实体类在各层之间实现数据传递

引言:

1. 使用java存储一个学生信息（包括：xh,name,age）

1.1创建学生类

Public class Student{ public int xh; public String name; public int age;}

1.2创建学生类的对象

Student stu=new Student();

stu.xh=101; stu.name=”张三”; stu.age=21;

1. 使用数据库的表存储学生信息（包括：xh,name,age）

2.1创建学生表

Create table Student(

Xh int, name varchar(15), age int

)

* 1. 存储数据

学号 姓名 年龄

101 张三 21 一行记录

102 李四 21 一行记录

表中一行记录对应一个类的对象，表中多行记录对应类的泛型集合.

1. 实休类
2. 什么是实体类:

实体类本质就是一个类，类和表对应.

注意:

1. 实体类名与表名对应
2. 实休类属性（私有属性，进行封装）与字段对应

1. 作用:

实现各层之间数据的传递.

1. 创建实体类

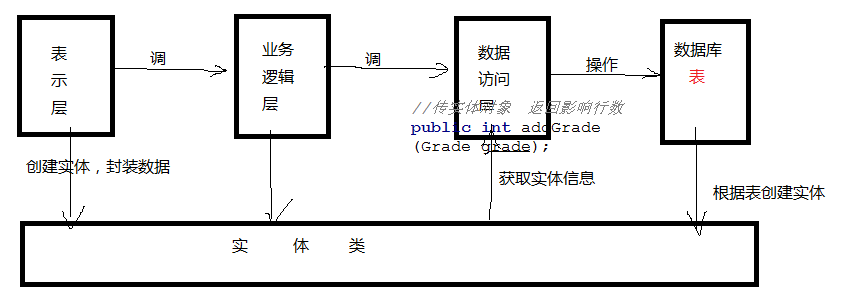
注意:实体类都放在entity包中

|  |
| --- |
| *//实体类 与表对应* **public class** Grade {  *//创建属性* **private int gid**; *//年级编号* **private** String **gname**; *//年级名称* **public int** getGid() {  **return gid**;  }   **public void** setGid(**int** gid) {  **this**.**gid** = gid;  }   **public** String getGname() {  **return gname**;  }   **public void** setGname(String gname) {  **this**.**gname** = gname;  } } |

1. 数据访问层总结(Dao):背
2. 一个表一个实体一个接口一个实现类
3. 一个功能(业务)一方法

*//添加 传实体对象 返回影响行数***public int** 方法名(实体类 名称);  
  
*//修改 传实体对象 返回影响行数***public int**方法名(实体类 名称);  
  
*//删除年级 传主键类型 返回影响行数***public int** 方法名(**主键类型** 名称);  
  
*//查询所有 返回实休集合 传查询条件***public** List<实体类>方法名([传查询条件]);  
  
*//查询单个 返回实体 传主键类型***public** 实体类 方法名(**主键类型** 名称);

三、使用实体类



1. 添加年级

1.1 编写dao层:

接口:

|  |
| --- |
| *//添加年级 //传实体对象 返回影响行数* **public int** addGrade(Grade grade); |

实现:

|  |
| --- |
| @Override **public int** addGrade(Grade grade) {  String sql=**"insert into grade(gid,gname) values(?,?)"**;  **return this**.executeUpdate(sql,**new** Object[]{grade.getGid(),grade.getGname()}); } |

* 1. 编写业务层

接口:

|  |
| --- |
| **public int** addGrade(Grade grade); |

实现:

|  |
| --- |
| @Override **public int** addGrade(Grade grade) {  *//调用数据访问层* GradeDao gradeDao=**new** GradeDaoImpl();  **return** gradeDao.addGrade(grade); } |

* 1. 编写表示层

|  |
| --- |
| **public static void** main(String[] args) {  *//添加年级* Scanner input=**new** Scanner(System.***in***);  System.***out***.println(**"请输入编号:"**);  **int** gid=input.nextInt();  System.***out***.println(**"请输入名称:"**);  String gname=input.next();   *//创建年级对象* Grade grade=**new** Grade(); *//一行* grade.setGid(gid);  grade.setGname(gname);   *//添加  //调用业务层实现添加* GradeService gradeService=**new** GradeServiceImpl();  **int** temp= gradeService.addGrade(grade);  **if**(temp>0)  System.***out***.println(**"添加成功"**);  **else** System.***out***.println(**"添加失败"**); } |

1. 修改年级

2.1编写dao层

接口:

|  |
| --- |
| *//修改年级 传实体对象 返回影响行数*  **public int** updateGrade(Grade grade); |

实现:

|  |
| --- |
| @Override **public int** updateGrade(Grade grade) {  String sql=**"update grade set gname=? where gid=?"**;  **return this**.executeUpdate(sql,**new** Object[]{grade.getGname(),grade.getGid()}); } |

2.2编写业务层

接口:

**public int** updateGrade(Grade grade);

实现:

@Override  
**public int** updateGrade(Grade grade) {  
 *//调用数据访问层* **return gradeDao**.updateGrade(grade);  
}

2.3编写表示层

|  |
| --- |
| **public static void** main(String[] args) {  *//添加年级* Scanner input=**new** Scanner(System.***in***);  System.***out***.println(**"请输入编号:"**);  **int** gid=input.nextInt();  System.***out***.println(**"请输入名称:"**);  String gname=input.next();   *//创建年级对象* Grade grade=**new** Grade(); *//一行* grade.setGid(gid);  grade.setGname(gname);  *//添加  //调用业务层实现添加* GradeService gradeService=**new** GradeServiceImpl();  **int** temp= gradeService.updateGrade(grade);  **if**(temp>0)  System.***out***.println(**"修改成功"**);  **else** System.***out***.println(**"修改失败"**); } |

1. 删除年级

3.1编写dao

接口:

*//删除年级 传主键类型 返回影响行数*

**public int** deleteGrade(**int** gid);

实现:

|  |
| --- |
| @Override **public int** deleteGrade(**int** gid) {  String sql=**"delete from grade where gid=?"**;  **return this**.executeUpdate(sql,**new** Object[]{gid}); } |

3.2编写业务

接口:

|  |
| --- |
| @Override **public int** deleteGrade(**int** gid); |

实现:

|  |
| --- |
| @Override **public int** deleteGrade(**int** gid) {  **return gradeDao**.deleteGrade(gid); } |

3.3编写表示层

|  |
| --- |
| **public static void** main(String[] args) {  *//删除年级* Scanner input=**new** Scanner(System.***in***);  System.***out***.println(**"请输入编号:"**);  **int** gid=input.nextInt();   *//调用业务层实现添加* GradeService gradeService=**new** GradeServiceImpl();  **int** temp= gradeService.deleteGrade(gid);  **if**(temp>0)  System.***out***.println(**"删除成功"**);  **else** System.***out***.println(**"删除失败"**); } |

1. 查询所有年级

4.1 编写dao层

接口:

|  |
| --- |
| *//查询所有年级* **public** List<Grade> getAllGrade(); |

实现:

|  |
| --- |
| @Override **public** List<Grade> getAllGrade() {  List<Grade> list=**null**;  **try** {  String sql=**"select \* from grade"**;  ResultSet rs=**this**.executeQuery(sql,**null**);  *//将结果集转化为实体集合* list=**new** ArrayList<>();  **while**(rs.next()){  *//循环一次就是一行* Grade grade=**new** Grade();  grade.setGid(rs.getInt(1));  grade.setGname(rs.getString(2));  *//将对象添加到集合* list.add(grade);  }  } **catch** (SQLException e) {  e.printStackTrace();  }**finally** {  **this**.closeAll(); *//关闭资源* }  **return** list; } |

* 1. 编写业务

接口:

|  |
| --- |
| @Override **public** List<Grade> getAllGrade() ; |

实现:

|  |
| --- |
| @Override **public** List<Grade> getAllGrade() {  **return gradeDao**.getAllGrade(); } |

4.3 编写表示层

|  |
| --- |
| **public static void** main(String[] args) {  *//显示所有年级   //调用业务层* GradeService gradeService=**new** GradeServiceImpl();  List<Grade> list=gradeService.getAllGrade();   *//3.获取结果进行展示* System.***out***.println(**"编号\t名称"**);  list.forEach(grade -> {  System.***out***.println(grade.getGid()+**"\t"**+grade.getGname());  }); } |

1. 查询单条年级信息

5.1 编写dao层

接口:

|  |
| --- |
| *//查询单个年级 返回实体 传主键类型* **public** Grade getGrade(**int** gid); |

实现:

|  |
| --- |
| @Override **public** Grade getGrade(**int** gid) {  String sql=**"select \* from grade where gid=?"**;  Grade grade=**null**;  **try** {  ResultSet rs=**this**.executeQuery(sql,**new** Object[]{gid});  *//将结果集转化为实体集合* **if**(rs.next()){  *//循环一次就是一行* grade=**new** Grade();  grade.setGid(rs.getInt(1));  grade.setGname(rs.getString(2));  }  } **catch** (SQLException e) {  e.printStackTrace();  }**finally** {  **this**.closeAll(); *//关闭资源* }  **return** grade; } |

5.2编写业务层

接口:

|  |
| --- |
| **public** Grade getGrade(**int** gid); |

实现:

|  |
| --- |
| @Override **public** Grade getGrade(**int** gid) {  **return gradeDao**.getGrade(gid); } |

5.3编写表示层

|  |
| --- |
| **public static void** main(String[] args) {  Scanner input=**new** Scanner(System.***in***);  System.***out***.println(**"请输入编号:"**);  **int** gid=input.nextInt();   *//调用业务层* GradeService gradeService=**new** GradeServiceImpl();  Grade g=gradeService.getGrade(gid);   **if**(g==**null**)  System.***out***.println(**"没有记录"**);  **else** {  System.***out***.println(**"编号:"** + g.getGid());  System.***out***.println(**"名称:"**+g.getGname());  } } |

作业:

1. 理解实体类与三层架构的关系:看图
2. 背诵数据访问层总结:

数据访问层总结(Dao):背

1. 一个表一个实体一个接口一个实现类
2. 一个功能(业务)一方法

*//添加 传实体对象 返回影响行数***public int** 方法名(实体类 名称);  
  
*//修改 传实体对象 返回影响行数***public int**方法名(实体类 名称);  
  
*//删除年级 传主键类型 返回影响行数***public int** 方法名(**主键类型** 名称);  
  
*//查询所有 返回实休集合 传查询条件***public** List<实体类>方法名([传查询条件]);  
  
*//查询单个 返回实体 传主键类型***public** 实体类 方法名(**主键类型** 名称);

1. 实现增，删，改，查，查单条一套全做完理解
2. 通过查询所有学生

选做: 通过地址查询所有学生