回顾

1. maven工程的目录结构:

工程名

Pom..xml maven工程管理文件

Src

Main 程序目录

Java java源码目录

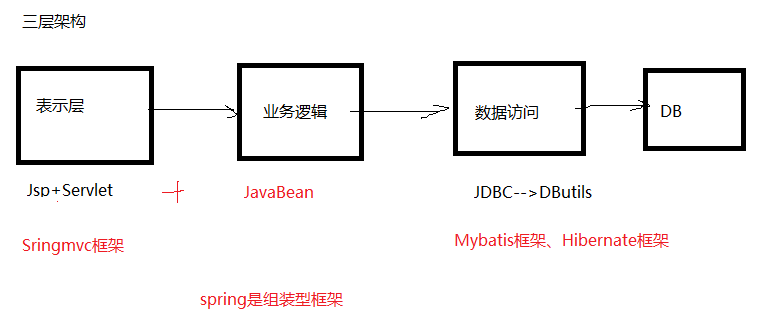
Resources 程序资源文件目录

Test 测试目录

java源码目录

Resources 程序资源文件目录

2. 三层架构:



## Mybatis第一章

课程目标:

1. 理解框架
2. **理解数据持久化概念和ORM原理 重点 难点**
3. **理解MyBatis的概念以及优点特性**
4. **（重点）搭建MyBatis环境**

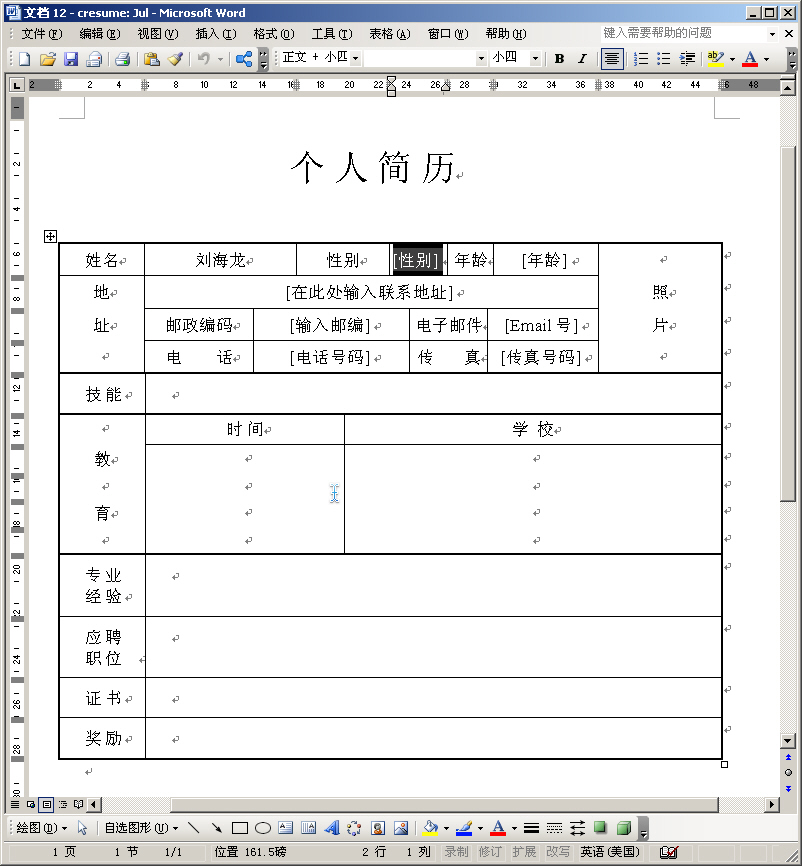
**>>实现年级查询功能**

1. **了解MyBatis与JDBC的区别与联系**

MyBatis就是JDBC封装

1. 框架
2. 为什么需要框架技术

**问题一: 如何更快更好地写简历？**



答: **使用word简历模板**

**问题二： 那使用模板有哪些好处呢?**

* **不用考虑布局、排版等，提高效率**
* **可专心在简历内容上**
* **结构统一，便于人事阅读**
* **新手也可以作出专业的简历**

1. 什么是框架技术

框架是一个应用程序的半成品，提供可重用的公共结构。按照规则组织的一组组件.

1. 框架优点 ：

不用考虑公共问题、专心在业务实现上、结构统一易于维护、提高开发效率

1. 常用框架

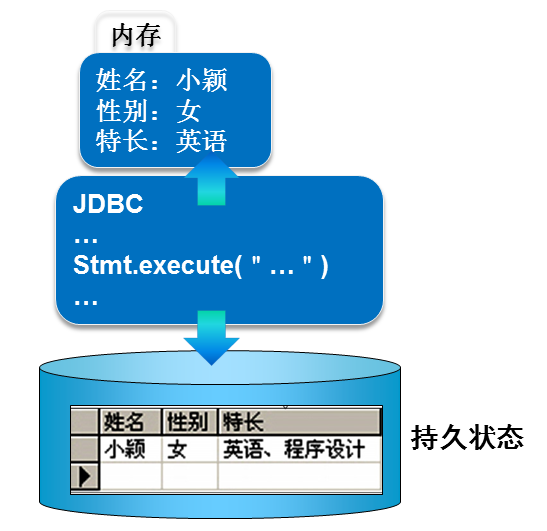
Mybatis、Spring🡪SpringBoot、SpringMVC 、Strtus2、Hibernate

SSM:Spring+SpringMvc+Mybatis

SSH: Spring+Struts2|Springmvc+Hibernate

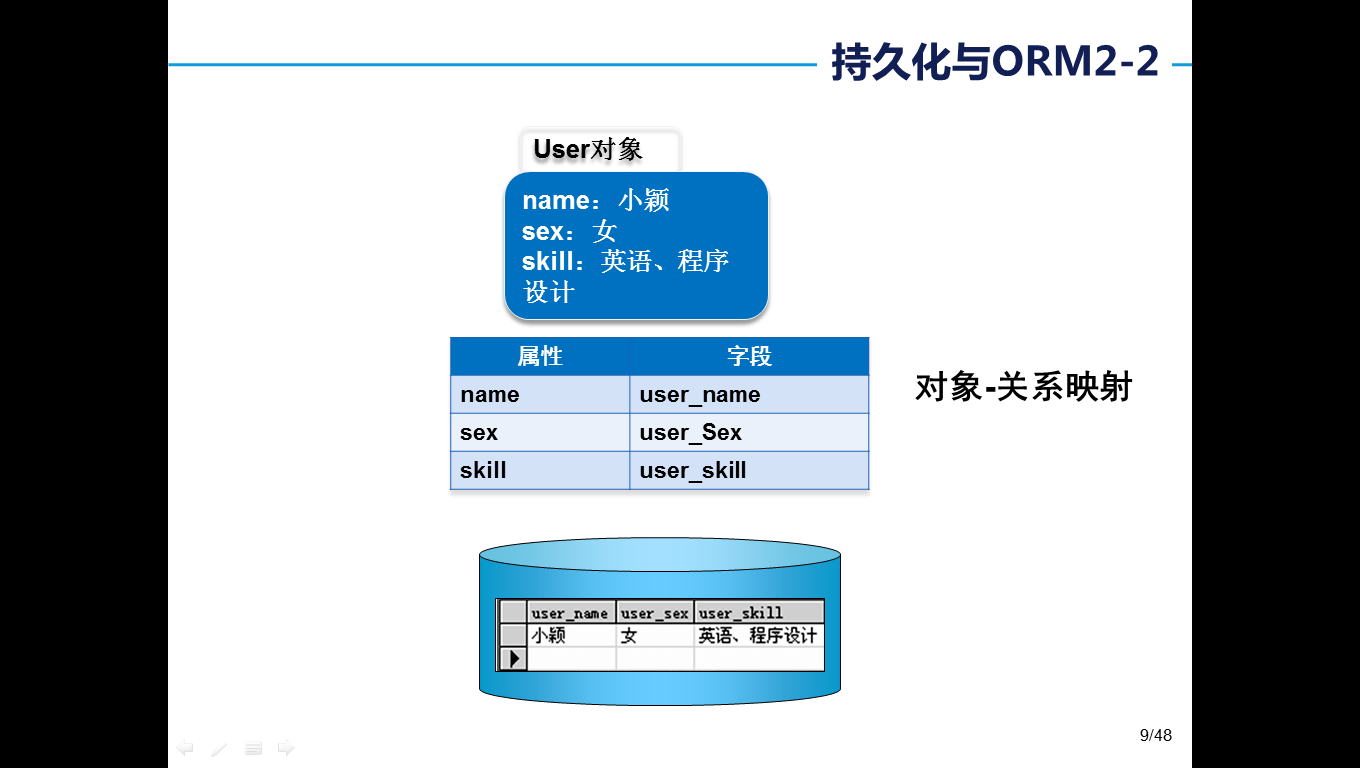
1. 理解持久化与ORM的概念
2. 持久化:

持久化是程序数据在瞬时状态与持久状态之间的转换过程.



1. (必需会)**ORM（Object Relational Mapping）:对象关系映射**
   1. **编写程序的时候，以面向对象的方式处理数据**
   2. **保存数据的时候，却以关系型数据库的方式存储**

注意:Mybatis是ORM半自动化(写sql)实现、而Hibernate是ORM全(无需编写sql)实现



1. Mybatis
2. 介绍MyBatis

MyBatis(前身ibatis)是Orm的半自动化（编写sql）实现，充当持久化框架。

1. (重点)搭建Mybatis的开发环境
   1. 添加mybatis相关的jar包(在pom.xm文件中添加依赖)

|  |
| --- |
| *<!-- MySQL依赖 start -->* <**dependency**>  <**groupId**>mysql</**groupId**>  <**artifactId**>mysql-connector-java</**artifactId**>  <**version**>5.1.25</**version**> </**dependency**> *<!-- MySQL依赖 end --> <!-- 加入MyBatis 依赖 start -->* <**dependency**>  <**groupId**>org.mybatis</**groupId**>  <**artifactId**>mybatis</**artifactId**>  <**version**>3.2.5</**version**> </**dependency**> *<!-- 加入MyBatis 依赖 end -->* |

2.2在工程中添加mybatis的配置文件(mybatis-config.xml),并放在resources目录中

a. 指定连接数据库的配置

b. 加载sql映射文件(定义的持久化操作)

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* **<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd"*>*** <**configuration**>  *<!-- 指定义连接数据的配置 -->* <**environments default="development"**>  <**environment id="development"**>  <**transactionManager type="JDBC"**/>  <**dataSource type="POOLED"**>  <**property name="driver" value="com.mysql.jdbc.Driver"**/>  <**property name="url" value=" jdbc:mysql://localhost:3306/k0501db?useUnicode=true&amp;characterEncoding=utf-8**  **"/>**  <**property name="username" value="root"**/>  <**property name="password" value="123"**/>  </**dataSource**>  </**environment**>  </**environments**>    *<!-- 加载sql映射文件-->* **<mappers>  <mapper resource=" com/team/project/dao/GradeMapper.xml"/>**  **</mappers>** </**configuration**> |

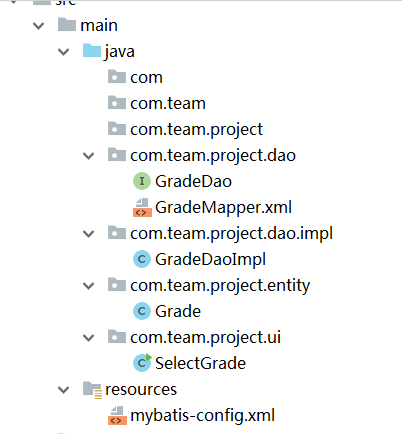
3 . 编写sql映射文件(文件名称:表名+Mapper.xml)，并放到接口包中

注意: sql映射文件中存放的都是sql语句(一个sql语句一个功能一个持久化操作)

|  |
| --- |
| <**mapper namespace="com.team.project.dao.GradeDao"**>  *<!--注意:namespace属性指定持久化操作存放的包，推荐包名用接口的限定名-->  <!--定义一系列的持久化操作-->  <!--定义一个查询的持久化操作  id属性表示操作的名称, 推荐id值为接口的方法名称  resultType 表示查询语句返回的实体类型  -->* <**select id="getAllGrade" resultType="com.team.project.entity.Grade"**>  select *\** from grade  </**select**> </**mapper**>  注意: **namespace的值推荐使用接口包的限定名** |

注意:sql映射文件一定义要在mybatis配置文件进行引入

1. 创建maven实现查询所有年级
2. 创建maven工程和还原数据



1. 创建Grade实体

|  |
| --- |
| **public class** Grade {  **private int gid**;  **private** String **gname**;  **setter和getter略** } |

1. 编写Dao层实现查询年级

接口:

|  |
| --- |
| **public interface** GradeDao {  **public** List<Grade> getAllGrade(); } |

实现: 使用Mybatis调用持久操作的步骤

|  |
| --- |
| **public class** GradeDaoImpl **implements** GradeDao {  @Override  **public** List<Grade> getAllGrade() {  **try** {  *//------编写sql,使用jdbc技术   //------使用Mybatis实现  //1.将sql语句定义在sql映射文件中  //2.Mybatis执行调用持久化操作  //2.1读取配置文件* Reader reader=Resources.*getResourceAsReader*(**"mybatis-config.xml"**);  *//2.2获取SqlSessionFactory工厂* SqlSessionFactoryBuilder sqlSessionFactoryBuilder=**new** SqlSessionFactoryBuilder();  SqlSessionFactory sqlSessionFactory=sqlSessionFactoryBuilder.build(reader);  *//2.3获取SqlSession对象调用持久化操作* SqlSession sqlSession=sqlSessionFactory.openSession();  *//2.4调用持久化操作  //sqlSession.selectList("持久化操作的id值");* List<Grade> grades= sqlSession.selectList(**"com.kgc.dao.GradeDao.getAllGrade"**);  *//2.5关闭sqlSession* sqlSession.close();  **return** grades;  }**catch** (Exception e){  e.printStackTrace();  }  **return null**;  } } |

1. 编写主函数:调用dao层

|  |
| --- |
| **public static void** main(String[] args) {  *//我是表示层，调用dao层获取所有年级* GradeDao gradeDao=**new** GradeDaoImpl();  List<Grade> list=gradeDao.getAllGrade();  *//显示* System.***out***.println(**"编号\t年级名称"**);  list.forEach(grade -> {  System.***out***.println(grade.getGid()+**"\t"**+grade.getGname());  }); } |

其它：

在mybatis工程运行时: Could not find resource com/kgc/dao/GradeMapper.xml

解决：maven编译|打包工程时，默认不支持将java目录中的配置文件进行打包，所以就会报找不到文件

在pom文件中添加以下代码

|  |
| --- |
| <**build**>  *<!--配置相关的资源进行打包-->* <**resources**>  <**resource**>  <**directory**>src/main/java</**directory**>  <**includes**>  <**include**>\*\*/\*.xml</**include**>  </**includes**>  </**resource**>  <**resource**>  <**directory**>src/main/resources</**directory**>  <**includes**>  <**include**>\*\*/\*.\*</**include**>  </**includes**>  </**resource**>  </**resources**> </**build**>  注意:build放到</dependencies>标签的后面 |