**core java和java什么区别？**

core java是JAVA核心知识，就是JAVA最基础的部分。

JAVA泛指整个JAVA平台和技术，包括各种扩展和第三方的技术

**面向对象**

**1、面向对象的特征有哪些方面？**

答：面向对象的编程语言有封装、继承 、多态等3个主要的特征。

**封装：**面向对象的封装就是把描述一个对象的属性和行为的代码封装在一个类中

**继承：** 在定义和实现一个类的时候，可以在一个已经存在的类的基础之上来进行，把这个已经存在的类所定义的内容作为自己的内容

继承是子类自动共享父类数据和方法的机制，这是类之间的一种关系，提高了软件的可 重用性和可扩展性。

**多态：**比喻：人吃饭，你看到的是左手，还是右手

向上造型

**2、java 中实现多态的机制是什么？**

靠的是父类或接口定义的引用变量可以指向子类或具体实现类的实例对象，

而程序调用的方法在运行期才动态绑定，就是引用变量所指向的具体实例对象的方法，

也就是内存里正在运行的那个对象的方法，而不是引用变量的类型中定义的方法。

**4.4 StringBuilder 类**

与 String 对象不同，StringBuilder 封装“可变”的字符串，有线程安全问题。

对象创建后，可通过调用方法改变其封装的字符序列。要点toString.

**1** **抽象类和接口的区别?**

**抽象类：**

1）抽象方法，只有行为的概念，没有具体的行为实现。使用abstract关键字修饰，没有方法体。子类必须重写这些抽象方法。

2）包含抽象方法的类，一定是抽象类。

3）抽象类只能被继承，一个类只能继承一个抽象类,单一继承。

**接口：**

1）全部的方法都是抽象方法，属型都是常量

2）不能实例化，可以定义变量。

3）接口变量可以引用具体实现类的实例

4）接口只能被实现，一个具体类实现接口，必须实现全部的抽象方法

5）接口之间可以多实现

6）一个具体类可以实现多个接口，实现多继承现象

**下面比较一下两者的语法区别：**

1.抽象类可以有构造方法，接口中不能有构造方法。

2.抽象类中可以有普通成员变量，接口中没有普通成员变量

3. 一个类可以实现多个接口，但只能继承一个抽象类

4.抽象类中可以包含非抽象的普通方法，接口中的所有方法必须都是抽象的，不能有非抽象的普通方法。

5.抽象类中的抽象方法的访问类型可以是 public，protected和（默认类型,虽然 eclipse 报错，但应该也不行），

但接口中的抽象方法只能是 public 类型的，并且默认即为 public abstract类型。

6. 抽象类中可以包含静态方法，接口中不能包含静态方法

7. 抽象类和接口中都可以包含静态成员变量，抽象类中的静态成员变量的访问类型可以任意， 但接口 中定义的变量只能是 publicstaticfinal 类型，并且默认即为 publicstaticfinal类型。

2、启动一个线程是用run()还是start()?

启动一个线程是调用start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由JVM调度并执行。这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

**3、****线程的生命周期？**

**1.线程的生命周期**  
线程是一个动态执行的过程，它也有一个从产生到死亡的过程。

(1)生命周期的五种状态

**新建（new Thread）**  
当创建Thread类的一个实例（对象）时，此线程进入新建状态（未被启动）。  
例如：Thread  t1=new Thread();

**就绪（runnable）**  
线程已经被启动，正在等待被分配给CPU时间片，也就是说此时线程正在就绪队列中排队等候得到CPU资源。例如：**t1.start();**

**运行（running）**  
线程获得CPU资源正在执行任务（run()方法），此时除非此线程自动放弃CPU资源或者有优先级更高的线程进入，线程将一直运行到结束。

**死亡（dead）**  
当线程执行完毕或被其它线程杀死，线程就进入死亡状态，这时线程不可能再进入就绪状态等待执行。

自然终止：正常运行run()方法后终止

异常终止：调用**stop()**方法让一个线程终止运行

**堵塞（blocked）**  
由于某种原因导致正在运行的线程让出CPU并暂停自己的执行，即进入堵塞状态。

正在睡眠：用sleep(long t) 方法可使线程进入睡眠方式。一个睡眠着的线程在指定的时间过去可进入就绪状态。

正在等待：调用wait()方法。（调用motify()方法回到就绪状态）

被另一个线程所阻塞：调用suspend()方法。（调用resume()方法恢复）

**4、线程的基本概念、线程的基本状态以及状态之间的关系**

线程指在程序执行过程中，能够执行程序代码的一个执行单位，每个程序至少都有一个线程，也就是程序本身。Java中的线程有四种状态分别是：运行、就绪、挂起、结束。

**5、 列举你所知道的线程同步的方法：**

1）wait():让线程等待。将线程存储到一个线程池中。

2）sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉InterruptedException异常

3）notify()：唤醒被等待的线程。通常都唤醒线程池中的第一个。让被唤醒的线程处于临时阻塞状态。

4）notifyAll(): 唤醒所有的等待线程。将线程池中的所有线程都唤醒。

**5、多线程有几种实现方法?同步有几种实现方法?**

1）多线程有两种实现方法，分别是继承Thread(**线程)**类与实现Runnable(**可运行**)接口

2）同步的实现方面有两种，分别是synchronized（同步）,wait（等待V）与notify （唤醒）

wait():使一个线程处于等待状态，并且释放所持有的对象的lock（锁）。

sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉InterruptedException异常。

notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，

而是由JVM确定唤醒哪个线程，而且不是按优先级。

Allnotify():唤醒所有处入等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

**12）线程并发安全问题：synchronized 关键字，线程安全锁、同步监视器。**

多线程在访问同一个数据时（写操作），可能会引发不安全操作。

①哪个线程报错不捕获，则线程死，不影响主程序。

②同步：同一时刻只能有一个执行，A 和 B 配合工作，步调一致的处理（B 得到 A

的执行结果才能继续）。如一群人上公交车。

异步：同一时刻能有多个执行，并发，各自干各自的。如一群人上卡车。

**3 、简述你对java反射机制的理解**

JAVA反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；

对于任意一个对象，都能够调用它的任意一个方法和属性

**4、构造器Constructor是否可被override（重写）?**

答：构造器不能被继承，因此也不能重写，但可以被重载。

**5、Overload(重载)和Override(重写)的区别？**

两者都是多态的表现

答：**Overload(重载)**，方法名相同，但参数类型，个数，顺序不同为重载。（是同一类中方法之间的区别）

**Override(重写)**，子类重写父类，子类的方法名与父类名一致，参数个数，

类型都与父类完全一样，为方法的重写。（是类与类，父子类中之间的区别）

重写遵循一同两小一大原则: 两同：1)方法名

2)形参列表

两小：1)返回值类型比父类更小或相等

2)异常比父类方法更小或相等

一大：子类权限比父类大或相等

**6、 java 中实现多态的机制是什么?**

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。

重写 Overriding 是父类与子类之间多态性的一种表现，重载 Overloading 是一个类中多态性的一种表现。

**7、"=="和 equals 方法究竟有什么区别？**

答：**==操作符**: 专门用来比较两个变量的值是否相等，也就是用于比较变量所对应的内存中所存储的数值是否相同，要比较两个基本类型的数据或两个引用变量是否相等，只能用==操作符。

**equals**: 方法是用于比较两个引用对象的内容是否相同，就好比去比较两个人的长相是否相同，它比较的两个对象是独立的。

**8、静态变量和实例变量的区别？**

答：在语法定义上的区别：静态变量前要加 static 关键字，类值接调用

而实例变量前则不加。

**9、说出一些常用的类，包，接口，请各举5个**

要让人家感觉你对java ee开发很熟，所以，不能仅仅只列core java中的那些东西，

要多列你在做ssh项目中涉及的那些东西。就写你最近写的那些程序中涉及的那些类。

常用的类：BufferedReader BufferedWriter FileReader FileWirter String Integer

java.util.Date，System，Class，List,HashMap

常用的包：java.lang java.io java.util java.sql java.awp, javax.servlet, org.apache.strtuts.action, org.hibernate

常用的接口：Remote List Map Document NodeList,Servlet,HttpServletRequest,HttpServletResponse,Transaction(Hibernate)、Session(Hibernate),HttpSession

**10、 对象序列化**

将一个对象转换为字节形式的过程就是对象序列化。序列化还有个名称为串行化，序 化后的对象再被反序列化后得到的对象，

与之前的对象不再是同一个对象。

1）对象序列化必须实现 Serializable 接口，但该接口无任何抽象方法，不需要重写方法， 只为了标注该类可序列化。

2）且同时建议最好添加版本号（编号随便写）：serialVersionUID。版本号，

用于匹配当 前类 与其被反序 列化的对象是否处于同样的特征（ 属性列表一 致等 ）。

反序列化时 ， ObjectInputStream 会根据被反序列化对象的版本与当前版本进行匹配，

来决定是否反序列化。 不加版本号可以，但是可能存在反序列化失败的风险。

**3）JDK 提供的大多数 java bean 都实现了该接口**

**4）transient 关键字：序列化时忽略被它修饰的属性。**

**5）对象的序列化使用的类：ObjectOutputStream**

writeObject(Object obj)：①将给定对象序列化。②然后写出。

**6）对象的反序列化使用的类：ObjectInputStream**

Object readObject()：将读取的字节序列还原为对象

**7）对于 HTTP 协议：通信一次后，必须断开连接，想再次通信要再次连接。**

----------------------------------------------------------------------------------------

**集合**

----------------------------------------------------------------------------------------

**1、List 和 Map 区别?**

一个是存储单列数据的集合，另一个是存储键和值这样的双列数据的集合。

List 中存储的数据是有顺序，并且允许重复。

Map 中存储的数据是没有顺序的，其键是不能重复的，它的值是可以有重复的,key,value,。

**2.、List, Set, Map 是否继承自 Collection接口?**

List，Set 是，Map 不是

**3、List、Map、Set三个接口，存取元素时，各有什么特点？**

List 以特定次序来持有元素，可有重复元素。

Set无法拥有重复元素,内部排序。

Map 保存 key-value 值，value 可多值。

**4、Hash Map和Hash Table有何不同？**

（1）Hash Map允许key和value为null，而Hash Table不允许。

（2）Hash Table是同步的，而Hash Map不是。所以Hash Map适合单线程环境，Hash Table适合多线程环境。

**5、Collection 和 Collections的区别。**

Collection 是集合类的上级接口，继承与他的接口主要有 Set和 List.

Collections是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、

排序、线程安全化等操作。

**7、Array和ArrayList有何区别？什么时候更适合用Array？**

Array可以容纳基本类型和对象，而ArrayList只能容纳对象。

Array是指定大小的，而ArrayList大小是固定的。

Array没有提供ArrayList那么多功能，比如addAll、removeAll和iterator等。尽管ArrayList明显是更好的选择，但也有些时候Array比较好用。

（1）如果列表的大小已经指定，大部分情况下是存储和遍历它们。

（2）对于遍历基本数据类型，尽管Collections使用自动装箱来减轻编码任务，在指定大小的基本类型的列表上工作也会变得很慢。

（3）如果你要使用多维数组，使用[][]比List<List<>>更容易。

**8、哪些集合类是线程安全的？**

Vector、HashTable、Properties和Stack是同步类，所以它们是线程安全的，可以在多线程环境下使用。

Java1.5并发API包括一些集合类，允许迭代时修改，因为它们都工作在集合的克隆上，所以它们在多线程环境中是安全的。

**9、你所知道的集合类都有哪些？主要方法？**

最常用的集合类是 List 和 Map， List 的具体实现包括 ArrayList 和 Vector，它们是可变大小的列表， 比较适合构建、存储和操作任何类型对象的元素列表。

List 适用于按数值索引访问元素的情形。

Map 提供了一个更通用的元素存储方法。 Map 集合类用于存储元素对（称作"键"和"值"），其中每个键映射到一个值。

线程安全

**10、队列和栈是什么，列出它们的区别？**

栈和队列两者都被用来预存储数据。java.util.Queue是一个接口，它的实现类在Java并发包中。

队列允许先进先出（FIFO）检索元素，但并非总是这样。Deque接口允许从两端检索元素。

-------------------------------------------------------------------------------------

**3、请写出你最常见到的5个runtime exception （运行时异常）。**

NullPointerException（空指针异常）：当应用程序试图在需要对象的地方使用 null 时，抛出NullPointerException异常

ArrayIndexOutOfBoundsException（数组下标异常）、

ClassCastException（类转换异常）：当试图将对象强制转换为不是实例的子类时，抛出该异常（ClassCastException)

ArithmeticException（数学计算异常）：一个整数“除以零”时，抛出ArithmeticException异常。

SystemException（系统异常）

InterruptedException （中断异常）

---------------------------------------------------------

**JS**

---------------------------------------------------------

**1、 什么是 JavaScript 对象**

1）JavaScript 是一种基于对象的语言，对象是 JavaScript 中最重要的元素。

2）JavaScript 包含多种对象：内置对象、自定义对象、浏览器对象、HTML DOM 对象、

ActiveX 对象

**2、使用对象**

1）对象由属性和方法封装而成。

2）属性的引用：使用点“．”运算符、通过下标的方式引用。

3）对象的方法的引用：ObjectName.methods()。

**3、 常用内置对象**

1）简单数据对象：String、Number、Boolean

2）组合对象：Array、Date、Math

3）复杂对象：Function(函数)、RegExp

---------------------------------------------------------------------------------------

**数据库**

**Oracle和 Mysql 的区别？**

oracle一般用于大型的软件系统，mysql比较实用与小型的软件

oracle是要付费的，是强大的商业化数据库，安全性高，优点一大堆，交钱肯定弄好货了。 mysql开源，可以不付费，但是mysql不支持视图;不支持commit-rollback，有效处理commit-rollback将需要完全不同于mysql使用的表布局，mysql也将需要而外的线程在表上做自清工作，而且磁盘用量将更高，而不用它，mysql比几乎所有的其他SQL数据库都快2到4倍，所以小型开发用mysql是很好的

**1.数据库游标？**

（1）游标的概念

游标是指向查询结果集的一个指针，它是一个通过定义语句与一条Select语句相关联的一组SQL语句，即从结果集中逐一的读取一条记录。游标包含两方面的内容：

1. 游标结果集：执行其中的Select语句所得到的结果集；

2. 游标位置：一个指向游标结果集内的某一条记录的指针

利用游标可以单独操纵结果集中的每一行。游标在定义以后存在两种状态：关闭和打开。当游标关闭时，其查询结果集不存在；只有当游标打开时，才能按行读取或修改结果集中的数据。

**2.怎么去除重复数据？**

**使用伪列去重复**

**union**与**unionAll**的区别 他们之间的区别就在于对重复的数据的处理上，前者不输出重复的数据，后者则需要输出重复的数据。 在效率上后者更高，确定合并的结果没有重复的数据那么就用unionall。

**4.数据库的存储过程？**

sql语句执行的时候要先编译，以一个名称存储并作为以个单元处理，然后执行。存储过程就是编译好了的一些sql语句。

应用程序需要用的时候直接调用就可以了，所以执行起来比单个SQL语句块，所以效率会高。

**4、存储过程和函数的区别**

存储过程是用户定义的一系列sql语句的集合，涉及特定表或其它对象的任务，用户可以调用存储过程，而函数通常是数据库已定义的方法，它接收参数并返回某种类型的值并且不涉及特定用户表。

**5.数据库存储过程的优点！**

1.存储过程只在创建时进行编译，以后每次执行存储过程都不需要重新编译，而一般SQL语句每执行一次就编译一次，所以使用存储过程可提高数据库执行速度。

2.当对数据库进行复杂操作时，可将此复杂操作用存储过程封装起来与数据库提供的事务

处理结合一起使用。

3.存储过程可以重复使用，可减少数据库开发人员的工作量。

**8、在数据库中条件查询速度很慢的时候,如何优化?**1.建索引  
2.减少表之间的关联  
3.优化sql，尽量让sql很快定位数据，不要让sql做全表查询，应该走索引,把数据量大的表排在前面  
4.简化查询字段，没用的字段不要，已经对返回结果的控制，尽量返回少量数据

5.少用“\*”来查询语句，使用“\*”会查询整张表。

9、一般情况下，关系数据模型与对象模型之间有哪些匹配关系（A,B,C）

A)表对应类

B)记录对应对象

C)表的字段对应类的属性

D)表之间的参考关系对应类之间的依赖关系

**二、数据连接池的工作机制是什么？**

服务器程序启动时会建立一定数量的池连接，当客户端程序需要连接时会从里面取，

池驱动程序会标记该连接为忙，当连接池中的连接使用完了之后，池驱动程序会再建一批连接。关闭连接时，这时候关闭的不是原始连接，关闭的是代理的连接connection,而是把原始connection对象还回到连接池中。

**三、 数据库设计三大范式**

**1. 第一范式(确保每列保持原子性)**

第一范式是最基本的范式。如果数据库表中的所有字段值都是不可分解的原子值，就说明该数据库表满足了第一范式。

**2．第二范式(确保表中的每列都和主键相关)**

第二范式在第一范式的基础之上更进一层。第二范式需要确保数据库表中的每一列都和主键相关，

而不能只与主键的某一部分相关（主要针对联合主键而言）。

也就是说在一个数据库表中，一个表中只能保存一种数据，不可以把多种数据保存 在同一张数据库表中。

**3．第三范式(确保每列都和主键列直接相关,而不是间接相关)**

第三范式需要确保数据表中的每一列数据都和主键直接相关，而不能间接相关。

**1、子查询**

先执行子查询得到结果，再进行主查询

1、 分页查询

分页是将查询的结果集分批显示

目的是为了解决时间，性能，资源消耗，和用户需求。

当查询结果集条目数非常多时，通常会使用分页。

**2、什么是视图 ？**

视图是数据库对象

视图在SQL语句中的体现形式是一个表的样子。

但它并不是真实的表，而是一个查询语句对应的结果集

作用：

**1：重用子查询、简化SQL语句的复杂度**

SELECT 语句中的FROM子句中我们常会使用一个子查询，然后将结果当做表再进行查询工作，

若很多SELECT语句中都要用到该子查询，就可以将这个子查询定义为一个视图进行重用，这样也可以简化SQL语句的复杂度

**2：限制数据访问，**可以隐藏真实的表中字段信息，表的名字，字段的名字这样可以避免访问敏感信息等。

**2、oracle 对象有哪些？**

答：试图，序列，存储函数，同义词，索引，表。

扩展(数据库对象)：

表

约束条件：保证数据完整性

试图：虚表，命名的查询语句

索引：加速查询的速度

序列：一串连续递增或递减的数字，步长相同，(代理键)。

同义词：(对象的别名)一个对象的另外一个叫法。

存储过程：用于操作。

函数：用作复杂运算的。用于计算。

触发器：由事件触发的存储过程。

包

**1、数据库触发器的好处：**

答：比如上网发日志，事先会自动通知好友；其实就是在增加(发)日志时做了一个后触发，

先告知好友，再向通知表中写入信息。效率高。

**2、说出一些数据库优化方面的经验？**

答：有外键约束会影响插入和删除性能，如果程序能够保证数据的完整性，

那么就在创建数据库时去掉外键约束。

**3、什么是数据库的事务？**

答：把两组DML绑在一起作处理，要么一起成功，要么一起失败，

成功就提交事务，失败就回滚事务。

事务是作为一个逻辑单元执行的一系列操作，一个逻辑工作单元必须有四个属性，称为 ACID（原子性、

一致性、隔离性和持久性）属性，只有这样才能成为一个事务：

原子性

事务必须是原子工作单元；对于其数据修改，要么全都执行，要么全都不执行。

一致性

事务在完成时，必须使所有的数据都保持一致状态。在相关数据库中，所有规则都必须应用于事务的修

改，以保持所有数据的完整性。事务结束时，所有的内部数据结构（如 B 树索引或双向链表）都必须是

正确的。

隔离性

由并发事务所作的修改必须与任何其它并发事务所作的修改隔离。事务查看数据时数据所处的状态，要么

是另一并发事务修改它之前的状态，要么是另一事务修改它之后的状态，事务不会查看中间状态的数据。

这称为可串行性，因为它能够重新装载起始数据，并且重播一系列事务，以使数据结束时的状态与原始事

务执行的状态相同。

持久性

事务完成之后，它对于系统的影响是永久性的。该修改即使出现系统故障也将一直保持。

6、游标的作用？如何知道游标已经到了最后？

游标用于定位结果集的行，通过判断全局变量@@FETCH\_STATUS可以判断是否到了最后，通常此变量

不等于0表示出错或到了最后。

**4、JDBC中的prepared Statement相比Statement的好处？**

答：一个sql命令发给服务器去执行的步骤是：语法检查，语言分析，编译成内部指令，

缓存指令，执行指令等过程。可以防止sql注入。

**5、处理数据库大数据量下的分页解决方法？**

答：最好的办法是利用sql语句进行分页，这样每次查询出的结果集中就只包含某页数据内容，在sql语句无法实现分页的情况下，可以考虑对大的结果集通过游标定位方式来获取某页的数据。

总结：简单就是用sql分页语句进行分页；再大的数据量可以考虑对大的结果集

通过游标定位方式来获取某页的数据。

扩展：(说说你对数据库大数据量怎么处理)？

答：**可以用分区功能：**

范围分区：范围分区就是对数据表中的某个值的范围进行分区，根据某个值的范围，

决定将该数据存储在哪个分区上。如根据序号分区，根据时间等来进行分区。

**Hash分区(散列分区)**：散列分区为通过指定分区编号来均匀分布数据的一种分区型，

因为通过在I/O设备上进行散列分区，使得这些分区大小一致。

也就是只命名分区名称，这样均匀进行数据分布。

**复合分区：**有时候我们需要根据范围分区后，每个分区内的数据再散列地分布在几个表空间中， 这样我们就要使用复合分区。复合分区是先使用范围分区， 然后在每个分区内再使用散列分区的一种分区方法。

**8. 数据库里面的索引和约束是什么?**

索引是为了提高数据的检索速度，索引是建立在数据表上，

根据一个或多个字段建立的约束是为了保持数据的完整性，

约束有非空约束，主键约束，外键约束等等。

**11、一般内连接的基本语法**

答：select \* from a,b where a.id = b.id;

**9.****数据库事务的四大特性？**

数据库事务是指作为单个逻辑工作单元执行的一系列操作，这些操作要么全做要么全不做，是一个不可分割的工作单位。

数据库事务的四大特性**(简称ACID)**是：

**(1) 原子性(Atomicity)**

事务的原子性指的是，事务中包含的程序作为数据库的逻辑工作单位，它所做的对数据修改操作要么全部执行，要么完全不执行。这种特性称为原子性。

例如银行取款事务分为2个步骤(1)存折减款(2)提取现金。不可能存折减款，却没有提取现金。2个步骤必须同时完成或者都不完成。

**(2)一致性(Consistency)**

事务的一致性指的是在一个事务执行之前和执行之后数据库都必须处于一致性状态。这种特性称为事务的一致性。假如数据库的状态满足所有的完整性约束，就说该数据库是一致的。

例如完整性约束a+b=10，一个事务改变了a，那么b也应随之改变。

**(3)分离性(亦称独立性Isolation)**

分离性指并发的事务是相互隔离的。即一个事务内部的操作及正在操作的数据必须封锁起来，不被其它企图进行修改的事务看到。假如并发交叉执行的事务没有任何控制，操纵相同的共享对象的多个并发事务的执行可能引起异常情况。

**(4)持久性(Durability)**

持久性意味着当系统或介质发生故障时，确保已提交事务的更新不能丢失。即一旦一个事务提交，DBMS保证它对数据库中数据的改变应该是永久性的，即对已提交事务的更新能恢复。持久性通过数据库备份和恢复来保证。

----------------------------------------------------------------------------------------------------

**JDBC数据库连接**

----------------------------------------------------------------------------------------------------

Jdbc就是我们JAVA用来访问数据库的

**JDBC**

1 JDBC 标准：一系列接口组成

2 数据库厂商提供了，JDBC的实现（接口的实现类）

这些实现类称为 “驱动” JDBC Driver

3 JDBC 用于实现数据的持久化操作的 CRUD

执行SQL实现 CRUD

**使用JDBC连接**

1 注册驱动

2 连接数据库（TCP Socket）

3 创建语句对象（用于执行SQL）

4 执行SQL （DQL executeQuery() DML executeUpdate DDL execute）

5 处理执行结果。

6 关闭连接

**何时使用JDBC**

1）访问数据库，处理数据时候！使用JDBC

2）使用JDBC可以执行SQL语句

3）使用JDBC能够实现数据的CRUD（增删改查）

**连接池**

1 并发连接少的时候，重用连接对象，性能好！

2 并发连接多的时候，可以控制总连接数，保护数据库，避免连接过载。

3 在并发连接多的应用中，必须使用连接池！

4 连接池的管理参数，要根据实际情况调整！

5 从连接池中获得的连接，当关闭时候，只是将连接归还给连接池，并不是真的断开与数据库的连接。

**连接池的使用**

1） 导入jar包

2） 创建 DataSource（数据源）实现类的实例

3） 设置数据库基本连接参数（注意：方法名不同）

driver

url

user

pwd

4） 设置 连接池管理策略参数（可选参数）

根据实际业务需要设置，具体方法参考手册

5）getConnection() 获得连接。使用连接访问DB

6）关闭连接（归还连接到连接池！）

连接池：DataSource 接口可以翻译为 "数据源"

"数据源" == 连接池 == 数据源连接池

何时必须使用连接池: 软件中有多个并发线程同时

访问了数据库，这时候必须使用连接池！

并发访问使用 DataSource

单线程访问使用 DriverManager

**2、事务总结：**

1、 在软件业务中，如果存在一系列的 CRUD 操作

在需要进行 原子控制 时候，必须使用事务进行控制。

简单说：

如果多个 更新（删除，添加）SQL，就必须使用事务一个SQL 能处理的事情，一般可以不使用事务！

批量更新：向数据库批量发送SQL或参数，批量执行

避免了单条SQL执行时候的反复通讯过程，性能好

**3、事务（交易）ACID**

1） 在一个业务过程中，如果存在多次CRUD

保证业务过程的完整性，就必须使用事务进行保护！

2) 只有一句SQL 无需事务

3） 只是查询，无需事务

4） 不想用事务时候！

----------------------------------------------------------------------------------------------------

**Java web前端**

----------------------------------------------------------------------------------------------------

**2、Tomcat的优化经验？**

答：去掉对web.xml的监视，把jsp提前编辑成Servlet。

有富余物理内存的情况，加大tomcat使用的jvm的内存。

**Tomcat有几种部署方式？**

（1）利用Tomcat自动部署

（2）利用控制台进行部署

（3）增加自定义的Web部署文件(%Tomcat\_Home%\conf\Catalina\localhost\AppName.xml)

（4）手动修改%Tomcat\_Home%\conf\server.xml文件来部署web应用

**2、什么是Servlet**

Sun 公司制订的一种用来扩展 Web 服务器功能的组件规范。

**3、解释一下什么是 servlet;**

答:servlet 有良好的生存期的定义，包括加载和实例化、初始化、处理请求以及服务结束。

这个生存期由 javax.servlet.Servlet 接口的 init,service 和 destroy 方法表达。

**4、****Servlet的生命周期？**

答：1：实例化：容器收到请求时，会创建一个serlvet实例。

2：初始化：容器在创建好servlet对象之后，会接着调用servlet对象的init()方法。

注意：该方法只会执行一次。作用是，获取资源。

3：就绪：调用servlet对象的service()方法

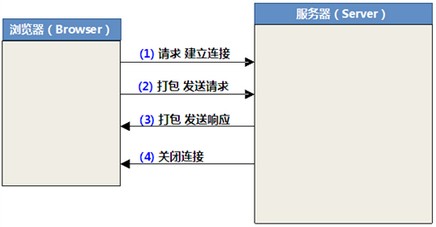
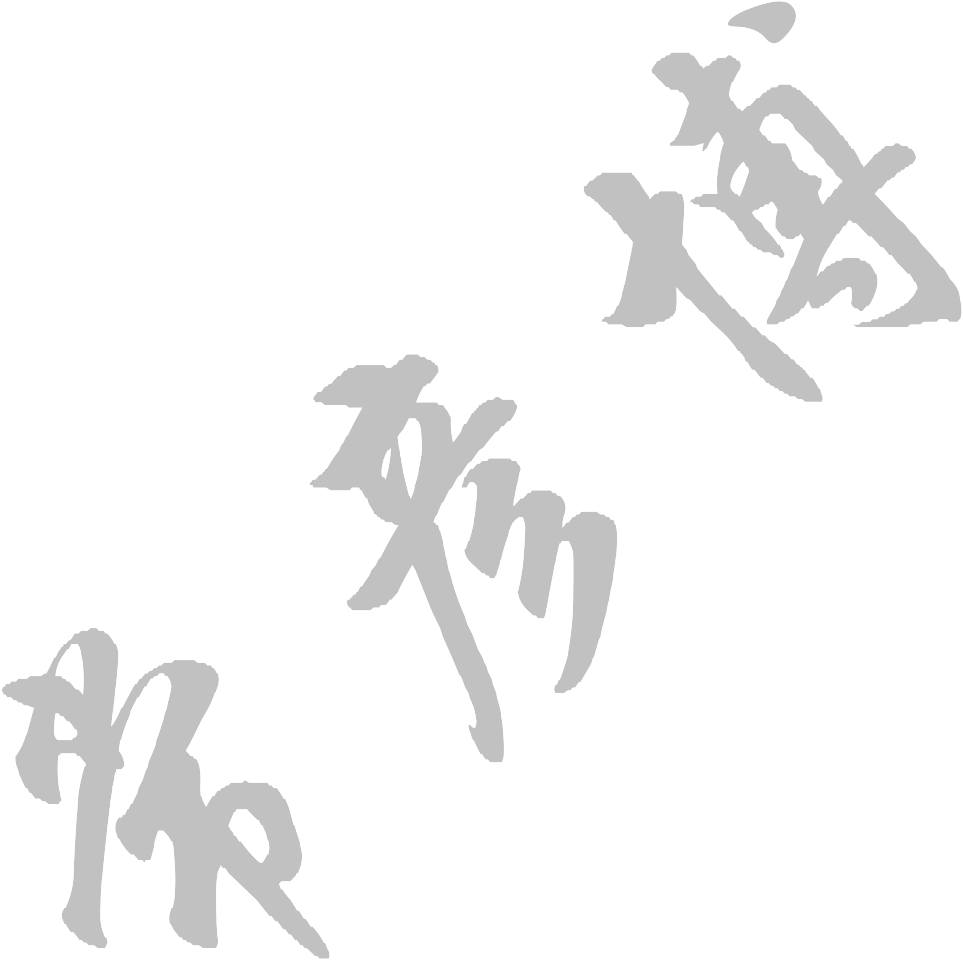
4：销毁：容器会依据自身的算法，删除servlet对象。在删除之前，会先调用destroy()方法。

**5、 什么是 HTTP 协议**

HTTP（Hypertext transport protocol）是超文本传输协议。是一种应用层协议，由 W3C 制 定，它定义了浏览器（或者其他客户端）与 Web 服务器之间通讯的过程及数据格式。

## 2.2 通讯的过程

step1：浏览器建立与 Web 服务器之间的连接（Socket）。



step2：浏览器要将请求数据打包（请求数据包），然后发送给 Web 服务器。 step3：Web 服务器将处理结果打包（响应数据包），然后发送给浏览器。

step4：Web 服务器关闭连接。

 注意事项：特点：一次请求，一次连接。优点：Web 服务器可以利用有限的连接个数为 尽可能多的客户服务（效率高）。如果浏览器要再发请求，就必须重新建立一个新的连接。

**1.get/post请求的区别**

1）get

A.如果需要向服务器传递少量数据用get；

B.get请求使用URL传值，即数据会附着在

URL上传递给服务器；

C.只能传递较少的数据；

2）post

A.如果需要提交表单，或者传递大量的

数据用post；

B.post请求使用请求数据包的实体内容

来传值；

C.可以传递大量数据；

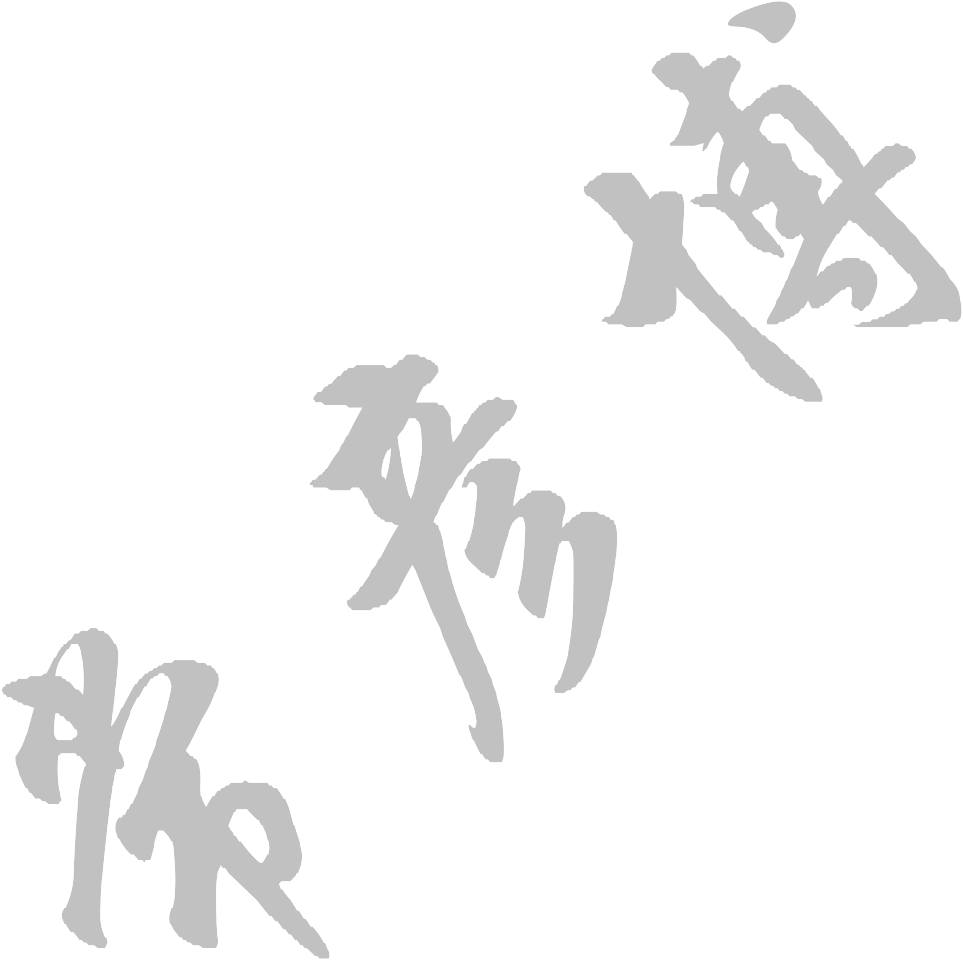
D.post请求隐私性更好，但是安全级别

和get完全相等，不能说它更安全。

## 什么是 JSP

JSP（Java Server Page）是 Java 服务器端动态页面技术。是 sun 公司制订的一种服务器端的动态页面生成技术规范。

## 为什么要使用 JSP

因为直接使用 Servlet，虽然也可以生成动态页面。但是，编写繁琐（需要使用 out.println 来输出），并且维护困难（如果页面发生了改变，需要修改 Java 代码），所以 sun 指定了 JSP 规范。

## JSP 与 Servlet 的关系

JSP 其实是一个以.jsp 为后缀的文件，容器会自动将.jsp 文件转换成一个.java 文件（其实 就是一个 Servlet），然后调用该 Servlet。所以，从本质上讲，JSP 其实就是一个 Servlet。

**5、编程用JAVA解析XML的方式？**

答：用SAX方式解析XML。用到SAXHandler这个类，继承了HandlerBase。

**13、JSP有哪些隐含对象？并简单的描述一下其含义？(记得全部就好，记不了全部就记常用的)**

答：application,config,out,page,PageContext,request,response,session,exception

application 正在执行的内容

config servlet的构架部件

out 用来传送回应的输出

page JSP网页本身

pageContext 网页的属性是在这里管理

request 此请求会包含来自get/post请求的参数

response 网页查传回客户端的响应

session 与请求有关的会话

exception 针对错误网页，未捕获的例外

**7、JSP的常用指令**

isErrorPage(是否能使用 Exception 对象)，isELIgnored(是否忽略表达式)

**8. JSP中动态 INCLUDE 与静态 INCLUDE 的区别？**

答：动态 INCLUDE 用 jsp:include 动作实现

静态 INCLUDE 用 include 伪码实现,定不会检查所含文件的 变化，适用于包含静态页面

**9、转发和重定向的区别？**

答：转发是一次请求：

转发是一件事未做完，再去做另一件事。(意思就是再去调用另外一个组件继续做。)

重定向是多次请求：

重定向是一件事已经做完了，再去做另外一件事。

**10、****MVC的各个部分都有那些技术来实现?如何实现?**

答:MVC是Model－View－Controller的简写。Model(模型) 代表的是应用的业务逻辑（通过 JavaBean，EJB组件实现），

View (视图)是应用的表示面（由JSP页面产生），

Controller (控制器)是提供应用的处理过程控制（一般是一个Servlet），通过这种设计模型把应用逻辑,处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

-----------------------------------------------------------------------------------------------

**1、什么是 Ajax**

异步的 JavaScript 和 Xml）。是一种用来改善用户体验 的技术，

其实质是利用浏览器内置的一个特殊对象（XMLHttpRequest，一般称之为 Ajax 对 象）异步地向服务器发送请求，**（Ajax 对象在向服务器发送请求时，浏览器并不会销毁当前页面，**

**用户仍然可以 对当前页面作其他的操作）**

服务器送回部分数据（**不是一个完整的新的页面，而是文本或者 Xml 文档**），

在浏览器端，可以利用这些数据部分更新当前页面。整个过程，页面无刷新，不打断用户的操作。

**1、Ajax 的优点**

1）页面无刷新，不打断用户的操作。

2）按需要获取数据，客户端（浏览器）与服务器端之间的数据传输量大大减少。

3）是一种标准化的技术，不需要下载任何插件。

**2、什么是同步请求**

Ajax 对象在向服务器发送请求时，浏览器会锁定当前页面，用户不能够对当前页面做任何的操作。

2）如何发送同步请求 open(请求方式，请求地址，false)

3）优先使用异步，因为同步会影响性能，当服务器端处理比较慢的时候，浏览器会锁定当前页面（“假死”）。

4）只有当客户端需要等待服务器的响应之后，才能继续向下执行时，应该使用同步。

 注意事项：Firefox 的某些版本（低版本，如 3、4）对于同步的支持比较特殊：

------------------------------------------------------------------------------------------------

**框架部分知识点**

------------------------------------------------------------------------------------------------

**1、struts是什么？**

struts1是基于JSP和servlet的一个开源的Web应用框架，使用的是MVC的设计模式

struts2是基于webwork技术的框架，是sun和webwork公司联手开发的一个功能非常齐全的框架，

struts2和struts1没有关系，是一个全新的框架。

**2、谈谈你对Struts的理解？**

答：struts是一个按MVC模式设计的Web层框架，其实它就是一个大大的servlet，

这个Servlet名为ActionServlet，或是ActionServlet的子类。我们可以在web.xml文件中

将符合某种特征的所有请求交给这个Servlet处理，这个Servlet再参照一个配置文件

（通常为/WEB-INF/struts-config.xml）将各个请求分别分配给不同的action去处理。

总结：struts其实是以MVC设计模式的web层框架，也就是一个大大的servlet，

而这个servlet我们称之为ActionServlet，我们可以在web.xml配置文件中

将符合某种特征的请求交给对应的Servlet类处理，再通过struts.xml配置文件

对应的action处理。

**1、spring是什么？**

spring是一个集成了许多第三方框架的大杂烩，其核心技术是IOC（控制反转，也称依赖注入）

和AOP（面向切面编程）。 可以降低系统组件之间的耦合度，便于系统组件的维护、扩展和替换。

**2、Spring框架的优点**

其实与Spring框架的作用相同：

在SSH中，主要是利用Spring

管理我们程序中的Action、DAO等Bean组件，

通过容器的IOC机制，可以降低Action、DAO之间调用的耦合度（关联度），利用AOP

可以降低共通组件和一批目标组件的耦合度，进行事务管理等共通部分的处理。

在SSH中，Struts2主要是利用它的控制器，而不是标签、表达式；Hibernate主要利用它的数据库访问；Spring主要是利用它的整合。

**3、Spring框架的容器**

Spring 框架的核心是提供了一个容器（是我们抽象出来的，代指后面的类型）。该容器类型是BeanFactory或ApplicationContext（建议用这个类型，它是BeanFactory的子类，功能更多）。

该容器具有以下功能：

1）容器可以创建和销毁 Bean 组件对象，等价于原来“工厂”类的作用。

2）容器可以采用不同的模式创建对象，如单例模式创建对象。

3）容器具有IOC机制实现。

4）容器具有AOP机制实现。

**3.1 IoC概念**

1）Inverse ofController被称为控制反转或反向控制，其实真正体现的是“控制转移”。

2）所谓的控制指的是负责对象关系的指定、对象创建、初始化和销毁等逻辑。

3）IoC指的是将控制逻辑交给第三方框架或容器负责（即把Action中的控制逻辑提出来，交给第三方负责），当两个组件关系发生改变时，只需要修改框架或容器的配置即可。

4）IoC主要解决的是两个组件对象调用问题，可以以低耦合方式建立使用关系

**什么是 AOP**

Aspect Oriented Programming，被称为面向方面编程。对单个对象（一对一）的解耦用 IOC，

而当有个共通组件，它对应多个其他组件（一对多），则解耦用 AOP。如，拦截器。

这也是为何在程序中大量的用 IoC，而 AOP 却用的很少，因为程序中不可能有很多的共通部分

**spring工作机制及为什么要用?**

1.spring mvc请所有的请求都提交给DispatcherServlet,

它会委托应用系统的其他模块负责负责对请求进行真正的处理工作。

2.DispatcherServlet查询一个或多个HandlerMapping,找到处理请求的Controller.

3.DispatcherServlet请请求提交到目标Controller

4.Controller进行业务逻辑处理后，会返回一个ModelAndView

5.Dispathcher查询一个或多个ViewResolver视图解析器,找到ModelAndView对象指定的视图对象

6.视图对象负责渲染返回给客户端。

**为什么要用：**

Spring即使一个AOP框架，也是一IOC容器。 Spring 最好的地方是它有助于您替换对象。

有了 Spring，只要用 JavaBean 属性和配置文件加入依赖性（协作对象）。

然后可以很容易地在需要时替换具有类似接口的协作对象。

**四、Spring 中各种类型的数据注入**

1 .Bean对象注入

<property name="属性"

ref="要注入Bean对象的id">

</property>

<property name="属性">

<bean class="包名.类名">

</property>

2.基本值注入

字符串,数值类型

<property name="属性" value="值">

3 .集合注入

\*List,\*Properties,Set,Map

4 .Spring表达式注入

如何获取现有对象的属性信息

**.3 Spring 中常用的事务类型**

1）REQUIRED：支持当前事务，如果当前没有事务，就新建一个事务。这是最常见的选 择，也是默认值。

2）SUPPORTS：支持当前事务，如果当前没有事务，就以非事务方式执行。

3）MANDATORY：支持当前事务，如果当前没有事务，就抛出异常。

4）REQUIRES\_NEW：新建事务，如果当前存在事务，把当前事务挂起。

5）NOT\_SUPPORTED：以非事务方式执行操作，如果当前存在事务，就把当前事务挂 起。

6）NEVER：以非事务方式执行，如果当前存在事务，则抛出异常。

7）NESTED：如果当前存在事务，则在嵌套事务内执行。如果当前没有事务，则进行与 REQUIRED 类似的操作。拥有多个可以回滚的保存点，内部回滚不会对外部事务产生影响。 只对 DataSourceTransactionManager 有效。

**8、MVC的各个部分都有那些技术来实现?如何实现?**

答:MVC 是 Model－View－Controller的简写。

M-Model 代表的是应用的业务逻辑（通过 JavaBean，EJB 组件实现），

V-View 是应用的表示面（由 JSP 页面产生），

C-Controller是提供 应用的处理过程控制（一般是一个 Servlet），

通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用

**9、Spring MVC 的工作流程**

1）客户端发送请求，请求到达 DispatcherServlet(**分发器**) 主控制器。

2）DispatcherServlet 控制器调用 HandlerMapping (**处理器映射**)处理。

3）HandlerMapping(**处理器映射**) 负责维护请求和 Controller 组件对应关系。HandlerMapping 根据请求 调用对应的 Controller 组件处理。

4）执行 Controller 组件的业务处理，需要访问数据库，可以调用 DAO 等组件。

5）Controller 业务方法处理完毕后，会返回一个 ModelAndView(**模型和视图**) 对象。该组件封装了模型数据和视图标识。

6）Servlet 主控制器调用 ViewResolver(**视图解析器**) 组件，根据 ModelAndView 信息处理。定位视图 资源，生成视图响应信息。

7）控制器将响应信息给用户输出。

**Struts 2的基本简要流程如下：**

1、客户端浏览器发出HTTP请求。

2、根据web.xml配置，该请求被FilterDispatcher接收。

3、根据struts.xml配置，找到需要调用的Action类和方法， 并通过IOC方式，将值注入给Action。

4、Action调用业务逻辑组件处理业务逻辑，这一步包含表单验证。

5、Action执行完毕，根据struts.xml中的配置找到对应的返回结果result，并跳转到相应页面。

6、返回HTTP响应到客户端浏览器。

**6、hibernate是什么？**

hibernate是基于ORM对象关系映射（完成对象数据到关系数据映射的机制）实现的，做数据持久化的工具。

Hibernate的缺点:

1.性能比较差，尤其是批处理方面，如果在大数据量开发的时候，最好使用JDBC。

**7、 Hibernate的核心接口？**

Session、SessionFactory、Transaction、Query、Criteria、Configuration

266、[请你谈谈SSH整合](http://www.mianwww.com/html/2009/01/1707.html" \o "Permanent Link: 请你谈谈SSH整合)

SSH：  
Struts（表示层）+Spring（业务层）+Hibernate（持久层）  
Struts：  
Struts是一个表示层框架，主要作用是界面展示，接收请求，分发请求。  
在MVC框架中，Struts属于VC层次，负责界面表现，负责MVC关系的分发。（View：沿用JSP，HTTP，Form，Tag，Resourse ；Controller：ActionServlet，struts-config.xml，Action）  
Hibernate：  
Hibernate是一个持久层框架，它只负责与关系数据库的操作。  
Spring：  
Spring是一个业务层框架，是一个整合的框架，能够很好地黏合表示层与持久层。