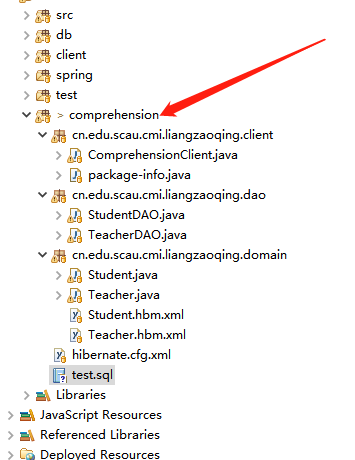
# 实验七：Hibernate框架实验参考

说明：

1. 实验参考的代码在comprehension源文件夹中。数据库有2张表，一个是student，一个是teacher，数据库的建表语句是test.sql，数据库名称是test。
2. 参考使用的是基于xml配置的情况，如果有同学使用注解方式完成的，也没问题。
3. 我只做了控制台客户端，并且只做了添加学生的功能，请同学们用JavaFX客户端完成，当然有同学说要用jsp完成，也可以。
4. 配置文件中需要注意的地方我用红色的字体，当然其他地方都需要注意。
5. **可能需要多次调试才能成功，祝大家顺利。**



# Mybatis的数据库配置文件

SqlMapConfig.xml

|  |
| --- |
| *<?***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**>  *<!-- 配置环境（使用的是mysql数据库）-->* <**environments default="mysql"**>  <**environment id="mysql"**>  *<!-- 配置事务管理类型-->* <**transactionManager type="JDBC"**></**transactionManager**>  *<!--配置数据源（连接池） -->* <**dataSource type="POOLED"**>  *<!--配置连接数据库的四个基本信息 -->* <**property name="driver" value="com.mysql.jdbc.Driver"**/>  <**property name="url" value="jdbc:mysql://localhost:3306/mybatisdemo?characterEncoding=UTF-8"**/>  <**property name="username" value="root"**/>  <**property name="password" value="123456"**/>  </**dataSource**>  </**environment**>  </**environments**>   *<!-- 告诉Mybatis区哪找持久化类的映射文件-->* <**mappers**>  <**mapper resource="cn/edu/scau/cmi/chenkairong/mybatisdemo/mapper/studentMapper.xml"**/>  <**mapper resource="cn/edu/scau/cmi/chenkairong/mybatisdemo/mapper/teacherMapper.xml"**/>  <**mapper resource="cn/edu/scau/cmi/chenkairong/mybatisdemo/mapper/advisorMapper.xml"**/>  </**mappers**> </**configuration**> |

# Mybatis的ORM映射文件

studentMapper.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"** *?>* **<!DOCTYPE mapper  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd"*>*** <**mapper namespace="cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.studentDao"**>   <**select id="getAll" resultType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Student"**>  select *\** from student  </**select**>   <**insert id="add" parameterType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Student"**>  insert into student(id,name) values (#{id},#{name})  </**insert**>   <**update id="update" parameterType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Student"**>  update student set id = #{id},name = #{name} where id = #{id}  </**update**>   <**delete id="delete" parameterType="java.lang.Integer"**>  delete from student where id = #{id}  </**delete**> </**mapper**> |

teacherMapper.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* **<!DOCTYPE mapper  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd"*>*** <**mapper namespace="cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.teacherDao"**>   *<!--查询所有-->  <!--resultType用来指定封装类-->* <**select id="getAll" resultType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Teacher"**>  select *\** from teacher  </**select**>  *<!--插入一组数据-->* <**insert id="add" parameterType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Teacher"**>  insert into teacher(id,name) values (#{id},#{name})  </**insert**>  *<!--修改一组数据-->* <**update id="update" parameterType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Teacher"**>  update teacher set name=#{name} where id=#{id}  </**update**>  *<!-- 删除一组数据-->* <**delete id="delete" parameterType="java.lang.Integer"**>  delete from teacher where id=#{id}  </**delete**> </**mapper**> |

AdvisorMapper.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"** *?>* **<!DOCTYPE mapper  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd"*>*** <**mapper namespace="cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.advisorDao"**>   <**select id="getAll" resultType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Advisor"**>  select *\** from advisor  </**select**>   <**insert id="add" parameterType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Advisor"**>  insert into advisor(student\_id,teacher\_id) values (#{student\_id},#{teacher\_id})  </**insert**>   <**update id="update" parameterType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Advisor"**>  update advisor set student\_id = #{student\_id},teacher\_id = #{teacher\_id} where student\_id = #{student\_id} or teacher\_id = #{teacher\_id}  </**update**>   <**delete id="delete" parameterType="cn.edu.scau.cmi.chenkairong.mybatisdemo.domain.Advisor"**>  delete from advisor where student\_id = #{student\_id} or teacher\_id = #{teacher\_id}  </**delete**> </**mapper**> |

# 实体类

Student.java

|  |
| --- |
| 1. **package** cn.edu.scau.cmi.chenkairong.mybatisdemo.domain;  **public class** Student {  **private int id**;  **private** String **name**;  *//private String teacher\_id;* **public** Student() {  }  **public** Student(**int** id, String name) {  **super**();  **this**.**id** = id;  **this**.**name** = name;  *//this.teacher\_id = teacher\_id;* }  **public int** getId() {  **return id**;  }  **public void** setId(**int** id) {  **this**.**id** = id;  }  **public** String getName() {  **return name**;  }  **public void** setName(String name) {  **this**.**name** = name;  }  @Override  **public** String toString() {  **return "Student [id="** + **id** + **", name="** + **name** + **"]"**;  }  } |

Teacher.java

|  |
| --- |
| **package** cn.edu.scau.cmi.chenkairong.mybatisdemo.domain;  **public class** Teacher {  **private int id**;  **private** String **name**;  **public** Teacher() {  }  **public** Teacher(**int** id, String name) {  **super**();  **this**.**id** = id;  **this**.**name** = name;  }  **public int** getId() {  **return id**;  }  **public void** setId(**int** id) {  **this**.**id** = id;  }  **public** String getName() {  **return name**;  }  **public void** setName(String name) {  **this**.**name** = name;  }  @Override  **public** String toString() {  **return "Teacher [id="** + **id** + **", name="** + **name** + **"]"**;  }  } |

Advisor.java

|  |
| --- |
| **package** cn.edu.scau.cmi.chenkairong.mybatisdemo.domain;  **public class** Advisor {  **private int student\_id**;  **private int teacher\_id**;   **public** Advisor(**int** student\_id, **int** teacher\_id){  **this**.**student\_id** = student\_id;  **this**.**teacher\_id** = teacher\_id;  }   **public int** getStudent\_id() {  **return student\_id**;  }   **public void** setStudent\_id(**int** student\_id) {  **this**.**student\_id** = student\_id;  }   **public int** getTeacher\_id() {  **return teacher\_id**;  }   **public void** setTeacher\_id(**int** teacher\_id) {  **this**.**teacher\_id** = teacher\_id;  }   @Override  **public** String toString() {  **return "Advisor{"** +  **"student\_id="** + **student\_id** +  **", teacher\_id="** + **teacher\_id** +  **'}'**;  } } |

# 客户端界面类

mybatisController.java

|  |
| --- |
| *//读取配置文件* InputStream **is** = Resources.*getResourceAsStream*(**"SqlMapConfig.xml"**); *//初始化mybatis，创建SqlSessionFactory类实例* SqlSessionFactory **sqlSessionFactory** = **new** SqlSessionFactoryBuilder().build(**is**); |
| **public void** addStudent(){  **int** id = Integer.*valueOf*(**student\_id**.getText());  String name = **student\_name**.getText();  Student student = **new** Student(id, name);  SqlSession session = **sqlSessionFactory**.openSession();  session.insert(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.studentDao.add"**,student);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"学生信息成功增加"**); } |
| **public void** addTeacher(){  **int** id = Integer.*valueOf*(**teacher\_id**.getText());  String name = **teacher\_name**.getText();  Teacher teacher = **new** Teacher(id, name);  SqlSession session = **sqlSessionFactory**.openSession();  session.insert(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.teacherDao.add"**,teacher);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"教师信息成功增加"**); } |
| **public void** addAdvisor(){  **int** sid = Integer.*valueOf*(**relation\_sid**.getText());  **int** tid = Integer.*valueOf*(**relation\_tid**.getText());  Advisor advisor = **new** Advisor(sid, tid);  SqlSession session = **sqlSessionFactory**.openSession();  session.insert(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.advisorDao.add"**,advisor);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"关系信息成功增加"**); } |
| **public void** updateStudent(){  **int** id = Integer.*valueOf*(**student\_id**.getText());  String name = **student\_name**.getText();  Student student = **new** Student(id, name);  SqlSession session = **sqlSessionFactory**.openSession();  session.update(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.studentDao.update"**,student);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"学生信息成功修改"**); } |
| **public void** updateTeacher(){  **int** id = Integer.*valueOf*(**teacher\_id**.getText());  String name = **teacher\_name**.getText();  Teacher teacher = **new** Teacher(id, name);  SqlSession session = **sqlSessionFactory**.openSession();  session.update(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.teacherDao.update"**,teacher);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"教师信息成功修改"**); } |
| **public void** updateAdvisor(){  **int** sid = Integer.*valueOf*(**relation\_sid**.getText());  **int** tid = Integer.*valueOf*(**relation\_tid**.getText());  Advisor advisor = **new** Advisor(sid, tid);  SqlSession session = **sqlSessionFactory**.openSession();  session.update(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.advisorDao.update"**,advisor);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"关系信息成功修改"**); } |
| **public void** deleteStudent(){  **int** id = Integer.*valueOf*(**student\_id1**.getText());  SqlSession session = **sqlSessionFactory**.openSession();  session.delete(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.studentDao.delete"**,id);  *//删除对应的关系student* Advisor advisor = **new** Advisor(id, 0);  session.delete(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.advisorDao.delete"**,advisor);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"学生信息成功删除"**); } |
| **public void** deleteTeacher(){  **int** id = Integer.*valueOf*(**teacher\_id1**.getText());  SqlSession session = **sqlSessionFactory**.openSession();  session.delete(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.teacherDao.delete"**,id);  *//删除对应的关系student* Advisor advisor = **new** Advisor(0, id);  session.delete(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.advisorDao.delete"**,advisor);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"教师信息成功删除"**); } |
| **public void** deleteAdvisor(){  **int** sid = Integer.*valueOf*(**relation\_sid1**.getText());  **int** tid = Integer.*valueOf*(**relation\_tid1**.getText());  Advisor advisor = **new** Advisor(sid, tid);  SqlSession session = **sqlSessionFactory**.openSession();  session.delete(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.advisorDao.delete"**,advisor);  *//提交事务* session.commit();  *//关闭Session* session.close();  **information**.setText(**"关系信息成功删除"**); } |
| **public void** allStudent(){  SqlSession session = **sqlSessionFactory**.openSession();  List<Student> stuList= session.selectList(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.studentDao.getAll"**);  *//提交事务* session.commit();  *//关闭Session* session.close();   StringBuilder stringBuilder = **new** StringBuilder();  stringBuilder.append(**"所有学生信息如下：\n"**);  **for**(Student stu:stuList){  stringBuilder.append(stu.getId() + **" "** + stu.getName() + **"\n"**);  }  **information**.setText(stringBuilder.toString()); } |
| **public void** allTeacher(){  SqlSession session = **sqlSessionFactory**.openSession();  List<Teacher> teachers= session.selectList(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.teacherDao.getAll"**);  *//提交事务* session.commit();  *//关闭Session* session.close();   StringBuilder stringBuilder = **new** StringBuilder();  stringBuilder.append(**"所有教师信息如下：\n"**);  **for**(Teacher teacher:teachers){  stringBuilder.append(teacher.getId() + **" "** + teacher.getName() + **"\n"**);  }  **information**.setText(stringBuilder.toString()); } |
| **public void** allRelation(){  SqlSession session = **sqlSessionFactory**.openSession();  List<Advisor> advisors= session.selectList(**"cn.edu.scau.cmi.chenkairong.mybatisdemo.dao.advisorDao.getAll"**);  *//提交事务* session.commit();  *//关闭Session* session.close();   StringBuilder stringBuilder = **new** StringBuilder();  stringBuilder.append(**"所有关系信息如下：\n"**);  **for**(Advisor advisor:advisors){  stringBuilder.append(advisor.getStudent\_id() + **" "** + advisor.getTeacher\_id() + **"\n"**);  }  **information**.setText(stringBuilder.toString()); } |

# DAO类

|  |
| --- |
| **public interface** advisorDao {  **public** List<Advisor> getAll();   **public void** add(Advisor advisor);   **public void** update(Advisor advisor);   **public void** delete(Advisor advisor); } |
| **public interface** studentDao {  **public** List<Student> getAll();   **public void** add(Student student);   **public void** update(Student student);   **public void** delete(String id); } |
| **public interface** teacherDao {  **public** List<Teacher> getAll();   **public void** add(Teacher teacher);   **public void** update(Teacher teacher);   **public void** delete(String id); } |

# 运行截图

