

## 目录

一、背景及目的要求.....	1
1.1 背景介绍.....	1
1.2 目的以及要求.....	1
二、设计思路.....	2
2.1 总体设计.....	2
2.1.1 思维导图.....	2
2.2 详细设计.....	3
2.2.1 相关包名介绍.....	3
(1)DataBase.....	3
(2) Main .....	4
(3) Students .....	4
(4)Teacher .....	5
三、设计实现及代码分析.....	6
四、程序功能测试及截图.....	17
4.1 程序代码测试.....	17
4.2 功能测试.....	18
五、参考文献.....	22
[1]王杉. Java 面向对象程序设计(第一版).北京:北京理工大学出版社, 2018.....	22
[2]明日科技. Java 从入门到精通(第 6 版).北京:清华大学出版社, 2021.....	22
六、总结与收获.....	23

# 一、背景及目的要求

## 1.1 背景介绍

随着高校规模的不断扩大,学生数量也急剧增加,有关学生的各种信息量也成倍增长,各类学生的统计分析工作也越来越困难,面对如此繁杂的工作,为了能够为高校学生信息管理提供一种更加高效实用的管理手段,为学生信息的存储、计算、统计、分析、交流提供一种更加安全快捷的信息平台,并且能够减少大量的人工操作,以及在人工操作中由于人为因素而引起的数据错误,保证学生信息数据的安全性和完整性,使学生管理人员能够轻松、正确无误地完成各项工作,急需依靠一种信息技术来完成此任务。

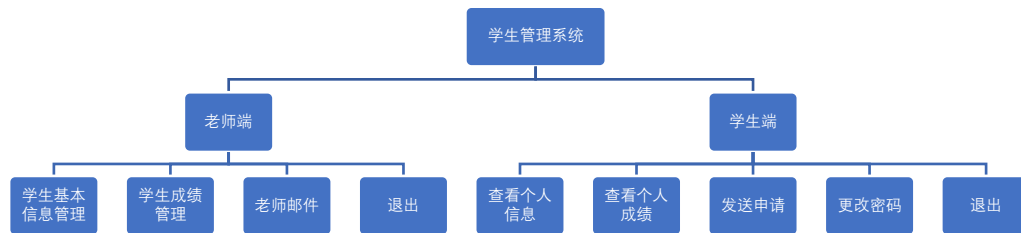
## 1.2 目的以及要求

本次我设计的是用 Java 写的学生管理系统,该管理系统分为老师端和学生端,使用学生通过学号和密码登录进入学生端可以查看本人的学习成绩和本人的学籍信息和完成转专业退学等基本业务。老师通过工号和密码登录进入此管理系统可以方便管理管理,通过此管理系统可以增加查看修改学生的基本信息和学生的成绩以及处理学生发来的申请等功能。利用此管理系统能降低学生管理的成本,并且能规范化学生管理工作。

## 二、设计思路

### 2.1 总体设计

本次我想设计的是一款老师和学生都可以登录的系统，该系统分为老师端和学生端，老师端又分为学生基本信息管理、学生成绩管理、老师邮件、退出并保存功能，其中学生基本信息管理

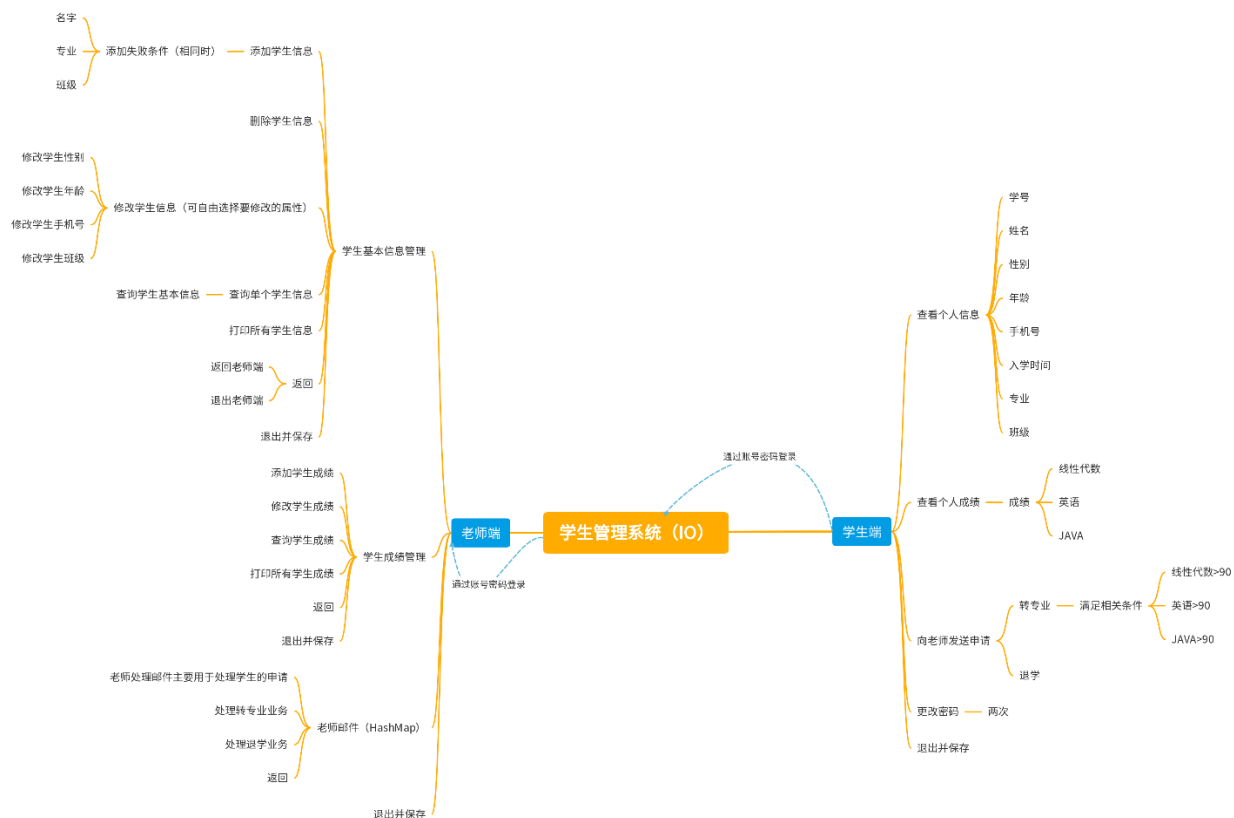


又可以分为添加学生信息、删除学生信息、修改学生信息、查询单个学生信息、打印所有学生信息、返回老师端总界面、退出并保存，其中添加学生信息中，当老师添加相同的名字、专业、班级时系统会提醒已存在该学生。在修改学生信息中，老师可以通过选择学生的基本信息来修改，比如修改学生的性别、年龄、手机号、班级，这些信息可以自由选择。

在学生成绩管理界面中，又可以分为添加学生信息修改学生信息，查询学生信息。打印所有学生信息和返回老师端主界面，或者退出并保存系统。在老师邮箱中可以接受到学生发来的请求，比如转专业申请，退学申请等。

在学生端中可以分为查看个人信息、查看个人成绩、发送申请、更改密码、退出等操作。

#### 2.1.1 思维导图



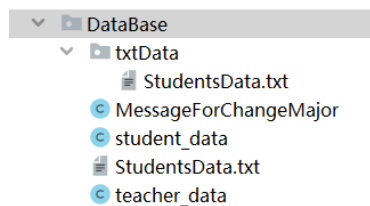
## 2.2 详细设计

### 2.2.1 相关包名介绍

- > DataBase
- > Main
- > Students
- > Teacher

本次学生管理系统分为 DataBase、Main、Students、Teacher 包，每一个包都有自己的方法和类。

#### (1) DataBase

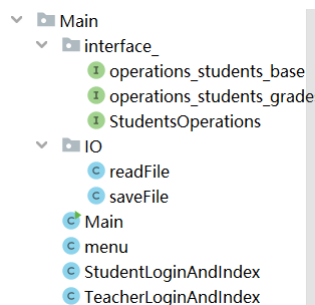


该包的主要作用为保存和提供数据，保存的为该系统的学生数据，如学生的基本信息、成绩等。该包也能为 Main 的主程序提供集合类，如学生需要通过 `LinkedHashSet` 集合来保存数据，

该包中的 `sudent_data` 方法即可返回一个集合。

同样，老师也是如此，其中的 `MessageForChangeMajor` 方法返回的是一个用于保存学生向老师发送信息的集合，该方法返回的类型为 `HashMap`。

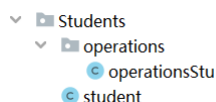
## (2) Main



该 Main 包作为该管理系统的主包，其主要作用为启动该程序，并且作为该系统的前端展示界面。该包下面的 `interface_` 包中含有关于老师和学生方法的接口。前两个是老师端的用户处理学生基本信息和学生成绩的接口，最后一个接口是学生端，学生的一些方法。

IO 包中包含的主要是，该管理系统的 IO 流操作，用于保存数据和读取数据。`Main()` 方法中是程序的入口，在该方法中会创建数据库如老师端集合和学生端的集合，该方法会调用 `menu` 方法，启动学生管理系统。`StudentLoginAndIndex` 方法中是学生的登录界面和学生端的主界面。`TeacherLoginAndIndex` 方法中是老师端的登录界面和老师端的主界面。

## (3) Students

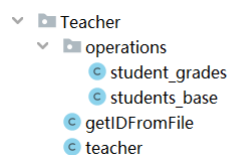


该包的可分为 `operations` 包和 `student` 类，其中 `operations` 包中含有学生端学生的一些方法，如查看个人信息、查看个人成绩、转专业等方法。将该系统分为

这么多包是便于以后的维和更新。其中 `student` 类中含有学生的属性，该学生属性全为私有，方便在其他方法中调用，在类中重写了 `equals` 和 `hashCode` 方法主要作用就是防止在一个专业班中出现姓名相同的情况。该学生类中包含学号、姓

名、性别、年龄、专业、班级、手机号码、登录密码、数学成绩、英语成绩、Java 成绩、入学时间关键信息。

#### (4) Teacher



该包中包含了 `operations` 包和两个方法，其中 `operations` 包中含有两个关于学生的类，第一个是学生的成绩类，该类中包含学生的成绩管理，查询成绩、修改成绩、查看所有学生成绩。第二个类是学生基本信息类，该类中包含修改学生信息、查看学生信息、查看所有学生信息、增加学生信息。

在 `getIDFromFile` 类中主要是从已保存的数据文件中获取最后一行学生的 ID，如果存在数据就返回最后一行学生 ID 否则将返回 2101 起始 ID，在 `students_base` 中的添加学生信息可以看到它的身影。

在 `teacher` 类中包含的是管理员老师的基本信息，主要包括姓名、工号、密码，因为老师也需要通过输入密码登录进入学生管理系统。

### 三、设计实现及代码分析

```
package Main;

import DataBase.student_data;
import DataBase.teacher_data;
import Main.IO.readFile;
import Students.student;
import Teacher.teacher;

import java.io.IOException;
import java.util.HashMap;
import java.util.LinkedHashSet;

/**
 * @author 陈勇
 * @Major 网络工程 K2102 班
 * @version 1.0
 */
public class Main {
    public static void main(String[] args) throws IOException {
        // 创建数据库
        LinkedHashSet<teacher> teacherSet =
teacher_data.return_teacherSet();
        LinkedHashSet<student> studentSet =
student_data.return_studentSet();
        HashMap<student,String> messageData =
DataBase.MessageForChangeMajor.return_messageData();
        // 创建 admin
        teacher teacher = new teacher(1, "陈勇", 123);
        teacherSet.add(teacher);

        // 老师测试案例
        // student student = new student(2101, "陈勇", "男", 18, "网络工程", "网络工程本科 21 级", "18236416916", 123, "2021 年");
        // student student1 = new student(2102, "王生", "男", 18, "网络工程", "网络工程本科 21 级", "18236416916", 123, "2021 年");
        // studentSet.add(student);
        // studentSet.add(student1);

        // 将数据导入集合中
    }
}
```

```

        readFile. inputStudentFile(studentSet);
    while(true) {
        int choice = menu. menu();
        switch (choice) {
            case 1:

StudentLoginAndIndex. student_login(messageData, studentSet);
                break;
            case 2:

TeacherLoginAndIndex. teacher_login(teacherSet, studentSet);
                break;
            case 3:
                break;
        }
    }
}

package Main;

import Main. IO. saveFile;
import Students. operations. operationsStu;
import Students. student;

import java.util. HashMap;
import java.util. LinkedHashSet;
import java.util. Scanner;

/**
 * @author Chen_Cat
 * @Time 2022/12/8 16:00
 */
@SuppressWarnings("all")
public class StudentLoginAndIndex {
    static Scanner scanner = new Scanner(System. in);
    public static void student_login(HashMap<student, String>
messageData, LinkedHashSet<student> studentSet) {
        int id = 0;
        System. out. println("-----学生端登录界面-----");
        try {
            scanner = new Scanner(System. in);

```



```

        System.out.printf("请输入学号: ");
        id = scanner.nextInt();
    } catch (Exception e) {
        System.out.println("输入有误, 请重新输入");
        student_login(messageData, studentSet);
    }
}

// 判断学生是否存在
boolean flag = false;
for (student s : studentSet) {
    if (s.getId() == id) {
        flag = true;
    }
}

if (flag == false) {
    System.out.println("未查询到相关用户信息");
    System.out.printf("是否重新输入? (y/n) ");
    String s = scanner.next();
    if (s.equals("y")) {
        student_login(messageData, studentSet);
    } else {
        System.out.println("退出系统");
        System.exit(0);
    }
}

int password = 0;
try {
    scanner = new Scanner(System.in);
    System.out.println();
    System.out.printf("请输入密码: ");
    password = scanner.nextInt();
} catch (Exception e) {
    System.out.println("输入有误, 请重新输入");
    student_login(messageData, studentSet);
}

// 判断密码是否正确
for (student s : studentSet) {
    if (s.getId() == id) {
        if (s.getPassword() == password) {
            System.out.println("登录成功");
            student_index(messageData, s);
        }
    }
}

```



```

        System.out.println("退出系统");
        System.exit(0);
        break;
    default:
        System.out.println("输入有误，请重新输入");
        break;
    }
}
}
}
}

```

```

package Main;

import Main.IO.saveFile;
import Students.student;
import Teacher.operations.student_grades;
import Teacher.operations.students_base;
import Teacher.teacher;

import java.io.*;
import java.util.LinkedHashSet;
import java.util.Scanner;

/**
 * @author Chen_Cat
 */
public class TeacherLoginAndIndex {

    static Scanner scanner = new Scanner(System.in);

    public static void teacher_login(LinkedHashSet<teacher>
teacherSet, LinkedHashSet<student> studentSet) throws IOException {
    //      用来判断教师是否登录成功;
    //      当 flag = -1 表示没有该教师
    //      当 flag = 0 表示有改教师 但是密码输入错误
    //      当 flag = 1 表示有改教师 并且密码输入正确
    int flag = -1;

```

```

int id = 0;
int password = 0;
teacher teacher = null;
System.out.println("-----教师端登录界面-----");
try {
    System.out.printf("请输入职工号: ");
    id = scanner.nextInt();
} catch (Exception e) {
    System.out.println("请输入数字, 请重新输入");
    teacher_login(teacherSet, studentSet);
}

System.out.println();
try {
    System.out.printf("请输入密码: ");
    password = scanner.nextInt();
} catch (Exception e) {
    System.out.println("请输入数字, 请重新输入");
    teacher_login(teacherSet, studentSet);
}

for (teacher t : teacherSet) {
    if (t.getId() == id) {
        flag = 0;
        if (t.getPassword() == password) {
            teacher = t;
            flag = 1;
        }
    }
}

if (flag == -1) {
    System.out.println("未查询到相关用户信息");
    System.out.printf("是否重新输入? (y/n) ");
    String s = scanner.next();
    if (s.equals("y")) {
        teacher_login(teacherSet, studentSet);
    } else {
        System.out.println("退出系统");
        System.exit(0);
    }
} else if (flag == 0) {

```

```

        System.out.println("密码输入有误");
        System.out.printf("是否重新输入? (y/n) ");
        String s = scanner.next();
        if (s.equals("y")) {
            teacher_login(teacherSet, studentSet);
        } else {
            System.out.println("退出系统");
            System.exit(0);
        }
    } else {
        System.out.println("登录成功");
        System.out.println("欢迎" + teacher.getName() + "老师");
        System.out.println("正在跳转到教师端界面....");
        teacher_index(teacher, studentSet);
    }
}

public static void teacher_index(teacher teacher,
LinkedHashSet<student> studentSet) {
    System.out.println("-----教师端主界面-----");
    System.out.println("1、学生基本信息管理");
    System.out.println("2、学生成绩管理");
    System.out.println("3、邮件信息");
    System.out.println("4、退出系统");
    int choice = 0;
    try {
        System.out.printf("请输入您的选择: ");
        choice = scanner.nextInt();
    } catch (Exception e) {
        System.out.println("请输入数字, 请重新输入");
        teacher_index(teacher, studentSet);
    }
    switch (choice) {
        case 1:
            System.out.println("正在跳转到学生基本信息管理界面....");
            student_info(teacher, studentSet);
            break;
        case 2:
            System.out.println("正在跳转到学生成绩管理界面....");
            student_grades(teacher, studentSet);

```

```

        break;
    case 3:
        System.out.println("正在跳转到邮件信息界面...");
        break;
    case 4:
        saveFile.saveStudentFile(studentSet);
        System.out.println("退出系统");
        System.exit(0);
        break;
    default:
        System.out.println("输入有误, 请重新输入");
        teacher_index(teacher, studentSet);
        break;
    }
}

public static void student_info(teacher teacher,
LinkedHashSet<student> studentSet) {
    while (true) {
        System.out.println("-----学生基本信息管理界面-----");
        System.out.println("1、添加学生信息");
        System.out.println("2、删除学生信息");
        System.out.println("3、修改学生信息");
        System.out.println("4、查询学生信息");
        System.out.println("5、打印所有学生信息");
        System.out.println("6、返回上一级");
        System.out.println("7、退出系统");
        int choice = 0;
        try {
            scanner = new Scanner(System.in);
            System.out.printf("请输入您的选择: ");
            choice = scanner.nextInt();
        } catch (Exception e) {
            System.out.println("请输入数字, 请重新输入");
            student_info(teacher, studentSet);
        }
        students_base studentsBase = new students_base();
        switch (choice) {
            case 1:
                System.out.println("正在跳转到添加学生信息界

```

```

面....");

        studentsBase.addStudent(studentSet);
        break;
    case 2:
        System.out.println("正在跳转到删除学生信息界
面....");

        studentsBase.deleteStudent(studentSet);
        break;
    case 3:
        System.out.println("正在跳转到修改学生信息界
面....");

        studentsBase.updateStudent(studentSet);
        break;
    case 4:
        System.out.println("正在跳转到查询学生信息界
面....");

        studentsBase.queryStudent(studentSet);
        break;
    case 5:
        System.out.println("正在跳转到打印所有学生信息界
面....");

        studentsBase.printAllStudent(studentSet);
        break;
    case 6:
        System.out.println("正在返回上一级....");
        teacher_index(teacher, studentSet);
        break;
    case 7:
        saveFile.saveStudentFile(studentSet);
        System.out.println("退出系统");
        System.exit(0);
        break;
    default:
        System.out.println("输入有误，请重新输入");
        student_info(teacher, studentSet);
        break;
    }
}

}

}

public static void student_grades(teacher teacher,

```

```

LinkedHashSet<student> studentSet) {
    while (true) {
        System.out.println("-----学生成绩管理界面-----");
        System.out.println("        1、添加学生成绩        ");
        System.out.println("        2、修改学生成绩        ");
        System.out.println("        3、查询学生成绩        ");
        System.out.println("        4、打印所有学生成绩

");

        System.out.println("        5、返回上一级        ");
        System.out.println("        6、退出系统        ");
        int choice = 0;
        try {
            System.out.printf("请输入您的选择: ");
            choice = scanner.nextInt();
        } catch (Exception e) {
            System.out.println("请输入数字, 请重新输入");
            student_grades(teacher, studentSet);
        }
        student_grades studentsGrades = new student_grades();
        switch (choice) {
            case 1:
                System.out.println("正在跳转到添加学生成绩界
面....");
                studentsGrades.addStudentGrades(studentSet);
                break;
            case 2:
                System.out.println("正在跳转到修改学生成绩界
面....");
                studentsGrades.updateStudentGrades(studentSet);
                break;
            case 3:
                System.out.println("正在跳转到查询学生成绩界
面....");
                studentsGrades.queryStudentGrades(studentSet);
                break;
            case 4:
                System.out.println("正在跳转到打印所有学生成绩界
面....");
                studentsGrades.printAllStudentGrades(studentSet);
                break;
            case 5:
                System.out.println("正在返回上一级....");

```



```

        teacher_index(teacher, studentSet);
        break;
    case 6:
        saveFile.saveStudentFile(studentSet);
        System.out.println("退出系统");
        System.exit(0);
        break;
    default:
        System.out.println("输入有误，请重新输入");
        student_grades(teacher, studentSet);
        break;
    }
}
}
}

```

## 四、程序功能测试及截图

### 4.1 程序代码测试

当运行程序时，系统会跳转到管理系统主界面，会提示输入选择，1、学生端，2、教师端，当输入1时会跳转到学生端，会显示让输入信息挑选功能包括：查看个人信息、查看个人成绩、申请转专业、更改密码、退出系统。此时选择查看个人信息，会显示当前的个人信息，选择查看个人成绩，如果老师未录入成绩那么将显示未考试，选择申请转专业，则系统会先判断是否满足条件成绩是否合格。更改密码，则会提示输入一遍然后再输入一遍密码，如果两次输入的不一樣则会提示输入不正确

## 4.2 功能测试

### 主界面

```
D:\JDK8\bin\java.exe "-javaagent:D:\Software\
-----欢迎使用学生管理系统-----
1、学生端
2、教师端
3、退出系统
请选择:
```

### 登录老师端

```
请选择:2
-----教师端登录界面-----
请输入职工号: 1

请输入密码: 123
登录成功
欢迎陈勇老师
正在跳转到教师端界面....
-----教师端主界面-----
1、学生基本信息管理
2、学生成绩管理
3、邮件信息
4、退出系统
请输入您的选择:
```

### 选择学生基本信息管理

```
请输入您的选择: 1
正在跳转到学生基本信息管理界面....
-----学生基本信息管理界面-----
1、添加学生信息
2、删除学生信息
3、修改学生信息
4、查询学生信息
5、打印所有学生信息
6、返回上一级
7、退出系统
请输入您的选择: |
```

## 添加学生信息

```

  请输入您的选择: 1
  当前ID为2101
  正在跳转到添加学生信息界面....
  请输入学生的姓名:
  张三
  请输入学生性别:
  男
  请输入学生年龄:
  18
  请输入学生专业:
  网络工程
  请输入学生班级:
  网络工程2班
  请输入学生电话:
  182356781
  请输入学生入学时间:
  2102
  请为学生设置密码:
  123
  添加成功

```

## 删除学生信息

```

  请输入您的选择: 2
  正在跳转到删除学生信息界面....
  请输入学生的姓名: 张三
  请输入学生专业 网络工程
  请输入学生班级 网络工程2班
  删除成功
  -----学生基本信息管理界面-----
                                1、添加学生信息

```

## 修改学生信息

```

  请输入您的选择: 3
  正在跳转到修改学生信息界面....
  请输入要修改的学生的姓名: 张四
  请输入需要修改的信息: 1.性别 2.年龄 3.电话 4.班级
  每个数字之间用空格隔开,不需要修改的信息输入0
  请输入: 1 0 0 0
  请修改学生的性别 女
  修改成功

```

## 查看学生信息

1、退出系统

请输入您的选择: 4

正在跳转到查询学生信息界面....

请输入学生的姓名: 张四

学号: 2103

姓名: 张四

性别: 男

年龄: 18

专业: 网络工程

班级: 网络工程2班

电话: 1823571729

入学时间: 2012年

## 打印所有学生信息

正在跳转到打印所有学生信息界面....

学号	姓名	性别	年龄	手机号	专业	班级	入学时间
2103	张四	女	18	1823571729	网络工程	网络工程2班	2012年

按回车键退出

## 返回上一级选择学生成绩管理中的添加学生信息

0、退出系统

请输入您的选择: 1

正在跳转到添加学生成绩界面....

请输入学生的姓名: 张四

请输入学生的线性代数成绩: 100

请输入英语成绩: 100

请输入JAVA成绩: 100

添加成功

-----学生成绩管理界面-----

## 修改学生信息

6、退出系统

请输入您的选择: 2

正在跳转到修改学生成绩界面....

请输入学生的姓名: 张四

请输入学生的线性代数成绩: 99

请输入英语成绩: 99

请输入JAVA成绩: 99

修改成功

-----学生成绩管理界面-----

## 查询学生信息

```
正在跳转到查询学生成绩界面....
请输入学生的姓名: 张四
学生姓名: 张四
学生学号: 2103
学生性别: 女
学生年龄: 18
学生线性代数成绩: 99.0
学生英语成绩: 99.0
学生JAVA成绩: 99.0
```

## 打印所有学生信息

```
请输入您的选择: 4
正在跳转到打印所有学生成绩界面....
姓名      学号      性别      年龄      线性代数成绩      英语成绩      JAVA成绩
张四      2103      女        18        99.0              99.0          99.0
-----学生成绩管理界面-----
1  添加学生成绩
```

## 退出系统

```
6、退出系统
请输入您的选择: 6
-----
写入成功!
退出系统
```

## 查看写入文件

```
StudentsData.txt
Main.java x StudentsData.txt x
2103, 张四, 女, 18, 网络工程, 网络工程2班, 1823571729, 123, 99.0, 99.0, 99.0, 2012年
```

## 五、参考文献

[1]王杉. Java 面向对象程序设计（第一版）.北京：北京理工大学出版社，  
2018

[2]明日科技. Java 从入门到精通（第 6 版）.北京：清华大学出版社，2021

## 六、总结与收获

该系统具有非常强的健壮性,加入了大量的异常捕获不管是用户输入的时候还是后端处理数据的时候不会出现 bug 和异常,但是由于加入了大量的异常捕获代码,导致代码量较大并且代码的冗余程度较高,应到对代码进行算法优化,使用算法优化后将可以使程序运行的更快占用更少的空间,该系统还有一些扩展功能比如通过总成绩对所有学生进行排序,也可以检索的时候通过专业班级来检索学生信息。

在这次实训项目中我得到了许多收获,我第一次体会到 JAVA 工程师的工作是什么样子的,也发现了很多不曾注意到的细节,在实训的演示过程中,我对作为一名 JAVA 工程师注意的事项应该从小事做起、从基本做起.把职业能力训练与职业素质有机的结合起来.相信这对我接下来学习专业知识会起到很有效的帮助,在接下来的几个月里,我会以作为一名职业人的身份在这几个方面要求自己,严格自我,向 JAVA 工程师靠近。

实训项目结束了感觉虽然辛苦,但很充实.我想这就是 JAVA 工程师的工作.我也体会到 JAVA 工程师成功地完成一个任务的那种兴奋,那种小有成就的感觉是只有置身其中的人才能体会的.

总之,这次实训为我提供了与众不同的学习方法和学习体会,从书本中面对现实,为我将来走上社会打下了扎实的基础.作为在校计算机专业的学生,现在我能做的就是吸取知识,提高自身的综合素质,提高自己的职业技能,自己有了能力,到时候才会是“车到山前必有路”.我相信在不久的未来,会有属于我自己的一片天空.