### 1.hashmap，hashtabe和concurrenhashmap的区别：

Hashmap线程不安全，允许有空键空值，hashtabe线程安全不允许空键空值，因为线程安全问题hashmap比hashtabe效率高

Hashtabe是同步的hashmap不是，concurrenthashmap使用了锁分段技术来保证线程安全

Hashtabe采用的是锁机制一次锁住整个表concurrentHashMap则是一次锁住一个桶 默认将hash表分成16个桶

2.什么是线程线程和进程的区别

线程是进程的一个实体，是cpu调度和分派的基本单位，比进程小可以独立运行的基本单位

一个进程包含多个线程具有独立功能的程序是操作系统进行资源分配和调度的一个独立单位

### 3.创建线程有几种方式

1) 继承thread类：本质是是实现了runnable接口的一个实例，代表一个线程的实例。启动线程的唯一方法就是通过tead类的start（）死大特start()是一个native内体武方法它将启动一个新线程，执行run（）

2) 实现runnable接口：如果自己的类已经extends 另一个类，就无法直接extends Thread，此时，可以实现一个Runnable 接口。

3) 实现卡了波callable接口：Callabled 接口有点像是Runnable 接口的增强版 ，它以call()方法作为线程执行体，call()方法比run()方法功能更强大。call()方法可以有返回

4） 通过线程池方法：常用的线程池有：

1).newSingleThreadExecutor：创建一个单线程的线程，按照任务的提交顺序执行

2).newFixedThreadPool：创建固定大小的线程池提交一个任务创建一个线程，直到达到线程池最大大小

3).newCachedThreadPool：创建一个可缓存的线程池，大小取决于jvm

4).newScheduledThreadPool：创建一个无限大小的线程池，支持定时和周期性执行任务

5).newSingleThreadExecutor：创建一个单线程的线程, 支持定时和周期性执行任务

4.线程的基本方法有什么:

wait: 线程等待,调用wait()方法后，会释放对象的锁 wait 方法一般用在同步方法或同步代码块中。

sleep 线程睡眠,不会释放当前占有的锁,

join() 方法，等待其他线程终止

yield线程让步, 会使当前线程让出CPU 执行时间片，与其他线程一起重新竞争

interrupt线程中断,给这个线程一个通知信号，会影响这个线程内部的一个中断标识位。这个线程本身并不会因此而改变状态(如阻塞，终止等)

notify,notifyAll()唤醒在此对象监视器上等待的单个线程

### 5.在java 中wait 和sleep 方法的不同？

等待时wait 会释放锁，而sleep 一直持有锁。。wait 通常被用于线程间交互，sleep 通常被用于暂停执行。

### 3.4同步锁与死锁

同步锁是保证线程同步互斥，就是指并发执行的多个线程。

死锁是多个线程同时被阻塞，它们中的一个或者全部都在等待某个资源被释放

### 3.5乐观锁和悲观锁是什么？

乐观锁：总是假设最好的，每次去拿数据都认为不会修改所以不上锁

悲观锁：总是假设最坏的，每次去拿数据都会认为别人会修改，所以会上锁。

### 3.6线程池原理

线程池就是一个线程缓存，负责对线程进行统一分配、调优和监控。

线程池组成：

1. 线程池管理器：用于创建并管理线程池
2. 工作线程：线程池中的线程
3. 任务接口：每个任务必须实现的接口，用于工作线程调度和运行
4. 任务队列用于存放待处理的任务，提供一个缓冲机制

线程复用：

我们可以继承重写thread类，在start方法中添加不断循环调用传递过来rennable对象

### 3.7线程执行的顺序

1. 当线程数小于核心线程数时，会一直创建线程直到线程数等于核心线程数；

2. 当线程数等于核心线程数时，新加入的任务会被放到任务队列等待执行；

3. 当任务队列已满，又有新的任务时，会创建线程直到线程数量等于最大线程数；

4. 当线程数等于最大线程数，且任务队列已满时，新加入任务会被拒绝。

### 3.8 线程池的核心参数有哪些？

1.corePoolSize （核心线程数）

2.queueCapacity （任务队列容量）

3.maxPoolSize （最大线程数）

4.keepAliveTime （线程空闲时间）

### 4.1 JVM 内存管理

1). JVM 将内存划分为 6 个部分：PC 寄存器.虚拟机栈.堆.方法区.运行时常量池.本地方法栈

2) PC 寄存器:用于记录当前线程运行时的位置

3) java 虚拟机栈:在创建线程时创建的，用来存储栈帧

4) java 堆:堆的主要作用就是存储对象

5) 方法区:方发区被各个线程共享，用于存储静态变量.运行时常量池等信息。

6) 本地方法栈 ： 本地方法栈的主要作用就是支持 native 方法

### 4.2 GC 分代收集算法和分区收集算法区别

1. 分代收集算法：

对象存活周期的不同将内存划分为几块，入JVM中的新生代，老年代，永久代

2. 分区收集算法：

分区算法则将整个堆空间划分为连续的不同小区间, 每个小区间独立使用, 独立回收

### 4.3 GC 垃圾收集器

实现了自动的内存清理，因此，我们的内存垃圾回收主要集中于 java 堆和方法区中，在程序运行期间，这部分内存的分配和使用都是动态的.

### 4.6 内存泄漏检查

年老代堆空间被占满

异常： java.lang.OutOfMemoryError: Java heap space

持久代被占满

异常：java.lang.OutOfMemoryError: PermGen space

堆栈溢出

异常：java.lang.StackOverflowError

线程堆栈满

异常：Fatal: Stack size too small

系统内存被占满

异常：java.lang.OutOfMemoryError: unable to create new native thread

### 二.Java Web

### 1.JDBC 技术

### 1.1 说下原生jdbc 操作数据库流程？

第一步：Class.forName()加载数据库连接驱动；

第二步：DriverManager.getConnection()获取数据连接对象;

第三步：根据SQL 获取sql 会话对象，有2 种方式Statement、

PreparedStatement ;

第四步：执行SQL 处理结果集，执行SQL 前如果有参数值就设置参数值

setXXX();

第五步：关闭结果集、关闭会话、关闭连接。

### 1.2 说说事务的概念，在JDBC 编程中处理事务的步骤。

1 事务是作为单个逻辑工作单元执行的一系列操作。

2，一个逻辑工作单元必须有四个属性，称为原子性、

一致性、隔离性和持久性(ACID) 属性，只有这样才能成为

一个事务事务处理步骤：

3，conn.setAutoComit(false);设置提交方式为手工提交

4，conn.commit()提交事务

5，出现异常，回滚conn.rollback();

### 1.3JDBC 的脏读是什么？哪种数据库隔离级别能防止脏读？

脏读：有一行数据刚更新，与此同时另一个查询读到了这个刚更新的值。

TRANSACTIONREPEATABLEREAD 和TRANSACTION\_SERIALIZABLE 隔离级别可以防止脏读

### 2.网路通讯部分

### 2.1TCP 与UDP 区别？

UDP:

a、是面向无连接, 将数据及源的封装成数据包中,不需要建立建立连接

b、每个数据报的大小在限制64k 内

c、因无连接,是不可靠协议

d、不需要建立连接,速度快

TCP：

a、建议连接，形成传输数据的通道.

b、在连接中进行大数据量传输，以字节流方式

C、通过三次握手完成连接,是可靠协议

d、必须建立连接m 效率会稍低

聊天、网络视频会议、桌面共享用的就是UDP

2.2 说一下什么是Http 协议？

客户端和服务器端之间数据传输的格式规范，格式简称为“超文本传输协议”。

是一个基于请求与响应模式的、无状态的、应用层的协议，基于TCP 的连接方式

2.3get 与post 请求区别？

区别一：

get 重点在从服务器上获取资源，post 重点在向服务器发送数据；

区别二：

get 传输数据是通过URL 请求

post 传输数据通过Http 的post 机制，

区别三：

Get 传输的数据量小，因为受URL 长度限制，但效率较高；

Post 可以传输大量数据，所以上传文件时只能用Post 方式；

get 是不安全的，因为URL 是可见的，

post 较get 安全性较高；

区别五：

get 方式只能支持ASCII 字符，向服务器传的中文字符可能会乱码。

post 支持标准字符集，可以正确传递中文字符

### 2.4 http 中重定向和请求转发的区别？。

本质区别：转发是服务器行为，重定向是客户端行为。

重定向特点：两次请求，浏览器地址发生变化，可以访问自己web 之外的资

源，传输的数据会丢失。

请求转发特点：一次强求，浏览器地址不变，访问的是自己本身的web 资源，

传输的数据不会丢失。

### 3.1 Cookie 和Session 的区别

Cookie 是web 服务器发送给浏览器的一块信息，

Session 是存储在web 服务器端的一块信息。

Cookie 和session 的不同点：

1、无论客户端做怎样的设置，session 都能够正常工作。当客户端禁用cookie

时将无法使用cookie。

2、在存储的数据量方面：session 能够存储任意的java 对象，cookie 只能存

储String 类型的对象。

### 4.1.讲下Servlet 的执行流程。

Servlet 的执行流程也就是servlet 的生命周期，当服务器启动的时候生命

周期开始，然后通过init()方法初始化servlet，再根据不同请求调用doGet 或doPost 方法，最后再通过destroy()

方法进行销毁。

### 4.2.doGet 和doPost 的区别

doGet 和doPost 都是接受用户请求的方法，

doGet 处理get 请求，doPost 处理post 请求，

doGet 用于地址栏提交，doPost 用于表单提交，

在页面提交数据时，get 的数据大小有限制4k，post 没有限制，get 请求提交的数据会在地址栏显示，post 不显示，所以post 比get 安全

### 4.3.Jsp 和servlet 的区别

jsp 的可读性强，容易维护，并且jsp 在最后会编译成servlet 容易调试

### 4.4.JSP 九大内置对象:

pageContext ：只对当前jsp 页面有效，里面封装了基本的request 和session 的对象

Request ：对当前请求进行封装

Session ：浏览器会话对象，浏览器范围内有效

Application ：应用程序对象，对整个web 工程都有效

Out ：页面打印对象，在jsp 页面打印字符串

Response ：返回服务器端信息给用户

Config ：单个servlet 的配置对象，相当于servletConfig 对象

Page ：当前页面对象，也就是this

Exception ：错误页面的exception 对象，如果指定的是错误页面，这个就是

异常对象

### 4.5.三大指令：

Page ：指令是针对当前页面的指令

Include ：用于指定如何包含另一个页面

Taglib ：用于定义和指定自定义标签

4.6 七大动作：

Forward，执行页面跳转，将请求的处理转发到另一个页面

Param ：用于传递参数

Include ：用于动态引入一个jsp 页面

Plugin插件 ：用于下载javaBean 或applet 到客户端执行

useBean ：使用javaBean

setProperty ：修改javaBean 实例的属性值

getProperty ：获取javaBean 实例的属性值

### 5.1. 谈谈你对ajax 的认识？

Ajax 沟通前后台的桥桥梁。是一种创建交互式网页应用的的网页开发技术；Asynchronous JavaScript

and XML”的缩写。

Ajax 的优势：

通过异步模式，提升了用户体验。

优化了浏览器和服务器之间的传输，减少不必要的数据往返，减少了带宽占用。

Ajax 的最大特点：

可以实现局部刷新，在不更新整个页面的前提下维护数据，提升用户体验度

### 5.2.手写ajax

$.ajax({

url:'http://www.baidu.com',

type:'POST',

data:data,

cache:true,

headers:{},请求头

beforeSend：function(){},

success成功:function(){},

error错误:function(){},

complete完成:function(){}

});

### 5.3.Ajax 你以前用过么?简单介绍一下

AJAX = 异步JavaScript 和XML。

AJAX 是一种用于创建快速动态网页的技术。

通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。

传统的网页（不使用AJAX）如果需要更新内容，必需重载整个网页面。

有很多使用AJAX 的应用程序案例：新浪微博、Google 地图、开心网等等。

### 5.4.Ajax 可以做异步请求么?

可以.ajax 请求默认是异步的.如果想同步把async 设置为false 就可以

了默认是true

如果是jquery

$.ajax({

43 / 196

url: some.php,

async: false,

success : function(){

}

});

如果是原生的js

xmlHttp.open("POST",url,false);

### 5.5. jsonp 原理

服务器调用客户端的函数（即回调函数），在客户端就能拿到服务端传入的参数（返回结果）

### 5.6.Jquery 常用选择器

1、ID 选择器#id

2、类选择器.class

3、元素选择器element

4、\*

### 6.2 vue 的优点是什么？

1. 低耦合：视图（View）可以独立于Model 变化和修改，一个ViewModel 可以绑

定到不同的"View"上，当View 变化的时候Model 可以不变，当Model 变化的时候View

也可以不变。

2. 可重用性：把一些视图逻辑放在一个ViewModel 里面，让很多view 重用这段视图逻辑

3. 独立开发：开发人员可以专注于业务逻辑和数据的开发（ViewModel），设计人

员可以专注于页面设计。

4. 可测试：界面素来是比较难于测试的，而现在测试可以针对ViewModel 来写。

### 6.3 请说出vue 中生命周期函数(钩子函数)

1、beforeCreate():组件实例刚刚被创建(el 和data 并未初始化)

2、created():组件创建完成,属性已绑定,但DOM 还未生成,$el 属性还不存在(完成data

数据的初始化)

3、beforeMount():模板编译/挂载之前(完成了el 和data 初始化)

4、Mounted():模板编译/挂载之后(完成挂载)

5、beforeUpdate():组件更新之前

6、updated()：组件更新之后

7、beforedestroy():组件销毁之前

8、destroyed():组件销毁之后

### 6.4.vue 的指令

v-if 指令判断是否隐藏

v-show 指令

v-else 指令

v-for 指令数据循环出来

v-bind:class 指令绑定一个属性

v-on 指令

v-model 指令： 实现双向绑定它是一个语法糖：真实写法

是：value=" msg "@input 注（ ：是v-bind，@是v-on）

v-text 读取文本，不能读取html 标签

v-html 能读取html 标签

三**.**数据库

### 3.1MySQL 和Oracle 的区别？

1. Oracle 是大型数据库而Mysql 是中小型数据库，Oracle 市场占有率达40%，Mysql只有20%左右，同时Mysql 是开源的而Oracle 价格非常高。

2. Oracle 支持大并发，大访问量，是OLTP 最好的工具。

3.MySQL 的分页关键词limit，Oracle 的分页rownum

### 3.2Sql 之连接查询

外连接：

1） 左连接（左外连接）以左表为基准进行查询,左表数据会全部显示出来,

2） 右连接（右外连接）以右表为基准进行查询,右表数据会全部显示出来,

3） 全连接就是先以左表进行左外连接，然后以右表进行右外连接。

内连接：

显示表之间有连接匹配的所有行。

### 3.3Sql 之聚合函数：

聚合函数：

SQL 中提供的聚合函数可以用来统计、求和、求最值等等。

分类：

–COUNT：统计行数量

–SUM：获取单个列的合计值

–AVG：计算某个列的平均值

–MAX：计算列的最大值

–MIN：计算列的最小值

### 3.4 Sql 之SQL 注入

使用mybaits 时mapper 中#方式能够很大程度防止sql 注入，$方式无法

防止sql 注入.

### 3.4 SQL Select 语句完整的执行顺序:

from--->where--->group by--->having--->计算所有的表达式--->order by-- ->select 输出

### 3.5 存储引擎

MySQL 5.5 及更高版本，默认的存储引擎是InnoDB。在5.5 版本之前，MySQL

的默认存储引擎是MyISAM

1、InnoDB 存储引擎

特点：行锁设计、支持外键、支持非锁定读

采用聚集的方式存储表中数据

2、MyISAM 存储引擎

不支持事务、表锁设计，支持全文搜索

缓冲池只缓存索引文件，不缓存数据文件

3、NDB 存储引擎

数据全部放在内存中，因此主键查找速度极快

4、Memory 存储引擎

表中的数据存放在内存，数据库重启或崩溃表中的数据都会消失

存储变长字段时按照定长字段方式，浪费内存（eBay 工程师给出了patch 解决方案）

5、Archive 存储引擎

只支持INSERT 和SELECT 操作，5.1 版本之后支持索引

适合存储归档数据，如ri’zhi

6、Federated 存储引擎

不存放数据，只是指向一台远程MySQL 数据库服务器上的表

7、Maria 存储引擎

支持缓存数据和索引文件

行锁设计

提供MVCC 功能

支持事务和非事务安全的选项

### 3.6 索引

**1.**什么是索引？

索引是MySQL 数据库中的重要对象之一，用于快速找出某个列中有某一特定值的

行。

**2.**为什么要使用索引

索引是MySQL 中一种十分重要的数据库对象。它是数据库性能调优技术的基础，

常用于实现数据的快速检索。

**4.**索引的优点

创建唯一性索引，保证数据库表中每一行数据的唯一性

大大加快数据的检索速度

**5.**索引的缺点

创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加

### 3.7 数据库三范式

第一范式（1NF），原子性，列或者字段不能再分

第二范式（2NF），唯一性，不存在部分依赖

第三范式（3NF），直接性，不存在传递依赖

### 3.8. 数据库事务

1) 原子性（Atomicity）：事务是一个完整的操作。事务的各步操作是

不可分的（原子的）；要么都执行，要么都不执行。

2) 一致性（Consistency）：当事务完成时，数据必须处于一致状态。

51 / 196

3) 隔离性（Isolation）：对数据进行修改的所有并发事务是彼此隔离的，

这表明事务必须是独立的，它不应以任何方式依赖于或影响其他事

务。

4) 永久性（Durability）：事务完成后，它对数据库的修改被永久保持，

事务日志能够保持事务的永久性

4. 事务的四种隔离级别

1) Read uncommitted

读未提交，。

2) Read committed

读已提交，

3) Repeatable read

可重复读，

4) Serializable 序列化

Serializable 是最高的事务隔离级别，

。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。

在MySQL 数据库中， 支持上面四种隔离级别， 默认的为

Repeatable read (可重复读)；而在Oracle 数据库中，只支持Serializable

(串行化)级别和Read committed (读已提交)这两种级别，其中默认的为

Read committed 级别。

### 3.9. 存储过程。

存储在数据库中，经过第一次编译后再次调用不需要再次编译，存储过程是数据库中的一个重要对象

### 3.10 触发器

触发器是一段能自动执行的程序，是一种特殊的存储过程

触发器是一个具有记忆功能的，具有两个稳定状态的信息存储器件，是构成多种时序电路的最基本逻辑单元，也是数字逻辑电路中一种重要的单元电路。

### 3.11 数据库锁

数据库锁一般可以分为两类，一个是悲观锁，一个是乐观锁。

乐观锁，直到修改完成准备提交所做的的修改到数据库的时候才会将数据锁住。完成更改后释放。

悲观锁：总是假设最坏的，每次去拿数据都会认为别人会修改，所以会上锁。

### 3.12 分区分表

1.什么是mysql 分表和分区

什么是分表，就是把一张表分成N 多个小表

什么是分区，分区呢就是把一张表的数据分成N 多个区块，这些区块可以在同一个

磁盘上，也可以在不同的磁盘上

### 3.13 应该使用哪一种方式来实施数据库分库分表，这要看数据库中数据量的瓶颈所在，并综合项目的业务类型进行考虑。

原本存储于一个库的数据分块存储到多个库上，把原本存储于一个表的数据分块存储到多个表上。

### 3.14 mysql 读写分离

是让主数据库处理事务性增、改、删操作。一般来说都是通过 主从复制（Master-Slave）的方式来同步数据，再通过读写分离（MySQL-Proxy）来提升数据库的并发负载能力 这样的方案来进行部署与实施的。

### MySQL SQL 语句优化

1. 对查询进行优化，应尽量避免全表扫描
2. 避免在where 子句中使用!=或<>操作符
3. 避免在where 子句中对字段进行null 值判断
4. 避免在where 子句中使用or 来连接条件
5. in 和not in 也要慎用，否则会导致全表扫

### 3.16 数据库优化方案整理

优化分片键

不要使用分区，分区性能很低，难以维护

不要使用主从数据库，读写分离有了缓存就再也不需要分主从了，毕竟延迟问题是无法解决的（虽然很小

2、优化索引

（1）索引列务必重复度低

（2）使用索引就不能用OR 查询，否则索引不起作用

（3）使用索引，like 模糊查询不能以%开头

（4）查询条件务必以索引列开头，否则索引失效

（5）联合索引遵守最左原则

3、优化缓存

分离冷热数据，对于大内存访问评率低的数据适当分离

提高缓存命中率，命名问题

**4**、优化**sql**

1. 列类型尽量定义成数值类型，且长度尽可能短，如主键和外键，类型字段等等

2. 建立单列索引

3. 根据需要建立多列联合索引

### 四.Spring 框架

### 1. Spring 是什么?

Spring 是一个轻量级的IoC 和AOP 容器框架。是为Java 应用程序提供基础

性服务的一套框架，目的是用于简化企业应用程序的开发，它使得开发者只需要关心业务需

求。

常见的配置方式有三种：基于XML 的配置、基于注解的配置、基于Java 的配置。

主要由以下几个模块组成：

Spring Core：核心类库，提供IOC 服务；

Spring Context：提供框架式的Bean 访问方式，以及企业级功能（JNDI、定时任

务等）；

Spring AOP：AOP 服务；

Spring DAO：对JDBC 的抽象，简化了数据访问异常的处理；

Spring ORM：对现有的ORM 框架的支持；

Spring Web：提供了基本的面向Web 的综合特性，例如多方文件上传；

Spring MVC：提供面向Web 应用的Model-View-Controller 实现。

### 2.Spring 的优点？

（1）spring 属于低侵入式设计，代码的污染极低；

（2）spring 的DI 机制将对象之间的依赖关系交由框架处理，减低组件的耦合性；

（3）Spring 提供了AOP 技术，支持将一些通用任务，如安全、事务、日志、权

限等进行集中式管理，从而提供更好的复用。

60 / 196

（4）spring 对于主流的应用框架提供了集成支持。

### 3、Spring 的AOP 理解？

OOP 面向对象，允许开发者定义纵向的关系，但并适用于定义横向的关系，导致了

大量代码的重复，而不利于各个模块的重用。

AOP，一般称为面向切面，作为面向对象的一种补充，用于将那些与业务无关，但

却对多个对象产生影响的公共行为和逻辑，抽取并封装为一个可重用的模块，这个模块被命

名为“切面”（Aspect），

Spring AOP 中的动态代理主要有两种方式，JDK 动态代理和CGLIB 动态代理：

### 4、Spring 的IoC 理解？

IOC 就是控制反转，是指创建对象的控制权的转移，对象与对象之间松散耦合, 也利于功能的复用.

（2）最直观的表达就是，IOC 让对象的创建不用去new 了，可以由spring 自动

生产

### 5、Spring 的IOC 有三种注入方式

Spring 的IOC 有三种注入方式：构造器注入、setter 方法注入、根据注解注入。

### 6、请解释Spring Bean 的生命周期？

（1）实例化Bean：

（2）设置对象属性（依赖注入）：

（3）处理Aware 接口：

（4）BeanPostProcessor：

（5）InitializingBean 与init-method：

（7）DisposableBean：

（8）destroy-method：

第一阶段：实例化（Instantiation）

第二阶段： 属性赋值

 第三阶段：初始化(Initialization)

初始化前： org.springFrameWork.beans.factory.config.BeanPostProcessor#postProcessBeforeInitialization

          初始化中         org.springFrameWork.bean.InitializingBean#afterPropertiesSet

初始化后org.springFrameWork.beans.factory.config.BeanPostProcessor#postProcessAfterInitialization

第四阶段：销毁

### 7、解释Spring 支持的几种bean 的作用域。

Spring 容器中的bean 可以分为5 个范围：

（1）singleton：默认，每个容器中只有一个bean 的实例，单例的模式由

BeanFactory 自身来维护。

（2）prototype：为每一个bean 请求提供一个实例。

（3）request：为每一个网络请求创建一个实例，在请求完成以后，bean 会失效

并被垃圾回收器回收。

（4）session：与request 范围类似，确保每个session 中有一个bean 的实例，

在session 过期后，bean 会随之失效。

（5）global-session：全局作用域，global-session 和Portlet 应用相关。

### 8、使用注解之前要开启自动扫描功能

@Configuration 把一个类作为一个IoC 容器，它的某个方法头上如果注册了@Bean，

就会作为这个Spring 容器中的Bean。

@Scope 注解作用域

@Lazy(true) 表示延迟初始化

@Service 用于标注业务层组件、

@Controller 用于标注控制层组件（如struts 中的action）

@Repository 用于标注数据访问组件，即DAO 组件。

@Component 泛指组件，当组件不好归类的时候，我们可以使用这个注解进行标注。

@Scope 用于指定scope 作用域的（用在类上）

---------------jsr250----

@PostConstruct 用于指定初始化方法（用在方法上）

@PreDestory 用于指定销毁方法（用在方法上）

@Resource 默认按名称装配，当找不到与名称匹配的bean 才会按类型装配。

----------

@DependsOn：定义Bean 初始化及销毁时的顺序

者，否则将抛出异常

@Autowired 默认按类型装配，如果我们想使用按名称装配，可以结合@Qualifier、、

### 9、Spring 框架中的单例Beans 是线程安全的么？

某种程度上说Spring 的单例bean 是线

程安全的。如果你的bean 有多种状态的话（比如View Model 对象），就需要自行保证

线程安全。最浅显的解决办法就是将多态bean 的作用域由“singleton”变更为“prototype”。

### 10、Spring 如何处理线程并发问题？

因为Spring 对一些Bean 中非线程安全状态采用ThreadLocal 进行处理，解决线程安全问题。

ThreadLocal 会为每一个线程提供一个独立的变量副本，从而隔离了多个线程对数

据的访问冲突。因为每一个线程都拥有自己的变量副本，从而也就没有必要对该变量进行同

步了。ThreadLocal 提供了线程安全的共享对象，在编写多线程代码时，可以把不安全的变

量封装进ThreadLocal。

**11**、**Spring** 基于**xml** 注入**bean** 的几种方式：

（1）Set 方法注入；

（2）构造器注入：①通过index 设置参数的位置；②通过type 设置参数类型；

（3）静态工厂注入；

（4）实例工厂；

### 12、Spring 的自动装配：

在spring 中，对象无需自己查找或创建与其关联的其他对象，由容器负责把需要相

互协作的对象引用赋予各个对象，使用autowire 来配置自动装载模式。

互协作的对象引用赋予各个对象，使用autowire 来配置自动装载模式。

在Spring 框架xml 配置中共有5 种自动装配：

（1）no：默认的方式是不进行自动装配的，通过手工设置ref 属性来进行装配bean。

（2）byName：通过bean 的名称进行自动装配

（3）byType：通过参数的数据类型进行自动装配。

（4）constructor：利用构造函数进行装配

（5）autodetect：自动探测，如果有构造方法，通过construct 的方式自动装配，

### 13、@Autowired 和@Resource 之间的区别

1) @Autowired 默认是按照类型装配注入的，默认情况下它要求依赖对象必须存

在（可以设置它required 属性为false）。

(2) @Resource 默认是按照名称来装配注入的，只有当找不到与名称匹配的bean

才会按照类型来装配注入。

### 14、Spring 框架中都用到了哪些设计模式？

（1）工厂模式：BeanFactory 就是简单工厂模式的体现，用来创建对象的实例；

（2）单例模式：Bean 默认为单例模式。

（3）代理模式：Spring 的AOP 功能用到了JDK 的动态代理和CGLIB 字节码生成

技术；

（4）模板方法：用来解决代码重复的问题。比

如. RestTemplate, JmsTemplate, JpaTemplate。

（5）观察者模式：定义对象键一种一对多的依赖关系，当一个对象的状态发生改

变时，所有依赖于它的对象都会得到通知被制动更新，如Spring 中listener 的实现

--ApplicationListener。

**15**、**Spring** 事务的实现方式和实现原理：

Spring 事务的本质其实就是数据库对事务的支持，没有数据库的事务支持，spring

是无法提供事务功能的。真正的数据库层的事务提交和回滚是通过 binlog 或者 redo log

实现的。

（1）Spring 事务的种类：

spring 支持编程式事务管理和声明式事务管理两种方式：

①编程式事务管理使用 TransactionTemplate。

②声明式事务管理建立在 AOP 之上的。其本质是通过 AOP 功能，对方法前后进行

拦截，将事务处理的功能编织到拦截的方法中，也就是在目标方法开始之前加入一个事务，

在执行完目标方法之后根据执行情况提交或者回滚事务。

（2）spring 的事务传播行为：

spring 事务的传播行为说的是，当多个事务同时存在的时候，spring 如何处理这些

事务的行为。

① PROPAGATION\_REQUIRED：如果当前没有事务，就创建一个新事务，如果当

前存在事务，就加入该事务，该设置是最常用的设置。

69 / 196

② PROPAGATION\_SUPPORTS：支持当前事务，如果当前存在事务，就加入该事

务，如果当前不存在事务，就以非事务执行。‘

③ PROPAGATION\_MANDATORY：支持当前事务，如果当前存在事务，就加入

该事务，如果当前不存在事务，就抛出异常。

④ PROPAGATION\_REQUIRES\_NEW：创建新事务，无论当前存不存在事务，都

创建新事务。

⑤ PROPAGATION\_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 PROPAGATION\_NEVER：以非事务方式执行，如果当前存在事务，则抛出异常。

⑦ PROPAGATION\_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，则按 REQUIRED 属性执行。

（3）Spring 中的隔离级别：

读未提交该隔离级别允许脏读，其隔离级别是最低的。也就是说，如果一个事务正在处理某一数据，并对其进行了更新，但同时尚未完成事务，因此还没有提交事务；而以此同时，允许另一个事务也能访问该数据。

读已提交是不同的时间执行时只能获取到已经提交的数据。

可重复读：保证在事务处理过程中，多次读取同一个数据时，该数据的值和事务开始时刻 是一致的。因此该事务级别禁止了不可重复读取和脏读，但是有可能出现幻读的数据。

串行化，顺序读是最严格的事务隔离级别。它要求所有的事务排队顺序执行，即事务只能一个接一个地处理，不能并发。