索引项加锁来实现的，意味着：只有通过索引条件检索数据，InnoDB才使用行级锁，否则，InnoDB将使用表锁；
3. 适用场景：有大量按索引条件并发更新少量不同数据，同时又有并发查询的应用；
4. 特点：开销大，加锁慢；会出现死锁；锁定粒度小，发生锁冲突的概率低，并发度高
5. 分析：show status like 'innodb\_row\_lock%';分析系统上行锁的争夺情况如果发现锁争用比较严重，如InnoDB\_row\_lock\_waits和InnoDB\_row\_lock\_time\_avg的值比较高，还可以通过设置InnoDB Monitors来进一步观察发生锁冲突的表、数据行等，并分析锁争用的原因。

表锁：

1. 支持的存储引擎：Innodb、MYIsam
2. 适用场景：以查询为主，只有少量按索引条件更新数据的应用
3. 特点：开销小，加锁快；不会出现死锁；锁定力度大，发生锁冲突概率高，并发度最低
4. git和svn有什么区别？

答：

1. GIT是分布式的，SVN不是。
2. GIT把内容按元数据方式存储，而SVN是按文件。
3. GIT分支和SVN的分支不同。
4. GIT没有一个全局的版本号，而SVN有。
5. GIT内容的完整性要优于SVN。
6. ant和maven有什么区别？

答：

Maven 是一个项目管理工具，可以对 Java 项目进行构建、依赖管理。

mavne

1. mave拥有约定，只要遵守约定，它就知道你的源代码在哪里。
2. Maven 是声明式的，你需要做的只是创建一个 pom.xml 文件然后将源代码放到默认的目录。
3. maven是有一个生命周期的。

ant

1、ant没有约定，项目生命周期，它是命令式的。

2、所有操作都要手动去创建、布置，甚至连build.xml文件都需要手动创建。

12、kafka和activemq有什么区别？

答：

Kafka优点：

1. kafka是一个分布式的mq, 具有高扩展，可以通过集群扩展。
2. Kafka是一个高性能，kafka的性能大大超过传统的activemq,kafka还支持batch操作。
3. 容错，Kafka每个Partition的数据都会复制到几台服务器上。当某个Broker故障失效时，ZooKeeper服务将通知生产者和消费者，生产者和消费者转而使用其它Broker。

Kafka缺点：

1、重复消息。Kafka 只保证每个消息至少会送达一次，虽然几率很小，但一条消息有可能会被送达多次。

2、消息乱序。虽然一个Partition 内部的消息是保证有序的，但是如果一个Topic 有多个Partition，Partition 之间的消息送达不保证有序。

3、复杂性。Kafka需要zookeeper 集群的支持，Topic通常需要人工来创建，部署和维护较一般消息队列成本更高

activemq

1. activemq是一个传统mq，完全支持JMS规范，
2. Java消息服务（Java Message Server） 及JMS，是一个Java平台中关于面向消息中间件的API，用于在两个应用程序之间或分布式系统中发送消息，进行异步通信。

13、java分布式系统定时任务，如何保证多台服务只执行一次？

答：

1. 通过数据库表处理。
2. 通过redis处理。
3. 服务器宕机，排查问题思路？

答：

一、

问题症状 ：服务器内存占用持续增长，性能低下，并发上不去，隔几天宕机

排查思路 ：分析出内存泄露模块，分析出性能瓶颈，调优JVM

使用工具 ：jconsole、jprofiler

问题一：HTTPCLIENT连接没有真正释放

问题二：JVM调优，开启JMX服务，使用JCONSOLE连接

问题三：内存泄露

问题四：连接数暴涨，分析发现磁盘已满

问题五：性能瓶颈，通过JPROFILER分析，找出消耗时间最多的方法，进行优化

问题六：数据库连接达到上限

问题七：其他问题

另外一种回答：

a、是否是应用程序导致内存溢出或者泄露，out of memory导致

b、是否是进程过多或者不断创建，耗尽资源导致

c、是否是数据库程序死锁，连接数过多导致

d、是否是应用程序异常导致

e、是否是流量负载过大导致

f、是否是遭受黑客入侵攻击导致

g、是否是误操作导致

15、数据库char、varchar、varchar2、nvarchar、nvarchar2区别是什么？

答：

char --的长度是为一个字节，表达的数值范围是0~255，char的长度是固定的,默认存储单位是字节。

varchar --存放定长的字符数据，最长是2000个字符,默认存储单位是字节。

varchar2 -- 存放可变长字符数据，最大长度为4000字符，默认存储单位是字节（oracle特有的）。

nvarchar -- 默认存储单位是字符。

nvarchar2 --默认存储单位是字符（oracle特有的）。

varchar(char 5) --默认存储单位是字符。

varchar2(char 5) --弄人存储单位是字符（oracle特有的）。

16、怎么解决kafka幂等性问题？

答：

Kafka为了实现幂等性，它在底层设计架构中引入了ProducerID和SequenceNumber。那这两个概念的用途是什么呢？

ProducerID：在每个新的Producer初始化时，会被分配一个唯一的ProducerID，这个ProducerID对客户端使用者是不可见的。

SequenceNumber：对于每个ProducerID，Producer发送数据的每个Topic和Partition都对应一个从0开始单调递增的SequenceNumber值。

17、1000个数据中如何找到2个相同的元素？

答：先排序，在使用二分查找算法实现。

1. switch可以使用的数据类型？

答：switchs使用的数据类型会处理成int类型再做比较。

可以使用的数据类型包括

基本数据类型：byte, short, char, int

包装数据类型：Byte, Short, Character, Integer

枚举类型：Enum

字符串类型：String（Jdk 7+ 开始支持）

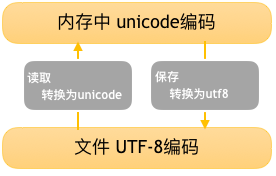
1. utf-8、gbk、unicode分别有什么区别？

答：utf-8、gbk是适用于中文的，unicode是国际标准组织统一标准字符集。

我们总结一下现在计算机系统通用的字符编码工作方式：

在计算机内存中，统一使用Unicode编码，当需要保存到硬盘或者需要传输的时候，就转换为UTF-8编码。

用记事本编辑的时候，从文件读取的UTF-8字符被转换为Unicode字符到内存里，编辑完成后，保存的时候再把Unicode转换为UTF-8保存到文件。如下图：



1. hashcode有什么作用？

答：hashcode主要是使用快速查找的。

1. 重载和重写的定义是什么？

答：

重载：在同一个类或者接口中，不同函数使用相同的函数名，但是函数的参数列表不同，参数列表不同指的是参数个数不一样或者参数类型不一样或者参数顺序不一样，和返回值类型不一样。

重写：也叫重写是指派生类对基类中的虚函数重新实现。

规则是函数名一样，参数列表一样，返回值类型是基类或者基类的子类，访问权限修饰符必须等于大于基类，抛出的异常必须是基类抛出异常的本类或者子类。

1. 什么是虚函数？

答：在某基类中声明为virtual并在一个或多个派生类中被重新定义的成员函数。

作用：实现多态性，通过指向派生类的基类指针或引用，访问派生类中同名覆盖成员函数。

1. 数组转集合和集合转数组实现方式？

答：

数组转集合

1. 可以使用遍历的方式
2. 可以使用Arrays工具类的asList方法

注意：

*/\*\* 当数组存储的是基本数据类型时，则是把该数组对象当作元素存进集合 \*/*int[] array1 = {1,2};  
List<int[]> list1 = Arrays.*asList*(array1);  
System.*out*.println(list1);  
  
*/\*\* 当数组存储的是基本类型时，则是把该数组每个元素放进集合中 \*/*Integer[] array2 = {1,2};  
List<Integer> list2 = Arrays.*asList*(array2);  
System.*out*.println(list2);

该方法返回的数组的固定大小的列表，不能对转化后的集合进行增删，不然会报错。

Exception in thread "main" [1, 2]

java.lang.UnsupportedOperationException

at java.util.AbstractList.remove(AbstractList.java:161)

at test.cn.jun.ArraysTest.main(ArraysTest.java:22)

1. jdk8支持

*/\*\* 如果是jdk8可以是这种方式 \*/*try {  
 int[] array3 = {1,2};  
 List<Object> collect = Arrays.*stream*(array3).boxed().collect(Collectors.*toList*());  
 System.*out*.println(collect);  
}catch (Exception ex){  
 ex.printStackTrace();  
}

1. 可以使用apache commons-lang3 jar



1. 可以使用Collections集合类的addALll方法，数组存储对象是只能是引用类型。

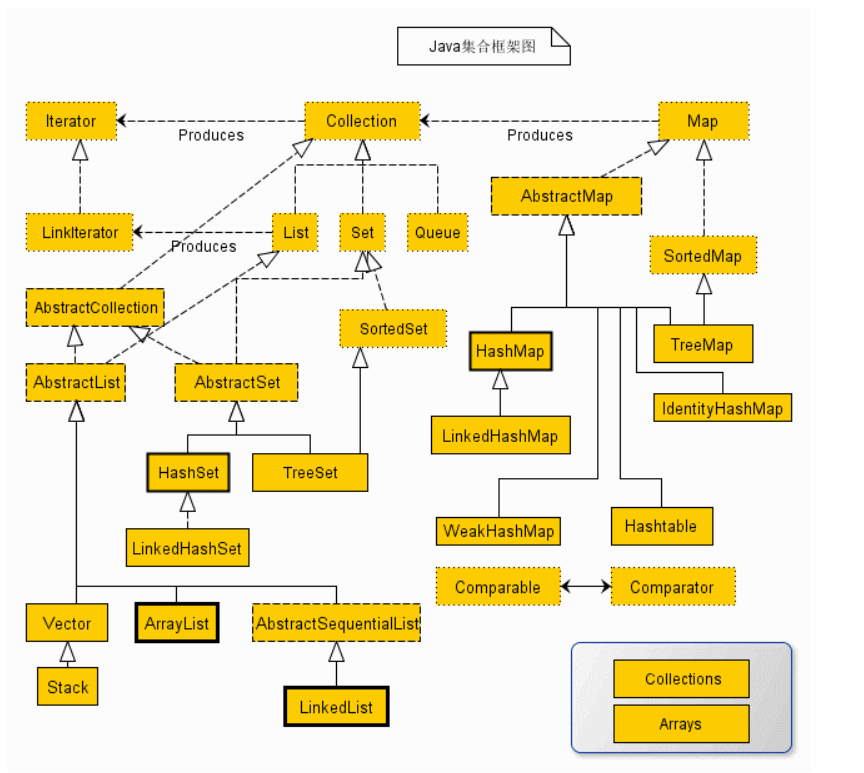
Integer[] array = {1,2,3};  
List<Integer> list = new ArrayList<Integer>();  
boolean b = Collections.*addAll*(list, array);  
System.*out*.println(list);  
list.add(1);  
System.*out*.println(list);

集合转数组

1. 可以使用遍历方式
2. 可以使用List的toArray方法

*/\*\* 集合转数组 \*/*ArrayList<String> arrayList1 = new ArrayList();  
arrayList1.add("1");  
arrayList1.add("2");  
String[] array = new String[arrayList1.size()];  
array = arrayList1.toArray(array);  
System.*out*.println(array);

1. 集合架构是怎么样的？



List和Set有什么区别?

1. List接口实例存储的是有序的，元素可以重复，Set接口实例存储的是无序的，不重复的数据。
2. Set检索效率低下，删除和插入效率高，插入和删除不会引起元素位置改变。
3. List可以动态增长，根据实际存储的数据的长度自动增长List的长度。查找元素效率高，插入删除效率低，因为会引起其他元素位置改变。

List中的ArrayList和LinkedList有什么区别？

1. ArrayList是实现了基于动态数组的数据结构，LinkedList是基于双向链表结构。
2. 对于随机访问的get和set方法，ArrayList要优于LinkedList,因为LinkedList要移动指针。
3. 对于新增和删除操作add和remove，LinkedList比较占优势，ArrayList要移动数据。

Set中HashSet、LinkedHashSet、TreeSet有什么特点？  
答：

hashSet有哈希表（实际上是一个hashmap的实例）支持，不保证hashSet的迭代顺序，允许使用null元素。

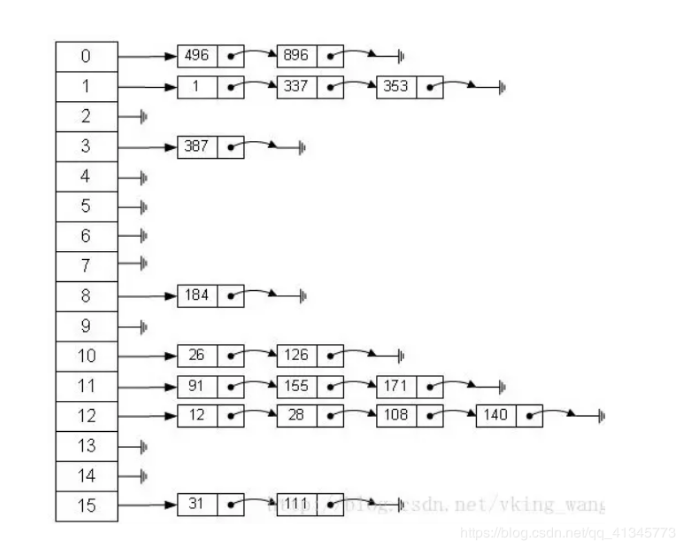
HashSet保存对象的唯一性，是通过equals和hashCode方法实现的，只有2个方法成立时，才能判断是否唯一，要求存入的对象必须实现hashCode方法。

LinkedHashSet是基于哈希表和双向链表实现的，元素不可以重复，可以保证元素插入的数据。

TreeSet是基于二叉树实现的，TreeSet数据是自动排好序的，不允许存入null。

Map中hashMap和hashTabLe区别？

HashMap是基于哈希表实现的，每个键值对组成了一个Entry实体，Entry类实际上是一个单向的链表结构。



从上图我们可以发现数据结构由数组+由单向链表组成，一个长度为16的数组中，每个元素存储的是一个链表的头结点。那么这些元素是按照什么样的规则存储到数组中呢。一般情况是通过hash(key.hashCode())%len获得，也就是元素的key的哈希值对数组长度取模得到。比如上述哈希表中，12%16=12,28%16=12,108%16=12,140%16=12。所以12、28、108以及140都存储在数组下标为12的位置。

Hashtable是基于散列表实现的，由数组和单向列表组成。

1. hashmap不安全，hashtable安全,hashtable使用了同步Synchronized方法。
2. Hashmap可以存储null,hashtable不可以存储null。

线程安全的集合类有哪些？

答：

1. Hashtable
2. Vetcor
3. Stack
4. ConcurrentHashMap、CopyOnWriteArrayList、CopyOnWriteArraySet
5. ArrayList扩容机制是怎么样的？

答：

总的来说分为两步：

1、扩容：把原来的数组复制到另一个内存空间更大的数组中

2、添加元素：把新元素添加到扩容以后的数组中

初始化方式有带参的和无带参的

无带参的默认为10，有带参的就使用带参的。

扩容因子为1.5，也就是存储到11个数的时候，就会扩容到15。

1. Spring的aop面向切面编程除了通过动态实现还有什么方式可以实现？

答：除了动态代理，还可以通过CGLIB代理实现，动态代理只能代理一个接口，CGLIB可以代理接口或则类。

1. 反射的应用是怎么样的？

答：

类的加载：系统会通过加载、连接、初始化三步来对一个类的初始化。

1. 加载：

把class文件读入内存中，并为之创建一个Class对象

任何类被使用的时，系统都会创建一个Class对象

1. 连接

验证 是否有正确的内部结构，并和其他内协调一致

准备 为类的静态成员分配内存，并设置默认初始化值

解析 将类的二进制数据中的符号引用替换为直接引用

1. 初始化

为类中的成员变量和成员方法分配内存地址，并为之赋值

什么是反射：反射就是在运行的时候通过Class对象，获取类中的构造方法Constructor、方法Method、属性Field对象，并对它们进行操作。

应用

1.各种框架的设计

2.IDE等软件系统的开发

3.通过获得Class对象的三种方法

1.直接通过静态的属性.class得到;例如：Date.class

2.通过对象的方法得到，getClass();在Object中有这个方法

3.通过类的全限定名得到，调用Class.forName("完全限定类");会有ClassNotFoundException 异常

4.三个方法得到的Class对象是一样的

主要有Class、Constructor、Method、Field类

1. String转字符数组需要注意什么？

答：getBytes需要指定编码格式，比如说”中”.getBytes(“utf-8”);

”中”.getBytes(“gbk”); ”中”.getBytes(“iso8859-1”);

当系统的请求头指定要求是iso8859-1时，可以使用下面方式转化

String s = new String("中".getBytes("UTF-8"), "ISO8859-1");  
System.*out*.println(s);  
  
String s1 = new String(s.getBytes("ISO8859-1"), "UTF-8");  
System.*out*.println(s1);

1. Utf-8、gbk、iso8859-1有什么区别？

答：utf-8、gbk可以存储中文字符，utf-8一个中文有3个bytes，gbk一个中文有2个bytes，iso8859-1不可以存储中文字符。

1. Redis和数据库同步问题？

答：

读操作

目前的读操作有个固定的套路，如下：

客户端请求服务器的时候，发现如果服务器的缓存中存在，则直接取服务器的；

如果缓存中不存在，则去请求数据库，并且将数据库计算出来的数据回填给缓存；

返回数据给客户端；

写操作

各种情况会导致数据库和缓存出现不一致的情况，这就是缓存和数据库的双写一致性问题；

目前缓存存在三种策略，分别是

Cache Aside 更新策略：同时更新缓存和数据库；

Read/Write Through 更新策略：先更新缓存，缓存负责同步更新数据库；

Write Behind Caching 更新策略：先更新缓存，缓存定时异步更新数据库；

Redis有哪几种数据类型？

开多个线程时，怎么判断所有线程执行完毕？

当线程池中的线程等于总数时，还要线程需要处理，这时要怎么办？

Objec中和线程相关的方法有哪些？  
运行时异常有哪些？

检查异常有哪些？

异常的结构框架是怎么样的？

String、StringBuffer、StringBuilder的区别？

Linux命令中查看线程的命令是什么？

Linux命令中启动jar包的命令是什么？  
jdk有几种类型？

有没有关注jdk版本的新特性？

Activemq和kafka的介绍？

Maven的介绍？

Maven私服的配置？

String类？

Integer类？

Equals和==？

对jvm的了解程度？

什么是线程池？

代码重构有哪些方法？

不停系统备份数据怎么处理？

二叉树数据结构?