**作业一 自己总结的SSM知识点：**

### **spring**

**spring IOC创建对象的两种方式**

1.创建容器对象

2.在当前包的所在类下找有关bean的配置文件 测试方便

**spring主配置文件创建对象的方式**

1.调用无参构造器

2.调用带参构造器

3.用工厂模式创建 工厂里面分为静态方法和非静态方法

1.构造函数赋值

2.set方法注入值

    分为 普通字段赋值和集合属性赋值(list map property)

**di 处理依赖关系**

四种方式

最常用的在主配置文件中通过set方法注入,在service接收dao注入值,在controller接收service注入值

内部bean

p名称空间,给属性注入值

自动装配 配置简单,但维护麻烦

**配置方式和注解方式的区别**

配置便于维护 但配置太多很繁琐

注解开发方便简化配置,但不利于后期维护,如果修改对象创建,关系处理等需要改代码

**Java中的代理 分为**

静态代理

    1.目标对象必须实现接口

    2.代理对象要实现和目标对象一样的接口

动态代理(jdk代理)

    目标对象一定要实现接口,代理对象不用实现接口

Cglib代理(spring默认支持) 也叫子类代理

    jdk动态代理有一个限制,使用动态代理的对象必须实现接口

    如果向代理没有实现接口类,使用cglib

    1.目标对象可以不实现接口

    2.目标类不能final 如果final报错

    3.方法是final/static,不会被代理拦截,会直接执行目标对象方法

**代理总结**

    1.目标对象有实现接口,使用jdk动态代理

    2.没有实现接口,使用cglib代理

    3.没有实现接口且为final,不能进行aop编程,不能生成代理会报错

**spring生成代理对象的过程**

    1.创建容器对象时,根据切入点表达式拦截的类,生成代理对象

    2.目标对象有实现接口,使用jdk动态代理

    3.没有实现接口,使用cglib代理

    4.从容器获取代理后的对象

    5.执行代理对象方法,运行时动态植入切面类中的通知

**AOP(面向切面编程)是什么**

    实现原理就是Java动态代理,但是动态代理必须实现接口,所以spring的aop是用cglib实现的

**事务**

    一组操作的执行单元,相对于数据库操作来讲事务管理的是一组SQL指令(增删改查)

    事务的一致性要求操作必须全部执行成功,出现任何一条SQL语句没有执行成功,这一组

    操作会全部回滚

spring声明式事务管理,就是粗粒度的事务控制

spring创建SessionFactory的方式

所有配置都在spring中完成

spring对dao操作的支持

spring提供了JDBCTemple模版的工具类,对原始的jdbc操作进行了简化

### **mybatis**

mybatis操作dao层 直接写SQL语句 更加灵活

查询操作:返回单条记录时,表字段和对应实体属性命名一致时可以不使用resultMap属性配置,直接使用resultType="返回的全类名"

查询操作:返回多条记录时,返回结果配置的不是集合类型,而是集合元素的类型,参数通过Map等方式进行封装

**mybatis两种查询方式**

association 用于多对一或一对一

collection 用于一对多或多对多

Mybatis和hibernate一样，也使用缓存；缓存分为一级缓存和二级缓存，一级缓存指在SqlSession内；二级缓存能被所有的SqlSession共享

mybatis简化配置:扫描式加载SQL映射文件

### **springmvc**

**配置文件都需要配置什么**

前端过滤器,控制器,视图解析器,中文乱码处理

简述SSM框架整合的步骤？

1、 建项目导jar包

2、 配置web.xml中的SpringMVC的核心控制器DispatcherServlet

3、 配置applicationContext-Mybatis.xml Spring配置文件 database.properties log4j.propertise mybatis-config.xml Springmvc-servlet.xml

4、 事务管理配置SqlSessionFactoryBean MapperScannerConfiguer

5、 对象数据模型pojo

6、 数据访问接口Dao

7、 系统服务接口 sevices

8、 前端控制层controller

9、 Tools工具类

10、 前端页面jsp和静态资源statics

面向过程编程，面向对象编程和面向切面编程理解

面向过程(ProcedureOriented)是一种以过程为中心的编程思想。这些都是以什么正在发生为主要目标进行编程，不同于面向对象的是谁在受影响。与面向对象明显的不同就是封装、继承、类。

面向对象编程（ObjectOriented Programming，OOP，面向对象程序设计）是一种计算机编程架构。OOP 的一条基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。OOP 达到了软件工程的三个主要目标：重用性、灵活性和扩展性。为了实现整体运算，每个对象都能够接收信息、处理数据和向其它对象发送信息。

面向切面编程（AspectOriented Programming(AOP)），是一个比较热门的话题。AOP主要实现的目的是针对业务处理过程中的切面进行提取，它所面对的是处理过程中的某个步骤或阶段，以获得逻辑过程中各部分之间低耦合性的隔离效果。

怎么理解SpringAOP？

SpringAOP是一种通过预编译和运行期间动态实现代理的方式实现了不修改代码的情况下给程序动态添加新功能的技术。

一方面，使用mySQL从数据库中读取数据进行缓存，就可以从内存中读取数据了，而不像数据库一样读磁盘，现实是读操作比写操作要多得多，所以缓存很多常用的数据。另一方面，它也能满足互联网高并发需要高速处理数据的场合。保证并发数据安全和一致性。