必考：

**jdbc的开发步骤**

(1) 、注册驱动、加载驱动

(2) 、获取数据库访问的连接对象

(3) 、获取到语句执行平台(Statement PreparedStatement)

(4) 、执行sql语句

executeUpdate() insert插入 update更新 delete删除

executeQuery() select查询

(5) 、处理结果集

(6) 、释放资源

**DbUtils（导包）使用步骤**

获取数据库连接

创建QueryRunner对象

使用QueryRunner对象执行sql语句

处理结果

释放资源

**javaBean的设计规范**

**1、类使用public修饰，公开**

**2、属性私有化，使用private修饰**

**3、为私有属性提供对应的setter、getter方法，方便赋值与取值**

**4、必须保证类中无参构造方法的存在！**

连接池含义，那些思想，怎么用

数据库连接池负责**分配、管理和释放**数据库连接，它允许应用程序**重复使用**一个现有的数据库连接，而不是再重新建立一个；释放空闲时间超过最大空闲时间的数据库连接来避免因为没有释放数据库连接而引起的数据库连接遗漏。这项技术能明显提高对数据库操作的性能。

使用数据库连接池技术目的是什么：**得到数据源对象！**

常用的连接池

dbcp2，，

druid 一个jar包，

c3p0 2个jar包

**(PreparedStatement)预编译平台与普通平台的区别是什么？？？**

1、预编译平台在创建时需要指定sql语句，不确定的值以？替代，从而达到了预编译的效果，有效的防止了sql注入，在执行语句之前需要先通过setObject()来为？绑定值。而普通平台Statement在创建时没有指定sql语句，所以无法防止sql注入

2、创建语句平台的方法不一样：

普通平台：连接对象.createStatement()

预编译平台：连接对象.prepareStatement(sql语句)

3、执行语句时方法的参数不一样，但是方法名一致。增删改executeUpdate() 查询executeQuery()

普通平台：方法有参数，为要执行的sql语句

预编译平台：方法是无参的

4、预编译平台执行sql的效率高！因为预编译平台在执行sql是只需要执行即可，而普通平台需要先编译再执行

servlet生命周期

**init()初始化，创建servlet**

第一次访问servlet时，会执行init()方法来创建servlet对象，而且整个生命周期中只会执行一次，所以是单例对象！

**在访问到达servlet之前就创建servlet对象！以空间换时间！在servlet配置时，添加一个loadOnStartUp属性、参数，取值范围为1~5，值越小优先级越高！**

**service()只要servlet被访问，也就是接收到请求，此方法就会执行**

**而且多个请求同时到达servlet时，也不会出现卡顿的现象，验证了servlet是支持多线程！**

**destroy()销毁**

**正常关闭tomcat服务器，右键stop**

**重新启动tomcat服务器**

**修改servlet代码也会自动执行destroy**

**服务器超时时间一到(30分钟)，也会执行destroy()**

响应头：列举常用关键字

事物（ACID四个特性）

数据库事务(database transaction) 是指作为**单个逻辑单元内的一系列操作**，要么全部成功，要么全部失败！

1. 原子性（Atomicity）  
   原子性是指事务包含的所有操作要么全部成功，要么全部失败回滚，因此事务的操作如果成功就必须要完全应用到数据库，如果操作失败则不能对数据库有任何影响。
2. 一致性（Consistency）  
   一致性是指事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态。
3. 隔离性（Isolation）  
   隔离性是当多个用户并发访问数据库时，比如操作同一张表时，数据库为每一个用户开启的事务，不能被其他事务的操作所干扰，多个并发事务之间要相互隔离。
4. 持久性（Durability）  
   持久性是指一个事务一旦被提交了，那么对数据库中的数据的改变就是永久性的，即便是在数据库系统遇到故障的情况下也不会丢失提交事务的操作。

**事务的四大隔离级别（由低 - -高 排序）**

**事务的四大隔离级别（由低 - -高 排序）**

**Read uncommitted 脏读的问题：**

读未提交，顾名思义，就是一个事务可以读取另一个未提交事务的数据

**Read committed 不可重复读：**

读提交，顾名思义，就是一个事务要等另一个事务提交后才能读取数据

**Repeatable read 幻读：**

重复读，就是在开始读取数据（事务开启）时，不再允许修改操作

**Serializable 序列化:**

**Serializable 是最高的事务隔离级别，在该级别下，事务串行化顺序执行，可以避免脏读、不可重复读与幻读。但是这种事务隔离级别效率低下，比较耗数据库性能，一般不使用。**

### 常见的响应状态码有哪些，举例说明含义?

状态码的分类：

100~199: 表示服务器已经接收到请求，但是没有处理完成，需要下一步请求才能完成，用得比较少

200~299：表示服务器成功接收请求并完成响应 **200成功状态码**

300~399：表示资源重定向、重定位 **304url重定向**

400~499：表示客户端错误

**404资源路径错误not found**

405 方法的路径问题

401 权限问题，未经授权

500~599：表示服务器内部错误 **500 空指针异常、数学异常、sql异常**

## 常见的响应头信息有哪些，说明含义。

Content-Disposition：告诉浏览器以下载的形式打开数据(支持下载)

Content-Length：响应的数据长度

**Content-Type：响应数据的MIME类型**

**MIME类型：大类型/小类型**

**text/html;charset=utf-8**

**image/jpg png jpeg**

**video/mp4 avi wmv**

**除了文本类型需要指定编码集之外，其余的MIME类型数据都是字节类型，不需要指定编码集！**

Date：响应的时间

refresh：定时刷新，应用场景居多，可以实现页面的延迟跳转，缓冲

## servlet的访问流程

**action---url-patter---servlet-name---- servlet-name---- servlet-class—servlet类—service重写**

数据库

**java类与数据库表的关系：**

1、java实体类就对应数据库的表

2、类中的属性就对应数据库表中的字段

3、类中属性的数据类型就对应表中字段的数据类型

4、类型中的属性名”必须”与表中的字段名一致

5、类创建的对象就对应表中的一条条记录

DML(数据操作语言)

insert into 表名 (字段1, 字段2, ...) value/values (值1, 值2, ...);

update 表名 set 字段名 = 值, 字段名 = 值, ... [where 条件];

delete from 表名 [where 条件];

DQL(数据查询语言)：

select 字段1, 字段2, ... from 表名 [where 条件];