

试题一

阅读以下

某公司拟开发一套在线软件开发系统，支持用户通过浏览器在线进行软件开发活动。该系统的重要功能，包括我的编辑，语法高亮提示，代码编频，系统调试，代码气库管理等，在需求分析与架构设计阶段，公司提出的需求和质量属性描述如下：

- a)根据用户的付费情况对用户进行分类，并根据类别提供相应的开发功能。
- b)在正常负载情况下，系统应该在 0.2s 内对用户的界面操作请求进行响应
- c)系统应该具备完善的安全防护措施，能够对黑客的攻击行为进行检测和防御
- d)系统主站点断电后应在 3s 内将请求重定向到备用站点
- e)系统支持中文昵称，但用户名必须以字母开头，长度不少于 8 个字符
- f)系统宕机后，需要在 15s 内发现错误，并启用备用系统
- g)在正常负载情况下，户的代码提交请求应在 0.5s 内完成
- h)系统支持硬件设备灵活扩容，应保证在 2 人天内完成
- i)系统需要针对代码仓库的所有操作进行详细记录；便于后期查阅与审计
- j)更改系统 web 界面风格需要在 4 人天内完成
- k)系统本身需要提供远程调试接口，支持开发团队进行远程排错

在对系统需求质量属性和架构特性进行分析的基础上，该公司的系统架构给了两种方案

【问题 1】（13 分）

请问该需求应该采用哪一种风格。标 1-1 是对这两种风格分别从数据处理方式、系统扩展方式和处理性能三个方面进行了比较，请填写表 1-1 中（1）～（4）处的空白。

表 1-1

架构风格名称	数据处理方式	系统扩展方式	处理性能
管道-过滤器	数据驱动机制，处理流程事先确定，交互性差	(2)	劣势：需数据格式转换，性能调低 优势：支持开发调用，性能调高
仓库	(1)	数据与处理松耦合，可动态添加或删除组件	劣势：(3) 优势：(4)

【问题 2】（12 分）

请分析题干中的需求描述，填写图中 1-2 中的（1）～（6）处的空白。

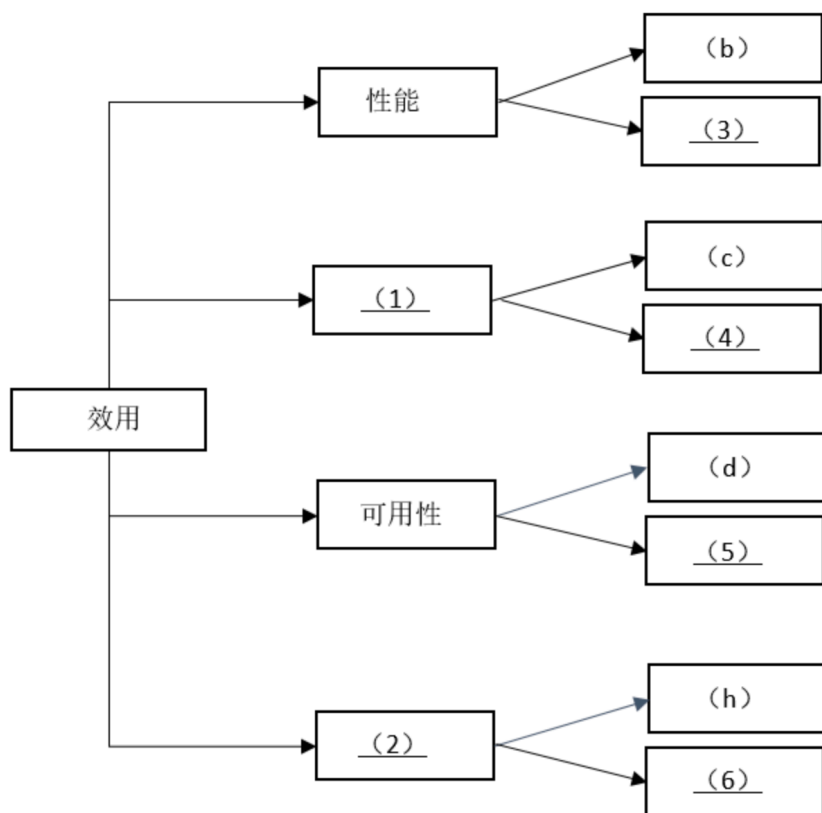


图 1-2 效用树

参考答案:

【问题 1】

1.应该采用仓库风格。(5 分)

2.表(1) - (4)空的空白分别为: (8 分)

(1)文件或模型驱动机制, 处理流程事先不确定, 交互性强

(2)数据与处理紧耦合, 以接口适配方式实现扩展

(3)中央系统构件体现系统状态, 系统执行没有确定的顺序, 效能调低?

(4)共享中心数据, 加载共享数据, 性能调高?

【问题 2】

(1) 安全性

(2)可修改性

(3) g

(4) i

(5) f

(6) j

答案解析:

本题考查的是架构设计过程中涉及到的一些质量属性, 以及架构风格的对比。

【问题 1】

在管道/过滤器风格的软件体系结构中, 每个构件都有一组输入和输出, 数据输入构件, 经过内部处理, 然后产生数据输出。因此, 这里的构件被称为过滤器, 这种风格的连接件就像是数据流传输的管道, 将一个过滤器的输出传到另一过滤器的输入。

在仓库(repository)风格中, 有两种不同的构件: 中央数据结构说明当前状态, 独立构件在中央数据存储上执行。一方面, 若构件控制共享数据, 则仓库是一传统型数据库; 另一方面, 若中央数据结构的当前状态触发进程执行的选择, 则仓库是一黑板系统。通过交互方式、数据结构、控制结构和扩展方法分别对仓库风格和管道过滤器风格进行对比, 如下所示:

交互方式: 管理过滤器很明显是顺序结构或循环结构, 数据在管理中进行传递。而仓库结构是数据在中心位置, 所有的处理均是中心结点与周边结点之间的交互, 从形态来看, 是星型的。

数据结构: 从数据结构来看, 仓库风格会使用一个文件将数据保存起来, 所有的操作围绕这个文件进行。而管道过滤器则是在过滤器之间传递数据流。

控制结构: 从控制结构来说仓库风格是业务功能驱动, 而管道过滤器是由数据流驱动的。

扩展方法:从扩展方法来讲,管道过滤器是通过过滤器提供标准接口与其它过滤器对接,而数据仓库风格,要共享数据,扩展功能,只要功能的操作与数据模型本身是匹配的就行了,就像我们要共享一个数据库做系统集成,此时共享同一数据库的多个应用系统所用的数据模型一定会是一致的,否则无法去共享。

【问题 2】本题主要考查考生对于软件质量属性的理解、掌握和应用。

本题考查的是架构设计过程中涉及到的一些质量属性,以及架构风格的对比。常用的质量属性包括:

1、性能

性能(performance)是指系统的响应能力,即要经过多长时间才能对某个事件做出响应,或者在某段时间内系统所能处理的事件的个数。

2、可靠性

可靠性(reliability)是软件系统在应用或系统错误面前,在意外或错误使用的情况下维持软件系统的功能特性的基本能力。

3、可用性

可用性(availability)是系统能够正常运行的时间比例。经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。

4、安全性

安全性(security)是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。安全性又可划分为机密性、完整性、不可否认性及可控性等特性。

5、可修改性

可修改性(modifiability)是指能够快速地对较高的性能价格比对系统进行变更的能力。通常以某些具体的变更为基准,通过考察这些变更的代价衡量可修改性。

6、易用性

软件开发工具应有十分友好的用户界面,用户乐于使用;工具应能剪裁和定制,以适应特定用户的需要;工具应能提示用户的交互操作,提供简单有效的执行方式;工具还应能检查用户的操作错误,尽可能自动改正错误。

识别软件架构质量属性是进行架构设计的重要步骤。根据对相关质量属性的定义和含义,其中:“c)系统应该具备完善的安全防护措施,能够对黑客的攻击行为进行检测和防御”、“i)系统需要针对代码仓库的所有操作进行详细记录;便于后期查阅与审计”属于安全性;“h)系统支持硬件设备灵活扩容,应保证在 2 人天内完成”、“j)更改系统 web 界面风格需要在 4 人天内完成”这描述的是系统的可修改性;“g)在正常负载情况下,用户的代码提交请求应在 0.5s 内完成”描述的是性能属性。

试题二

某企业委托软件公司开发一套包裹信息管理系统,以便于对该企业通过快递收发的包裹信息进行统一管理,在系统设计阶段,需要对不同快递信息的包裹单信息进行建模,其中,邮政包裹单如图:

[问题 1] (13 分)

请说明关系型数据库开发中,逻辑数据模型设计过程包含哪些任务?

[问题 2] (6 分)

请说明什么是超类实体?结合图中包裹单信息,试设计一种超类实体,给出完整的属性列表。

[问题 3] (6 分)

请说明什么是派生属性?结合图中包裹单信息说明哪个属性是派生属性。

国内快递包裹详情单（通知单联）

邮1106乙



中国邮政
CHINA POST



KA01905413323

接收局号码:

收件人	□□□□□□	内件品名及数量	是否保价 是 <input type="checkbox"/> 否 <input type="checkbox"/>	寄件人名章:
	姓名: _____ 电话: _____			
	单位名称: _____			
	详细地址: _____			
寄件人	姓名: _____ 电话: _____	保价金额: _____ 元	重量: _____ 克	
	单位名称: _____	寄件人声明: 同意并遵守背面的“使用须知”, 如包裹无法投递, 按如下选择处理:	资费: _____ 元	
	详细地址: _____	<input type="checkbox"/> 退还寄件人 <input type="checkbox"/> 拒收处理	挂号费: _____ 元	
	用户代码: _____ 邮政编码: _____	签字: _____	保价费: _____ 元	
		检查人员名章: _____	回执费: _____ 元	
			总计: _____ 元	

KA 0190-5413 3 23

填写本单前, 请阅读背面的“使用须知”。您的签名意味着您理解并接受“使用须知”内容。

国家邮政局监制 10-10-03

图2-1 包裹详情单

参考答案:

[问题 1]

逻辑结构设计包含的任务主要有:

- (1)把概念结构设计阶段设计好的基本 E-R 图转换为关系数据库的关系模式;
- (2) 对关系模式进行优化;
- (3) 设计用户视图。

设计的关系模式主要有:

收件人信息。主键:证件号

寄件人信息。主键:用户代码

包裹单信息。主键:包裹单编号

快递员信息。主键:员工编号

邮局站点信息。主键:邮局编号

[问题 2]

将一些子实体所共有的属性抽象为一个单独的新实体, 这个新的实体就是超类实体。

比如根据这个包裹单信息可以设计一个“人员信息”的超类:人员信息(姓名, 电话, 单位名称, 详细地址, 邮政编码)。

[问题 3]

可以从其它属性得来的属性就叫派生属性。包裹图中的“总计”属性是派生属性。

答案解析:

数据库设计分为概念结构设计、逻辑结构设计物理结构设计:

概念设计也称为概念结构设计, 其任务是在需求分析阶段产生的需求说明书的基础上, 按照特定的方法将它们抽象为一个不依赖于任何 DBMS 的数据模型, 即概念模型。概念模型的表现形式即 ER 模型。

逻辑设计也称为逻辑结构设计, 其主要任务是将概念设计阶段设计好的 E-R 图转换为与选用的具体机器上的 DBMS 所支持的数据模型相符合的逻辑结构(如: 关系模式)。

物理设计也称为物理结构设计, 其任务是对给定的逻辑模型选取一个最适合应用环境的物理结构, 所谓数据库的物理结构, 主要是指数据库在物理设备上的存储结构和存取方法。

[问题 2]

当较低层次上实体类型表达了与之联系的较高层次上的实体类型的特殊情况时, 就称较高层次上实体类型为超类型, 反之为子类型。子类到超类的过程为概化, 超类到子类的过程为特化。

①子类与超类之间具有继承特点, 即子类包含了超类的所有属性, 并且可以比超类拥有更多的属性。

②这种继承性是通过子类实体和超类实体有相同的实体标识符实现的。

[问题 3]

可以从其它属性得来的属性就叫派生属性。包裹图中的“总计”属性是派生属性。可以从资费、挂号费、保价费、回执费累加计算出来。

试题三

阅读以下关于数据管理的方案设计，在答题纸商回答问题 1 至问题 3.

互联网公司因业务发展要建立网上平台为用户提供一个对网络文化产品进行评论（小说，电影等）交流的平台，该平台的部分功能如下：

- (a)用户帖子的评论计数器
- (b)支持粉丝列表功能
- (c)支持标签管理
- (d)支持共同好友功能
- (e)提供排名功能
- (f)用户的信息结构化存储
- (g)提供好友信息的发布/订阅功能。

该系统在性能上需要考虑高性能，并发，以支持大量的用户同时访问，经过考虑，在数据管理上，决定采用 Redis+数据库的解决方案。

[问题 1]

Redis 支持丰富的类型，请选择题干描述的(a)-(g)功能选项。填入表 4-1 中(1)-(5)空白处。

表 4-1 Redis 数据类型与业务功能对照表

数据类型	存储的值	可实现的业务功能
String	字符中等	(1)
List	列表	(2)
Set	无序集合	(3)
Hash	包括键值对的无序列表	(4)
Zset	有序集合	(5)

[问题 2]

缓存中存储当前的热点数据 Redis 为每个 key 值都设置了过期时间，以提高缓存命中率，为了消除非热点数据 Redis 选择了定期删除加惰性删除策略。如果该策略失效 Redis 内存使用率会越来越高，一般采用内存淘汰机制来解决。请用 100 字以内的文字，简要描述该策略的失效场景，并给出三种内存淘汰机制。

参考答案：

[问题 1]

- (1) a
- (2) b
- (3) d
- (4) f
- (5) e

[问题 2]

由于 redis 定期删除是随机抽取检查，不可能扫描清除掉所有过期的 key 并删除，然后一些 key 由于未被请求，惰性删除也未触发。这样 redis 的内存占用会越来越高。此时就需要内存淘汰机制。

常用的内存淘汰机制是六种(volatile-lru、volatile-ttl、volatile-random、allkeys-lru、allkeys-random、no-enviction)。注:任选其中三种都可得分。

答案解析：

【问题 1】

类型	简介	特性	场景
String (字符串)	二进制安全	可以包含任何数据,比如 jpg 图片或者序列化的对象,一个键最大能存储 512M	—
Hash (字典)	键值对集合,即编程语言中的 Map 类型	适合存储对象,并且可以像数据库中 update 一个属性一样只修改某一项属性值 (Memcached 中需要取出整个字符串反序列化成对象修改完再序列化存回去)	存储、读取、修改用户属性
List (列表)	链表 (双向链表)	增删快,提供了操作某一段元素的 API	1、最新消息排行等功能 (比如朋友圈的时间线) 2、消息队列
Set (集合)	哈希表实现,元素不重复	1、添加、删除,查找的复杂度都是 $O(1)$ 2、为集合提供了求交集、并集、差集等操作	1、共同好友 2、利用唯一性,统计访问网站的所有独立 ip 3、好友推荐时,根据 tag 求交集,大于某个阈值就可以推荐
Sorted Set (有序集合)	将 Set 中的元素增加一个权重参数 score,元素按 score 有序排列	数据插入集合时,已经进行天然排序	1、排行榜 2、带权重

【问题 2】

过期策略:即 redis 针对过期的 key 使用的清除策略,策略为:定期删除+惰性删除。

1、定期删除:

redis 会将每个设置了过期时间的 key 放入到一个独立的字典中,以后会定期遍历这个字典来删除到期的 key。redis 默认是每隔 100ms 就随机抽取一些设置了过期时间的 key,检查其是否过期,如果过期就删除。注意这里是随机抽取的。为什么要随机呢?你想想假如 redis 存了几十万个 key,每隔 100ms 就遍历所有的设置过期时间的 key 的话,就会给 CPU 带来很大的负载。

2、惰性删除:

所谓惰性策略就是在客户端访问这个 key 的时候,redis 对 key 的过期时间进行检查,如果过期了就立即删除,不会给你返回任何东西。

定期删除可能会导致很多过期 key 到了时间并没有被删除掉。所以就有了惰性删除。假如你的过期 key,靠定期删除没有被删除掉,还停留在内存里,除非你的系统去查一下那个 key,才会被 redis 给删除掉。这就是所谓的惰性删除,即当你主动去查过期的 key 时,如果发现 key 过期了,就立即进行删除不返回任何东西。

由于 redis 定期删除是随机抽取检查,不可能扫描清除掉所有过期的 key 并删除,然后一些 key 由于未被请求,惰性删除也未触发。这样 redis 的内存占用会越来越高。此时就需要内存淘汰机制。

主要有如下一些策略:

1、volatile-lru: 从设置过期时间的数据集中挑选出最近最少使用的数据淘汰。没有设置过期时间的 key 不会被淘汰,这样就可以在增加内存空间的同时保证需要持久化的数据不会丢失。

2、volatile-ttl:除了淘汰机制采用 LRU,策略基本上与 volatile-lru 相似,从设置过期时间的数据集中挑选将要过期的数据淘汰,ttl 值越大越优先被淘汰。

3、volatile-random: 从已设置过期时间的数据集中任意选择数据淘汰。当内存达到限制无法写入非过期时间的数据集时,可以通过该淘汰策略在主键空间中随机移除某个 key。

4、allkeys-lru: 从数据集中挑选最近最少使用的数据淘汰,该策略要淘汰的 key 面向的是全体 key 集合,而非过期的 key 集合。

5、allkeys-random: 从数据集中选择任意数据淘汰。

6、no-eviction: 禁止驱逐数据,也就是当内存不足以容纳新入数据时,新写入操作就会报错,请求可以继续,线上任务也不能持续进行,采用 no-eviction 策略可以保证数据不被丢失,这也是系统默认的淘汰策略。

试题五

关于 Web 系统架构设计的叙述。

说明:开发基于 Web 的基业设备检测系统,以实现多种工业数据的分类采集,运行状态检测以及相关信息的管理该系统应具备以下功能:

现场设备状态采集功能，根据数据类型对设备检测指标状态信号进行分类采集

设备采集数据传输功能：9-11 月可靠的传输技术，实现将设备数据从制造现场传输到系统后台

设备检测显示功能：对设备的运行状态工作以及报警状态进行检测并提供相应的图形化界面

设备信息管理功能：支持设备运行历史状态，报警记录参数信息的查询。

同时，该系统还需满足以下非功能性需求。

- (a) 系统应支持大于 100 个工业设备的进行检测
- (b) 设备数据以制造现场传输到系统后台传输时间小于 1s
- (c) 系统应在 7*24 小时工作
- (d) 可抵御见 XSS 攻击
- (e) 系统在故障情况下，应在 0.5 小时内恢复
- (f) 支持数据审计

面对系统需求，公司召开项目讨论会议，制定系统设计方案最终决定使用三层拓补结构，即现场设备数据采集层、Web 检测服务层和前端 Web 显示层。

[问题 1]（6 分）

请按照性能、安全性和可用性三种非功能需求分类在选择题的(a)~(f)填入(1)~(3)，非功能性需求归类表：

非功能性需求类别	非功能性需求
性能	(1)
安全性	(2)
可用性	(3)

[问题 2]（14 分）

该系统 Web 检测服务层拟采用 SSM 框架进行系统研发，SSM 工作流程图如图 5-1 所示，请从下面给出的(a)~(k)中进行选择，补充完善图 5-1 中(1)~(7)处空白的内容：

- (a) Connection pool
- (b) Struts2
- (c) Persistent Layer
- (d) Mybatis
- (e) HTTP
- (f) MVC
- (g) Kafka
- (h) ViewLayer
- (i) Jsp
- (j) Controller Layer
- (k) Spring

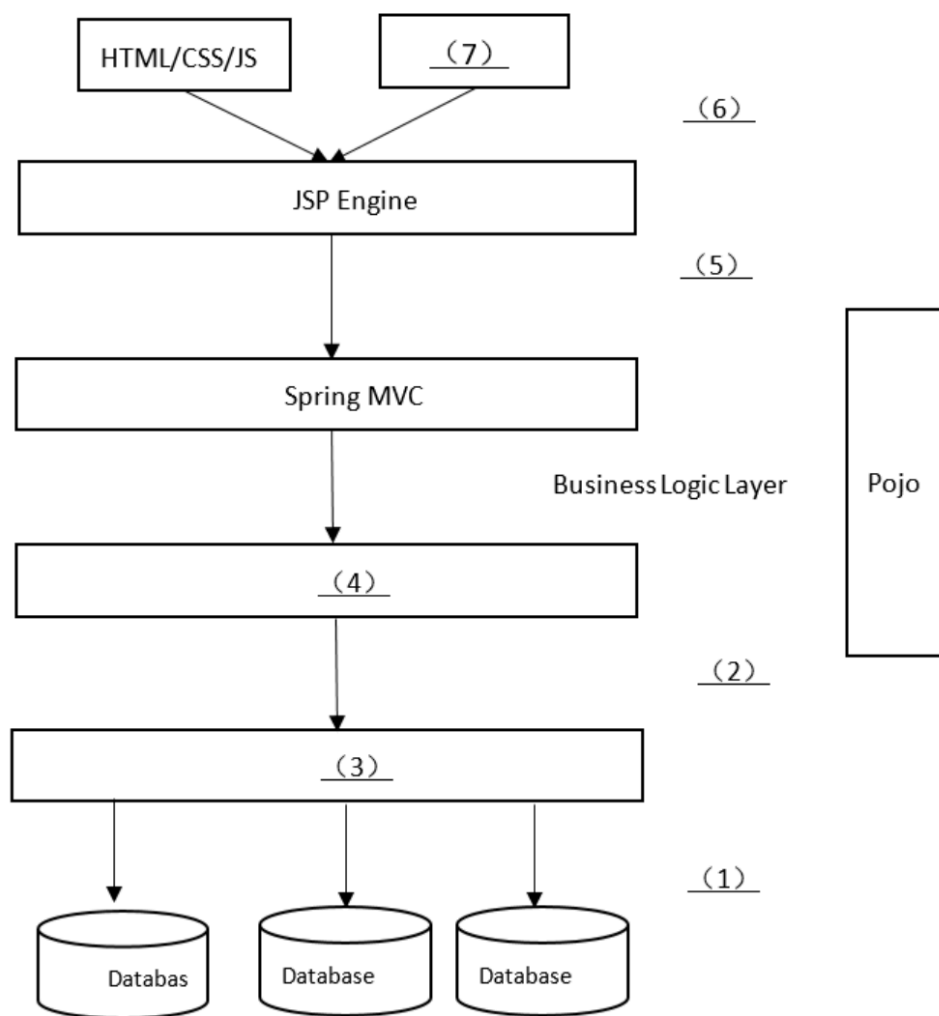


图 5-1 SSM 框架工作流程图

【问题 3】（5 分）

该工业设备检测系统拟采用工业控制领域中统一的数据访问机制，实现与各种不同设备的数据交互，请用 100 以内的文字说明采用标准的数据访问机制的原因。

参考答案：

【问题 1】

(1) b、a

(2) d、f

(3) c、e

【问题 2】

(1) a

(2) c

(3) d

(4) k

(5) j

(6) h

(7) i

【问题 3】

通过采用标准的数据访问机制，可以屏蔽不同设备之间数据交互的差异，解决了系统使用数据不一致性，又在一定程度上减低了数据结构与应用系统的耦合度，减少了应用系统的维护的工作量。

答案解析：

Spring 是一个轻量级的企业级应用开发框架，于 2004 年由 Rod Johnson 发布了 1.0 版本，经过多年的更新迭代，已经逐渐成为 Java 开源世界的第一框架，Spring 框架号称 Java EE 应用的一站式解决方案，与各个优秀的 MVC 框架如 SpringMVC、Struts2、JSF 等可以无缝整合，与各个

ORM 框架如 Hibernate、MyBatis、JPA 等也可以无缝衔接，其他各种技术也因为 Spring 的存在而被很容易地整合进项目开发之中，如 Redis 整合、Log4J 整合等等

SpringMVC 是 Spring 框架体系中的全功能 MVC 模块。SpringMVC 是基于 Java 语言实现 MVC 设计模式的请求驱动类型的轻量级 Web 框架，目的是将 Web 开发模块化及代码简化。其提供了 DispatcherServlet 前端控制器分派请求，同时提供灵活的配置处理程序映射、视图解析,并支持文件上传，目前已经是众多 MVC 框架中的佼佼者。

MyBatis 的前身是 Apache 社区的一个开源项目 iBatis,于 2010 年 更名为 MyBatis。MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀的持久层框架，避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集，使得开发人员更加关注 SQL 本身和业务逻辑，不用再去花费时间关注整个复杂的 JDBC 操作过程。

Spring+spring mvc+mybatis 整合的框架组件图如下所示:

