# 实验二：ORM设计模式和DAO模式

# 项目建立

说明：如果同学们提交的实验一如果自己觉得复杂，可以裁剪，规模适当减小，另外，所有的黄色的内容根据情况修改或者删除。

使用eclipse建立一个普通的Java应用项目，项目名称规范是学号姓名。例如（201530690101张山），注意学号和姓名之间不要留空格。

建立包，包名规范是cn.edu.scau.cmi.myname，其中myname是你的中文名称的全名，例如cn.edu.scau.cmi.zhangshan，本课程的所有Java类都必须放置再这个包或者这个包的子包下。例如，实体类放置到cn.edu.scau.cmi.zhangshan.domain，DAO类放置到cn.edu.scau.cmi.zhangshan.dao，工具类放到cn.edu.scau.cmi.zhangshan.util。

# 实体类、DAO类、工具类、客户端类的设计

客户类、实体类、DAO类、数据库访问类的UML设计



图1 系统相关类图

# 实体类设计

根据实验一建立的需求和数据库，设计相应的实体类。建立的实体类放置到cn.edu.scau.cmi.zhangshan.domain.

将设计的实体类代码复制到下面的代码框中。

//Student类

package cn.edu.scau.cmi.zhouxudong.domain;

public class Student {

private String StuID;

private String Name;

private String Sex;

private String Grade;

private String Academy;

public String getStuID() {

return StuID;

}

public void setStuID(String stuID) {

StuID = stuID;

}

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public String getSex() {

return Sex;

}

public void setSex(String sex) {

Sex = sex;

}

public String getGrade() {

return Grade;

}

public void setGrade(String grade) {

Grade = grade;

}

public String getAcademy() {

return Academy;

}

public void setAcademy(String academy) {

Academy = academy;

}

}

//Book类

package cn.edu.scau.cmi.zhouxudong.domain;

public class Book {

private String BookID;

private String Title;

private String Author;

private String Publisher;

private String PublishDate;

private String Kind;

private String Status;

public String getBookID() {

return BookID;

}

public void setBookID(String bookID) {

BookID = bookID;

}

public String getTitle() {

return Title;

}

public void setTitle(String title) {

Title = title;

}

public String getAuthor() {

return Author;

}

public void setAuthor(String author) {

Author = author;

}

public String getPublisher() {

return Publisher;

}

public void setPublisher(String publisher) {

Publisher = publisher;

}

public String getPublishDate() {

return PublishDate;

}

public void setPublishDate(String publishDate) {

PublishDate = publishDate;

}

public String getKind() {

return Kind;

}

public void setKind(String kind) {

Kind = kind;

}

public String getStatus() {

return Status;

}

public void setStatus(String status) {

Status = status;

}

}

# DAO类设计

根据实验一建立的需求和数据库，设计相应的DAO类。建立DAO类放置到cn.edu.scau.cmi.zhangshan.dao。将设计的DAO类代码复制到下面的代码框中。

//StudentDao

package cn.edu.scau.cmi.zhouxudong.dao;

import cn.edu.scau.cmi.zhouxudong.domain.\*;

public interface StudentDao {

void newStudent(Student student);

void deleteStudent(Student student);

void findStudent(Student student) ;

void updateStudent(Student student) ;

}

//BookDao

package cn.edu.scau.cmi.zhouxudong.dao;

import cn.edu.scau.cmi.zhouxudong.domain.\*;

public interface BookDao {

void newBook(Book book);

void deleteBook(Book book);

void findBook(Book book);

void updateBook(Book book);

}

# 数据库访问的公共类设计

根据课堂所讲的思想，设计相应的数据库防伪的公共类。将设计的类代码复制到下面的代码框中。

package cn.edu.scau.cmi.zhouxudong.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConn {

private static final String url = "jdbc:mysql://localhost:3306/test"; //数据库地址

private static final String username = "zxd"; //数据库用户名

private static final String password = "123456"; //数据库密码

private static final String driver = "com.mysql.jdbc.Driver"; //mysql驱动

private static final Connection conn = null;

/\*\*

\* 连接数据库

\* @return

\*/

public static Connection conn() {

Connection conn = null;

try {

Class.forName(driver); //加载数据库驱动

try {

conn = DriverManager.getConnection(url, username, password); //连接数据库

} catch (SQLException e) {

e.printStackTrace();

}

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

return conn;

}

/\*\*

\* 关闭数据库链接

\* @return

\*/

public static void close() {

if(conn != null) {

try {

conn.close(); //关闭数据库链接

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

//find 增删改查

package cn.edu.scau.cmi.zhouxudong.util;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class DBUtil {

private static Connection conn = null;

private static PreparedStatement ps = null;

private static ResultSet rs = null;

public static void main(String[] args) {

conn = DBConn.conn(); //调用 DBconnection 类的 conn() 方法连接数据库

String sql = "select \* from student "; //sql语句

try {

Statement stmt =(Statement) conn.createStatement();

ResultSet rs = stmt.executeQuery(sql);

while(rs.next()) {

String StuID=rs.getString(1);

String Name=rs.getString(2);

String Sex=rs.getString(3);

String Grade=rs.getString(4);

String Academy=rs.getString(5);

System.out.println(StuID+" "+Name+" "+Sex+" "+Grade+" "+Academy);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

DBConn.close();

}

}

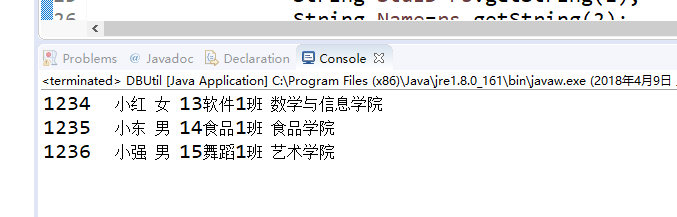
# 客户端设计

将客户端代码复制到下面的框中

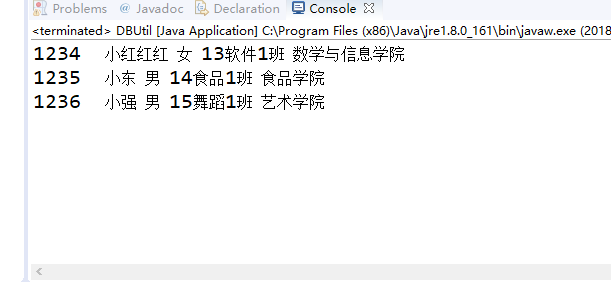
//没有客户端...

# 运行效果

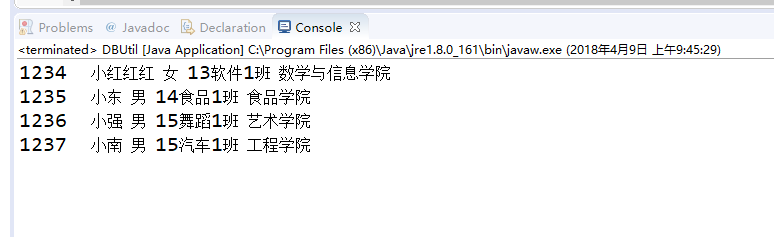
查询



更新



插入



删除

