河南工业大学

《JAVA 程序设计》实验报告

专业班级：物联网 1603　　学号：201616070320　　姓名：郭治洪

实验单元四　综合性程序设计

实验六 简单学生信息管理系统

实验时间：2018.12

【实验目的】

1、了解 JDBC 的作用，中我通过 JDBC 访问数据库的方法。

2、能够实现对数据库中数据的添加、删除、修改和查询。

【实验环境】

JDK、Eclipse

【实验内容】

教材 P288：实验内容

【详细分析】

我使用了五个类完成了该题要求的动作。

公共类 Main 类里有主方法，这是程序的入口。在 Main 方法中实例化其他四个类并且调用。

ConnectToDatabase 类里有初始化连接数据库的相关方法，完成后将可以得到实例化的 Connection 静态变量 conn 供 Student 类操作数据库使用；该类中还有类似于 C++析构函数一样 close()方法关闭实例化的 Connection 静态变量 conn；还有未做好的自动检查数据库和创建数据库的方法，但是并未成功。

Show 类是显示菜单类，所有的菜单输出都是在 String show[]数组内，并且调用循环输出 String 数组所有内容，以及调用 length 方法对选项个数进行统计。

Input 类是输入类，包含输入数据库的所有信息学号，姓名，分数的静态成员变量以及相关输入的方法，所有的输入方法都含有正则表达式匹配确保输入

的范围正确；以及含有对用户菜单选项输入处理的方法，对用户使用升序或者降序排列成绩的输入处理的方法，对用户输入确认和取消的输入处理的方法，也使用正则表达式完成校验，这两个方法返回值都可以用静态的成员变量得到；当然为了保证程序工作正常，还有将所有静态成员变量重置的方法；输入器使用的 Scanner 类，也包括构造和 close()方法。

Student 类学生类，是处理对数据库操作类，将 ConnectToDatabase 类实例化的 Connection 静态变量 conn 传入用来对数据库进行操作。该类中包括列取数据库所有学生信息方法，通过学号查找方法，通过姓名查找方法，按成绩正序或者倒叙排序方法（让用户选择输入，Input 类的方法和成员变量负责处理和传入）；以及通过学生学号删除信息方法，该方法先调用通过学号查找方法判断学号是否存在，存在再尝试删除；还有添加（修改）学生信息的方法，通过 Input 类的方法和成员变量负责处理和传入输入学生信息，使用传入的信息按照学号调用通过学号查找方法判断用户是否存在，不存在则直接添加进数据库，存在比对姓名和成绩是否同原来数据库不同，询问用户是否要更新，如果输入同原来信息相同还会要求重输姓名和成绩。对数据库的所有操作都是传入的静态变量 conn 新建的私有类成员变量 preparedStatement 实现；该类还包括构造和 close() 方法。

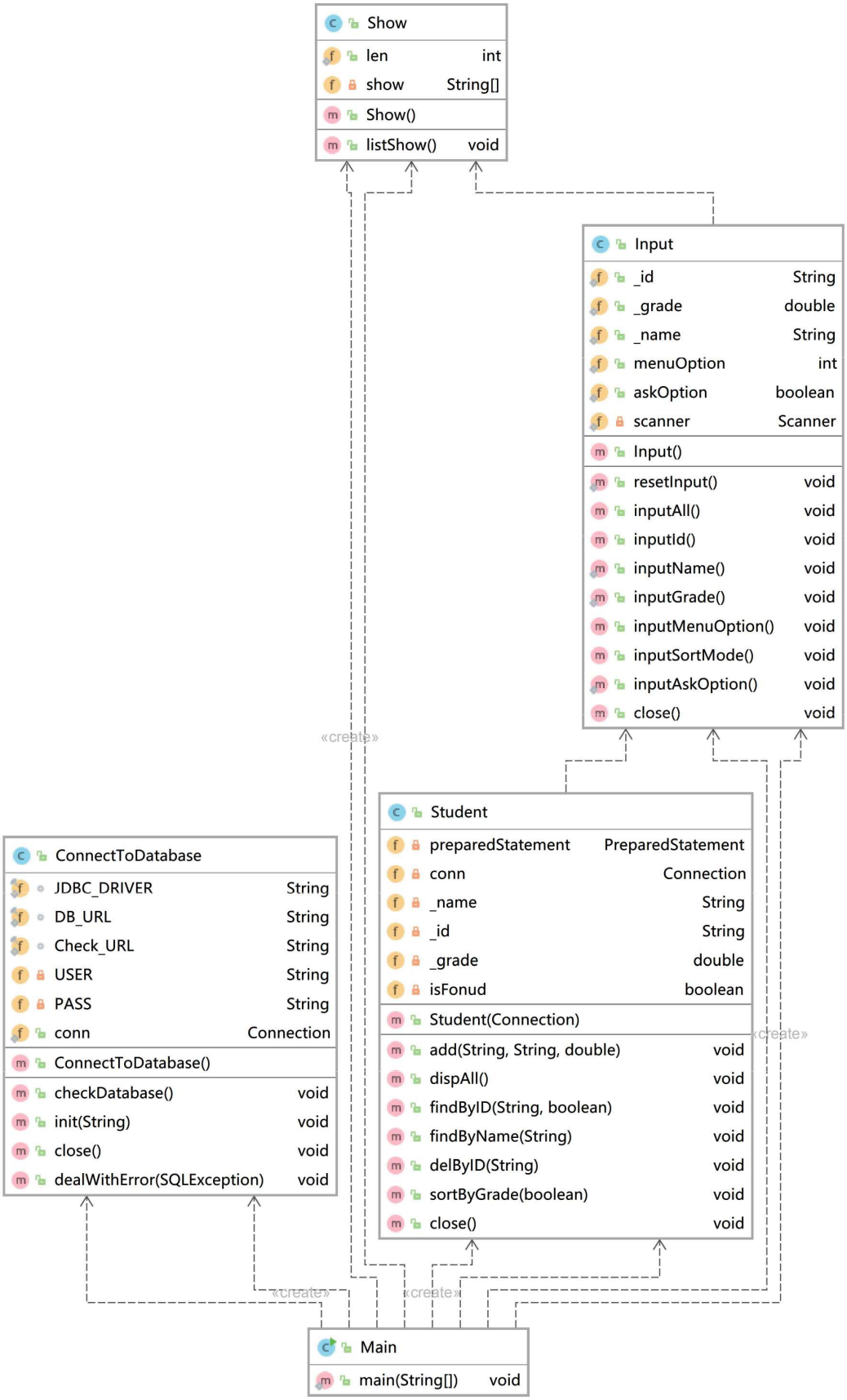
以上的四个类都包含构造方法，用以实例化构造相关成员变量和成员方法。

Main 类的主方法通过调用 Input 的输出方法输出菜单，Input 类得到检验和用户的输入，Student 类对数据库进行相关操作，ConnectToDatabase 进行数据库的连接初始化。

但是也有很多不足，比如并没有使用重写重载方法，没有使用 UI 实现，没有自定义异常，没有使用接口，没有使用继承，没有自动创建数据库和数据表的功能，没有更人性化的数据处理，功能和数据结构过于简单，注释写的过于简单和不够标准，程序逻辑仍然不够清晰，相比于去年我们学长学姐的课程作业，我感觉还差一大截。

因为我们学长姚东阳和其他计科学长以及我的朋友都说 SQL SERVRR 很难与 JDBC 一起使用，所以，我们都选择了 MYSQL，此外感谢他们对我的帮助。

下页是使用 IDEA 工具自动生成的 UML 图。


Show
len int
show String[]
Show()
listShow() void
Input
_id String
_grade double
_name String
menuOption int
askOption boolean
scanner Scanner
Input()
resetInput() void
inputAll() void
inputId() void
inputName() void
inputGrade() void
inputMenuOption() void
inputSortMode() void
inputAskOption() void
close() void
«create»
Student
preparedStatement PreparedStatement
conn Connection
_name String
_id String
_grade double
isFonud boolean
Student(Connection)
add(String, String, double) void
dispAll() void
findByID(String, boolean) void
findByName(String) void
delByID(String) void
sortByGrade(boolean) void
close() void
«create»
ConnectToDatabase
JDBC_DRIVER String
DB_URL String
Check_URL String
USER String
PASS String
conn Connection
ConnectToDatabase()
checkDatabase() void
init(String) void
close() void
dealWithError(SQLException) void
«create»
«create»
Main
main(String[]) void

【实验源码】

由于没有自动创建数据库功能，首先要创建一个数据库。

```
--CREATE DATABASE AND TABLE SQL
--为了简单方便，使用MYSQL数据库
CREATE DATABASE Student;
USE Student;
CREATE TABLE `Student` (
    `id` varchar(20) DEFAULT '' NOT NULL COMMENT 'id',
    `name` char(20) NOT NULL DEFAULT '' COMMENT 'name',
    `grade` float NOT NULL DEFAULT 0.0 COMMENT 'grade',
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

这是 Main 类，包含主方法，分别会将其他类实例化：

```
package JDBC;
//Main类，包括主方法，程序的入口

public class Main {
    public static void main(String args[]) {
        Show show=new Show();
        ConnectToDatabase connectToDatabase=new ConnectToDatabase();
        Student student=new Student(ConnectToDatabase.conn);
        Input input=new Input();
        while (true) {
            show.listShow();
            Input.resetInput();
            input.inputMenuOption();

System.out.println("-------------------------------------");
            if(Input.menuOption==0) {
                student.dispAll();
            } else if(Input.menuOption==1) {
                Input.resetInput();
                input.inputAll();
                student.add(Input._id,Input._name,Input._grade);
            } else if(Input.menuOption==2) {
                input.inputId();
                //System.out.println(Input._id);
                student.findByID(Input._id,false);
```

```
            } else if(Input.menuOption==3) {
                input.inputName();
                //System.out.println(Input._name);
                student.findByName(Input._name);
            } else if(Input.menuOption==4) {
                input.inputId();
                student.delByID(Input._id);
            } else if(Input.menuOption==5) {
                input.inputSortMode();
                student.sortByGrade(Input.askOption);
            } else if(Input.menuOption==6) {
                input.close();
                student.close();
                connectToDatabase.close();
                System.exit(0);
            } else {
                System.out.println("Input error!");

System.out.println("-------------------------------------");
            }
        }
    }
}
```

这是 **ConnectToDatabase** 类，负责连接数据库等相关操作。

```
package JDBC;
//ConnectToDatabase类，负责连接数据库等相关操作
import java.sql.*;

public class ConnectToDatabase {
    static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    static final String DB_URL =
"jdbc:mysql://localhost:3306/Student?useSSL=false&serverTimezone=UTC&
allowPublicKeyRetrieval=true";
    static final String
Check_URL="jdbc:mysql://localhost:3306/INFORMATION_SCHEMA?useSSL=fals
e&serverTimezone=UTC&allowPublicKeyRetrieval=true";
    private String USER = "root";
    private String PASS = "iFTJw2E";
    public static Connection conn = null;
    //public Statement stmt = null; //网上说为了提升效能不再使用这种方法
    public ConnectToDatabase () {
```

```
        super();
        init(Check_URL);
        checkDatabase();
        init(DB_URL);
    }
    public void checkDatabase() {
        //see
here:https://stackoverflow.com/questions/838978/how-to-check-if-mysql
-database-exists
        String sql="SELECT SCHEMA_NAME FROM INFORMATION_SCHEMA.SCHEMATA
WHERE SCHEMA_NAME = \'student\'";
            try {
            PreparedStatement
preparedStatement=conn.prepareStatement(sql);
            ResultSet rs=preparedStatement.executeQuery(); //本方法适用与
查询语句
            if(!rs.next()){
                //无法自动建库，请手动建库。
                System.out.println("数据库不存在，请手动建库");
                close();
                System.exit(-1);
//                String create="CREATE DATABASE Student;\n" +
//                        "USE Student;\n" +
//                        "CREATE TABLE `Student` (\n" +
//                        "    `id` varchar(20) DEFAULT '' NOT NULL, \n" +
//                        "    `name` char(20) NOT NULL DEFAULT '' COMMENT
'name',\n" +
//                        "    `grade` float NOT NULL DEFAULT 0.0 COMMENT
'grade',\n" +
//                        "    PRIMARY KEY (`id`)\n" +
//                        ") ENGINE=InnoDB DEFAULT CHARSET=utf8";
//                preparedStatement = conn.prepareStatement(create);
//                preparedStatement.executeUpdate();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    public void init(final String database_URL) {
        try {
            Class.forName(JDBC_DRIVER);
            System.out.println("Connect to mysql server...");
            conn = DriverManager.getConnection(database_URL, USER, PASS);
            System.out.println("Create the statement object...");
```

```
        } catch (SQLException se) {
            dealWithError(se);
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(-1);
        }
    }

      public void close() {
//        try {
//            if (stmt != null)
//                stmt.close();
//        } catch (SQLException se2) {
//            dealWithError(se2);
//        }
        try {
            if (conn != null)
                conn.close();
        } catch (SQLException se) {
            dealWithError(se);
        }
    }

    public void dealWithError(SQLException se) {
        se.printStackTrace();
        System.exit(-1);
    }
}
```

下面是Show类，负责菜单的显示：

```
package JDBC;
//这是Show类，负责菜单显示。
public class Show {
    public static int len=0;
    private String show[]={
            "0.显示所有学生的信息",
            "1.添加（更新）学生信息",
            "2.按学号查找",
            "3.按姓名查找",
            "4.按学号删除",
            "5.按成绩排序",
            "6.退出"
```

```
    };
    public Show() {
        super();
        len=show.length;
    }
    public void listShow() {
        for(String output:show) {
            System.out.println(output);
        }
    }
}
```

下面是Input类，负责处理用户的输入：

```
package JDBC;

//Input类，负责处理用户的输入并判断时候存在问题

import java.util.Scanner;

public class Input {
    public static String _id="";
    public static double _grade=0.0;
    public static String _name="";
    public static int menuOption=0;
    public static boolean askOption=false;
    private static Scanner scanner=null;

    public static void resetInput() {
        Input._name="";
        Input._id="";
        Input._grade=0.0;
        Input.menuOption=0;
        Input.askOption=false;
    }

    public Input() {
        super();
        scanner=new Scanner(System.in);
    }
    public void inputAll() {
        Input.resetInput();
        inputId();
        Input.inputName();
```

```
        Input.inputGrade();
    }
    public void inputId() {
        Input._id="";
        String _ID;
        while(true) {
            System.out.println("Please input the student's ID:");
            _ID=scanner.next();
            if(_ID.matches("^[0-9]{1,12}") == false) {
                System.out.println("Input error,please input the correct
ID and retry again.");
            }
            else {
                this._id=_ID;
                break;
            }
        }
    }
     public static void inputName() {
        Input._name="";
        String _name;
        while(true) {
            System.out.println("Please input the student's name:");
            _name=scanner.next();

if(_name.matches("^[\\u4E00-\\u9FA5A-Za-z\\s]+(·[\\u4E00-\\u9FA5A-Za-
z]+)*$") == false) {
                System.out.println("Input error,please input the correct
name and retry again.");
            }
            else {
                Input._name=_name;
                break;
            }
        }
    }
    public static void inputGrade() {
        Input._grade=0.0;
        double _grade = 0.0;
        while(true) {
            System.out.println("Please input the student's grade:");
            _grade= scanner.nextDouble();
            if(_grade<0.0||_grade>100.0) {
                System.out.println("Input error,please input the correct
```

```
grade and retry again.");
            }
            else {
                Input._grade=_grade;
                break;
            }
        }
    }
    public void inputMenuOption() {
        menuOption=0;
        int len=Show.len-1;
        //System.out.println(len);
        String regex="^[0-9]+$";
        String input;
        while (true) {

System.out.println("-------------------------------------");
            System.out.println("Please input your option (0 to "+len+")");
            input=scanner.next();
            if(input.matches(regex)==false) {
                System.out.println("Input error,please input your option
and retry again.");
            } else {
                menuOption=Integer.parseInt(input);
                break;
            }
        }
    }
    public void inputSortMode() {
        askOption=false;
        String _askOption;
        while(true) {

//System.out.println("-------------------------------------");
            System.out.println("Please input 1 to ASC, input 2 to DESC:");
            _askOption=scanner.next();
            if(_askOption.matches("^[1,2]$") == false) {
                System.out.println("Input error,please input the your
option and retry again.");
            } else {
                int input=Integer.parseInt(_askOption);
                if(input==1)
                    askOption=true;
                else
```

```
                askOption=false;
                break;
            }
        }
    }
    public static void inputAskOption() {
        askOption=false;
        String _askOption;
        while(true) {

System.out.println("-------------------------------------");
            System.out.println("Please input 1 to OK, input 2 to Cancel:");
            _askOption=scanner.next();
            if(_askOption.matches("^[1,2]$") == false) {
                System.out.println("Input error,please input the your
option and retry again.");
            } else {
                int input=Integer.parseInt(_askOption);
                if(input==1)
                    askOption=true;
                else
                    askOption=false;
                break;
            }
        }
    }
    public void close() {
        scanner.close();
    }
}
```

最重要的一个Student类，负责对数据库进行相关的操作。

```
package JDBC;
//Student类，负责对数据库进行相关的操作。
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Student {
    private PreparedStatement preparedStatement=null;
    private Connection conn = null;
    private String _name=null;
```

```
    private String _id="";
    private double _grade=0.0;
    private boolean isFonud=false;
    public Student (Connection conn) {
        super();
        this.conn=conn;
    }
    public void add(String id,String name,double grade) {
        //System.out.println(id);
        findByID(id,true);//删除模式
        if(!isFonud) {
            //添加学生信息
            String sql="INSERT INTO Student(id,name,grade) values
(?,?,?)";
            try {
                preparedStatement = conn.prepareStatement(sql);
                preparedStatement.setString(1,id);
                preparedStatement.setString(2,name);
                preparedStatement.setDouble(3,grade);
                preparedStatement.executeUpdate();
                System.out.println("OK");

System.out.println("-------------------------------------");
            } catch (SQLException e) {
                e.printStackTrace();
            }

        } else {
            isFonud = false;
            //要求确认是否要更新信息，如果是更新则更新
            //做出预判是否可以更新
            System.out.println("您是否要更新该学生信息？");
            Input.inputAskOption();
            if(Input.askOption) {
                String sql = "UPDATE Student SET name=?,grade=? WHERE id=?";
                while(true) {
                    if(name.equals(_name)&&grade==_grade) {//如果输入的数据
和原来数据相同则要求重输
                        System.out.println("对不起，输入的数据和源数据一样，请您
输入不同的数据！");
                        Input.inputName();
                        Input.inputGrade();
                    } else { //更新数据
                        break;
```

```
                }
            }
            try {
                preparedStatement = conn.prepareStatement(sql);
                preparedStatement.setString(1, Input._name);
                preparedStatement.setDouble(2, Input._grade);
                preparedStatement.setString(3, id);
                preparedStatement.executeUpdate();
                System.out.println("OK");

System.out.println("--------------------------------------");
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
    public void dispAll() {
        String sql="SELECT * FROM Student";
        try {
            preparedStatement=conn.prepareStatement(sql);
            ResultSet rs=preparedStatement.executeQuery(); //本方法适用与
查询语句
            if(rs.next()){
                isFonud=true; //有学生信息
                rs.beforeFirst();
                System.out.println("id       name       grade");
                while (rs.next()) {
                    _name=rs.getString("name");
                    _id=rs.getString("id");
                    _grade=rs.getDouble("grade");
                    //System.out.println(_id+"   "+_name+"  "+_grade);

System.out.printf("%-10s%-10s%3.2f\n",_id,_name,_grade);
                }
            }
            if(!isFonud) { //没有学生信息
                System.out.println("没有学生信息显示，请您输入学生信息");
                isFonud=false;
            }

System.out.println("--------------------------------------");
        } catch (SQLException e) {
```

```
            e.printStackTrace();
        }
    }
        public void findByID(String id,boolean delMode) {
        if(!delMode) {

System.out.println("-------------------------------------");
            System.out.println("从学号查询学生：");
        }
        String sql="SELECT * FROM Student WHERE id=\""+id+"\""; // SELECT
* FROM Student WHERE ID="XXXXX"; \"是转义输出"
        //System.out.println(sql);
        try {
            preparedStatement=conn.prepareStatement(sql);
            ResultSet rs=preparedStatement.executeQuery(); //本方法适用与
查询语句
            if(rs.next()) {
                rs.beforeFirst();
                isFonud=true; //有学生信息
                if(!delMode)
                    System.out.println("id       name       grade");
                while (rs.next()) {
                    _name=rs.getString("name");
                    _id=rs.getString("id");
                    _grade=rs.getDouble("grade");
                    if(!delMode) {
                        //System.out.println(_id+"   "+_name+"  "+_grade);

System.out.printf("%-10s%-10s%3.2f\n",_id,_name,_grade);

System.out.println("-------------------------------------");
                    }
                }
            } else { //没有学生信息
                if(!delMode) {
                    System.out.println("没有查询到该学生的信息！");
                }
                isFonud=false;

System.out.println("-------------------------------------");
             }
        } catch (SQLException e) {
            e.printStackTrace();
        }
```

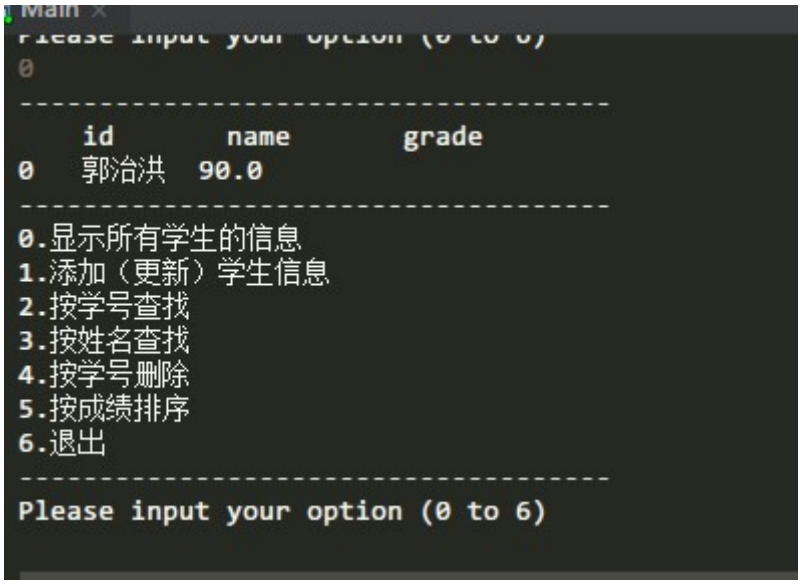
```
    }
    public void findByName(String name) {
        System.out.println("-------------------------------------");
        System.out.println("从名字查询学生：");
        String sql="SELECT * FROM Student WHERE name=?"; // 这是另外一种
查询方法
        try {
            preparedStatement=conn.prepareStatement(sql);
            preparedStatement.setString(1,name); //替换第一个？为String类
型的name
            ResultSet rs=preparedStatement.executeQuery(); //本方法适用与
查询语句
            if(rs.next()){
                rs.beforeFirst();
                isFonud=true; //有学生信息
                System.out.println("id       name       grade");
                while (rs.next()) {
                    _name=rs.getString("name");
                    _id=rs.getString("id");
                    _grade=rs.getDouble("grade");
                    //System.out.println(_id+"   "+_name+"  "+_grade);

System.out.printf("%-10s%-10s%3.2f\n",_id,_name,_grade);
                }
            } else { //没有学生信息
                System.out.println("没有查询到该学生的信息");
                isFonud=false;
            }

System.out.println("-------------------------------------");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    public void delByID(String id) {
        System.out.println("-------------------------------------");
        System.out.println("通过学号删除信息：");
        findByID(id,true);//删除模式
        if(!isFonud) {
            System.out.println("没有查询到要删除的该学生的信息，无法删除");

System.out.println("-------------------------------------");
        } else {
            isFonud=false;
```

```
            String sql="DELETE FROM Student WHERE id=\""+id+"\"";
            try {
                preparedStatement=conn.prepareStatement(sql);
                preparedStatement.executeUpdate();
                System.out.println("删除以下学生信息成功：");
                System.out.println("id       name       grade");
                //System.out.println(_id+"   "+_name+"  "+_grade);

System.out.printf("%-10s%-10s%3.2f\n",_id,_name,_grade);

System.out.println("---------------------------------------");
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

    }
    public void sortByGrade(boolean isASC) {
        System.out.println("按照成绩排序学生：");
        System.out.println("---------------------------------------");
        String mode="ASC";
        if(!isASC)
            mode="DESC";
        String sql="SELECT * FROM Student ORDER BY grade"+" "+mode; // 可
以选择如何排序
        try {
            preparedStatement=conn.prepareStatement(sql);
            ResultSet rs=preparedStatement.executeQuery(); //本方法适用与
查询语句
            if(rs.next()){
                isFonud=true; //有学生信息
                rs.beforeFirst();
                System.out.println("id       name       grade");
                while (rs.next()) {
                    _name=rs.getString("name");
                    _id=rs.getString("id");
                    _grade=rs.getDouble("grade");
                    //System.out.println(_id+"      "+_name+"
"+_grade);

System.out.printf("%-10s%-10s%3.2f\n",_id,_name,_grade);
                }
            }
            if(!isFonud) { //没有学生信息
```

```
                System.out.println("没有学生的信息所供排序！");
            }

System.out.println("-----------------------------------");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    public void close() {
        try {
            if (preparedStatement != null)
                preparedStatement.close();
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
}
```

【实验结果】

本程序很简单，但我进行了很多测试和一些必要处理确保程序能够正确运行。



【实验体会】

此实验，我由于时间不足，做的还是很简单的，还请老师您不要介意！以及感谢老师对我教导和帮助。

另外我还尝试在课余写了一个 GUI 版本，但是只有登录界面，还没有其他界面，有时间我一定会继续学习的。