



北京科技大学
University of Science and Technology Beijing

北 京 科 技 大 学

《Java》综合作业实验报告

姓 名：_____ 黄德希 _____ 卢冠男 _____

班 级：_____ 工商 1902 _____

学 号：_____ 41907054 _____ 41907038 _____

学 院：_____ 东凌经济管理学院 _____

指导老师：_____ 宋 晏 _____

2020 年 06 月 05 日

目 录

一、实验目的	2
二、实验环境	3
三、实验分工	3
四、指令集合	3
五、测试样本	4
六、代码分析	4
七、测试结果	5
八、总结与心得体会	8
九、附录	9

一、实验目的

题目要求：

从若干文本文件汇总学生各门课成绩数据，将汇总后的数据保存至一个.csv 文件。（每行数据之间以逗号分隔，可以直接使用 Excel 打开。）

例如：

大基.txt - 记事本

姓名	学号	大基
张小艳	41926013	95.0
徐奥	41926079	95.0
唐凯悦	41926081	95.0
魏勃轩	41926087	95.0

选修.txt - 记事本

姓名	学号	选修
唐凯悦	41926081	93.75
董明宇	41926085	90.0
姚之涵	41926077	70.0
彭真	41926084	87.5

高数.txt - 记事本

姓名	学号	高数
彭真	41926084	98
唐凯悦	41926081	98
阎佳	41926022	97
李泽轩	41926006	96
李安典	41926038	96

英语.txt - 记事本

姓名	学号	英语
梁梦琪	41926083	94.4
傅思宇	41926086	93.8
李昊伟	41926068	93.5
杨宇贤	41926040	93.39

注意：各文件学生名单不尽相同，需要进行全统计，不存在科目成绩记为 0，统计之后的数据如下：

	A	B	C	D	E	F
1	学号	姓名	大基	英语	高数	选修
2	41926013	张小艳	95	89.8	61	0
3	41926079	徐奥	95	91.04	53	0
4	41926081	唐凯悦	95	90.63	98	93.75

在汇总数据基础上，进行数据统计。在命令行循环输入命令，并输出统计结果，输入 exit 时，系统结束。

例如：

请输入命令: avg 唐凯悦 大基
唐凯悦的平均分是: 94.35
请输入命令: 请输入命令: avg 唐凯悦
唐凯悦的平均分是: 94.35
请输入命令: avg 大基
大基的平均分是: 81.40
请输入命令: avg Java
你输入的既不是课程名，也不是学生名
请输入命令: get 唐凯悦 英语
唐凯悦的英语的成绩是: 90.63
请输入命令: get 王林林 英语
没有 王林林 这个人
请输入命令: get 唐凯悦 Java
唐凯悦没有 Java 这门课程

请输入命令: sort 高数

名次	姓名	大基	高数	英语	选修
1	唐凯悦	95.0	98.0	90.63	93.75
2	彭真	95.0	98.0	78.5	47.5
3	阎佳	90.0	97.0	91.12	0.0
4	汪龙	90.0	96.0	87.6	0.0
5	李安典	90.0	96.0	91.55	0.0
6	李泽轩	90.0	96.0	92.46	0.0
7	姚之涵	95.0	95.0	92.9	70.0
8	韩学敏	90.0	95.0	91.16	0.0
9	吕奉泽	90.0	95.0	92.34	0.0
10	刘立奇	95.0	94.0	90.15	0.0

其他数据此处省略

请输入命令: exit
退出查询系统! byebye!

必须完成的命令

(1) 平均分统计

avg 学生姓名: 统计学生平均分。

avg 课程名: 统计课程平均分。

avg 后面如果既不是课程名，也不是学生名：给出错误提示。

(2) 某人某门课成绩查询

get 学生姓名 课程名：获取指定学生指定课程的分。

get 课程名 学生姓名：获取指定课程指定学生的分。

get 后面如果学生或课程不存在：给出相应提示。

(3) 成绩排序

sort 课程：按照指定课程的分降序输出成绩列表。

sort 后面的课程如果不存在：给出相应提示。

(4) 退出查询系统

exit：结束程序运行。

二、实验环境

1、javaSE 8.0

2、Eclipse

三、实验分工

黄德希主要负责学生成绩系统需求分析、登录系统代码、教师端部分代码、学生端代码、运行视频录制、代码分析及实验报告书写，约占小组工作的 50%。

卢冠男主要负责成绩系统需求分析、API 文档查询、学生端代码、教师端部分代码、实验报告书写及实验报告排版，约占小组工作的 50%。

四、指令集合

此次完成的是一个学生成绩的查询系统，经过小组的讨论并结合实际需求，我们决定分为两个用户端，一为教师端，其命令和对应用途如表 1，二为学生端，其命令和对应用途如表 2，并在原有基础上新增一些功能。

表格 1 教师端命令行及用途

命令	用途
set 姓名 学号 (新增)	新增一名学生，需输入对应学号及姓名
set 姓名 (学号) 科目 科目成绩 (新增)	设定学生的某一科的成绩
avg 姓名	统计学生平均分
avg 课程名	统计课程平均分
get 姓名 课程名	获取指定学生指定课程的分
get 课程名 姓名	获取指定课程指定学生的分
get 姓名 加权分 (新增)	获取指定学生的加权分
get 课程名 不及格 (新增)	获取某个科目的不及格名单
import (新增)	新增一门科目的成绩

sort 课程	按照指定课程的分数降序输出成绩列表
sort 加权分 (新增)	按照加权分数降序输出成绩列表
export	导出学生成绩为.csv 文件
exit	结束程序运行

表格 2 学生端命令行及用途

命令	用途
get 成绩 (新增)	获取学生所有课程成绩
get 课程 排名 (新增)	获取本学生指定课程指定的排名
get 加权分 排名 (新增)	获取本学生的加权分排名
exit	结束程序运行

五、测试样本

测试样本的学生姓名、学号、成绩均为虚拟，见大基.txt、高数.txt、选修.txt、英语.txt，为了测试 import 功能，额外加入化学.txt。

六、代码分析

该系统由四大部分构成，第一个部分是登录部分，包含 Login、LoginData 和 SignIn 三个 class，该部分代码是通过小组学习研讨、查询 API 完成的，教师端的账户密码都是 teacher，学生的账户密码都是学号，如需添加学生，可通过 LoginData.Register(String account,String password)方法来添加。

第二个部分是教师端部分，包含添加学生的 setStudent()方法，获取学生信息的 getInformation()，添加课程 import()方法，import 添加的课程在下次打开程序会自动添加，排名的 sortCourse()和 rankCourse()方法，最后还有导出学生成绩的为 csv 文件的 exportCsv()方法，若需按照一定排名顺序导出，可先用 rankCourse()方法后再导出，

表格 3 Teacher 类的方法

方法	用途
void setStudent(String temp[])	新增学生，设定学生成绩
void sortCourse(String course)	对某个科目进行排名
void rankCourse(String courseName)	为 sortCourse(String course)方法服务
void importCourse()	导入新的科目成绩（下次打开程序会自动导入）
void getAvg(String temp[])	获得某个学生或课程的平均分
void exportCsv()	导出课程成绩
void getInformation(String[] temp)	获得某个学生或科目的具体情况

第三个部分是学生端部分，属性有 name 姓名、ID 学号，具备常规的 set、get 方法，

courseList 课程成绩列表、totalHour 总学时、electiveCourseHour 选修课学分、weightedScore 加权分、仅有 get 方法。构造方法有 Student()、Student(String name, String iD), 以下为 Student 类的方法。

表格 4 Student 类的方法

方法	用途
String addCourse(String name,double hour,double score)	添加课程
String getCourse()	获得该学生所有课程的具体信息
String getCourseAverage(String courseName)	获得该学生该科的成绩和该科的平均分
double getTotalHour()	获得总学分
double getElectiveCourseHour()	获得选修学分
double getWeightedScore()	获得加权分
void getRank(String[] temp)	获得某一科的排名
String toRow()	为教师端导出课程成绩的方法服务

为了获得排名, Student 类和 Teacher 类都继承了 SortCourseCourse 类。

第四部分是 Main 类, 有 stuList 学生列表组, 初始化的 static 静态代码块, 对 importcourses.txt 中的科目进行初始化导入, 以及与用户交互的 switch 代码和 static void addStudentList(String courseName,double hour,String path)的通过文本添加学生信息的代码, 特别地, 对未参加该科目学习的学生记为 0 分。

七、测试结果

欢迎进入学生成绩管理系统!

 请输入您的账号: teacher
 请输入您的密码: teacher
 登录成功, 欢迎您: teacher

欢迎进入学生成绩管理系统!

 请输入您的账号: 41926081
 请输入您的密码: 41926081
 登录成功, 欢迎您: 41926081

欢迎进入学生成绩管理系统!

 请输入您的账号: 123456
 请输入您的密码: 123456
 登录失败
 请输入您的账号: teacher
 请输入您的密码: teacher
 登录成功, 欢迎您: teacher

请输入您的账号: teacher
请输入您的密码: teacher
登录成功, 欢迎您: teacher
set 黄德希 41907054
已添加学生: 黄德希 41907054

set 黄德希 高数 93
学生: 黄德希 课程: 高数 分数已设为: 93

set 41907054 英语 85
学生: 黄德希 课程: 英语 分数已设为: 85

avg 唐凯悦
姓名: 唐凯悦 学号: 41926081
已修学科如下:
高数: 98 大基: 95 英语: 91 选修: 94 化学: 85 平均分: 92

avg 大基
课程: 大基 平均分: 91.12

get 唐凯悦 高数
姓名: 唐凯悦 学号: 41926081 高数: 98

get 高数 唐凯悦
姓名: 唐凯悦 学号: 41926081 高数: 98

get 大基 不及格
您好, 大基没有不及格人员

get 高数 不及格
高数的不及格人数为: 6
姓名 学号 成绩
廖洲 41926057 60
徐奥 41926079 53
扎西旺堆 41926062 42
张凡 41926042 40
郝琦 41926018 19
符谓 41926054 6

import
请输入课程名: 化学
请输入课程学分: 2
请输入文本路径: D:\Desktop\java\化学.txt
已添加课程: 化学

sort 加权分

名次	姓名	学号	高数	大基	英语	选修	化学	总学分	加权分
1	姚之涵	41926077	95	95	93	70	0	14.5	94.33
2	阎佳	41926022	97	90	91	0	94	14.5	93.76
3	唐凯悦	41926081	98	95	91	94	85	16.5	93.66
4	李泽轩	41926006	96	90	92	0	93	14.5	93.58
5	吕奉泽	41926034	95	90	92	0	95	14.5	93.40
6	李安典	41926038	96	90	92	0	0	12.5	93.38
7	董明宇	41926085	93	95	93	90	0	14.5	93.27
8	刘立奇	41926004	94	95	90	0	94	14.5	93.11
9	韩学敏	41926023	95	90	91	0	95	14.5	93.08
10	汪龙	41926072	96	90	88	0	96	14.5	92.65
11	崔慧玲	41926082	93	90	91	0	93	14.5	92.06
12	李昊伟	41926068	91	90	94	0	93	14.5	91.79
13	晋发俊	41926026	91	90	92	0	91	14.5	91.17
14	彭真	41926084	98	95	79	88	89	16.5	90.86
15	莫卓霖	41926078	91	85	93	86	92	16.5	90.53
16	龚健南	41926021	91	90	90	0	91	14.5	90.49
17	李洋	41926007	88	95	91	0	0	12.5	90.43
18	韩阳阳	41926025	90	90	92	64	90	16.5	90.43
19	吴俊豪	41926011	89	90	93	0	89	14.5	90.14
20	朱克迪	41926065	88	90	91	0	94	14.5	90.03
21	黄德希	41907054	93	0	85	0	0	10.0	89.80
22	李响	41926039	88	90	93	0	88	14.5	89.74
23	江亚桥	41926067	89	90	91	0	89	14.5	89.71
24	傅思宇	41926086	89	95	94	0	74	14.5	89.29
25	马雪璞	41926030	85	90	93	0	88	14.5	88.46
26	魏天宇	41926058	87	0	91	0	87	12.0	88.42
27	缪京达	41926029	91	95	90	0	68	14.5	88.35
28	吴文俊	41926070	88	90	92	0	80	14.5	88.27
29	杨清彬	41926009	85	90	92	0	85	14.5	87.76
30	闫庆明	41926035	89	90	84	0	89	14.5	87.71
31	杨宇炎	41926040	82	90	93	0	88	14.5	87.35
32	殷逸菲	41926080	81	95	92	0	0	12.5	87.29
33	王宇航	41926061	83	90	93	0	83	14.5	86.84
34	赵智祥	41926076	83	90	92	0	83	14.5	86.76
35	魏勃轩	41926087	80	95	92	86	80	16.5	86.02
36	蔡宗岐	41926027	82	95	88	63	82	16.5	85.95
37	张畋	41926044	81	90	92	0	81	14.5	85.59

38	何天森	41926071	79	95	93	0	79	14.5	85.52
39	王展华	41926031	82	90	90	0	82	14.5	85.51
40	赵沁瑞	41926049	81	90	91	0	81	14.5	85.41
41	徐一博	41926020	80	90	91	0	80	14.5	84.86
42	范思当	41926015	82	95	90	0	67	14.5	84.43
43	许倩	41926036	80	95	90	0	73	14.5	84.29
44	艾合买提	41926033	92	85	72	0	0	12.5	84.29
45	于辰浩	41926059	85	90	93	0	57	14.5	84.15
46	杨智坤	41926008	92	90	67	0	0	12.5	83.52
47	马震	41926060	81	95	80	0	81	14.5	83.08
48	胡梅钰	41926050	87	90	91	0	47	14.5	83.08
49	张轩	41926073	77	90	90	0	77	14.5	82.84
50	刘可	41926003	76	90	91	0	76	14.5	82.43
51	梁梦琪	41926083	80	70	94	0	80	14.5	82.25
52	查宝平	41926019	80	90	92	0	57	14.5	81.86
53	付永辉	41926064	71	95	92	0	78	14.5	81.86
54	张明鑫	41804016	77	0	91	98	77	14.0	81.58
55	白玛卓嘎	41926032	72	90	91	63	78	16.5	81.09
56	张越	41926045	71	95	92	0	71	14.5	80.86
57	邓星平	41926063	75	95	91	0	59	14.5	80.74
58	章逸凡	41926055	72	90	92	0	72	14.5	80.63
59	梁保盼	41926056	67	90	92	0	67	14.5	77.85
60	张小艳	41926013	61	95	90	0	78	14.5	77.15
61	周婷婷	41926075	65	90	93	0	65	14.5	76.97
62	赵宇	41926048	65	90	74	75	94	16.5	75.89
63	丁可可	41926001	61	90	89	0	0	12.5	75.81
64	张生云	41926043	62	95	80	0	80	14.5	75.03
65	张慧娟	41926074	62	90	91	0	62	14.5	74.73
66	刘晓凡	41926066	62	90	90	0	62	14.5	74.57
67	郑文超	41926017	65	0	91	0	65	12.0	73.60
68	陈彩媚	41926014	66	0	89	0	66	12.0	73.59
69	徐典	41926079	53	95	91	0	53	14.5	70.74
70	束宇航	41926010	61	0	89	0	61	12.0	70.28
71	廖洲	41926057	60	0	82	0	49	12.0	65.60
72	张凡	41926042	40	90	86	0	0	12.5	64.82
73	扎西旺堆	41926062	42	90	61	0	0	12.5	57.62
74	郝琦	41926018	19	90	88	0	0	12.5	55.17
75	尚归兴	41926047	61	0	46	0	0	10.0	55.15
76	符譞	41926054	6	0	0	0	0	6.0	6.00

登录成功, 欢迎您:teacher

export

导出成功!

exit

感谢您的使用!

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	学号	姓名	高数	大基	英语	选修	化学	总学时	选修学时	加权分	排名				
2	41926077	姚之涵	95	95	93	70	0	14.5	2	94.33	1				
3	41926022	阎佳	97	90	91	0	94	14.5	0	93.76	2				
4	41926081	唐凯悦	98	95	91	94	85	16.5	2	93.66	3				
5	41926006	李泽轩	96	90	92	0	93	14.5	0	93.58	4				
6	41926034	吕奉泽	95	90	92	0	95	14.5	0	93.4	5				
7	41926038	李安典	96	90	92	0	0	12.5	0	93.38	6				
8	41926085	董明宇	93	95	93	90	0	14.5	2	93.27	7				
9	41926004	刘立奇	94	95	90	0	94	14.5	0	93.11	8				
10	41926023	韩学敏	95	90	91	0	95	14.5	0	93.08	9				
11	41926072	汪龙	96	90	88	0	96	14.5	0	92.65	10				
12	41926082	崔慧玲	93	90	91	0	93	14.5	0	92.06	11				
13	41926068	李昊伟	91	90	94	0	93	14.5	0	91.79	12				
14	41926026	曾发俊	91	90	92	0	91	14.5	0	91.17	13				
15	41926084	彭真	98	95	79	88	89	16.5	2	90.86	14				
16	41926078	莫卓霖	91	85	93	86	92	16.5	2	90.53	15				
17	41926021	龚健南	91	90	90	0	91	14.5	0	90.49	16				
18	41926007	李洋	88	95	91	0	0	12.5	0	90.43	17				
19	41926025	韩璐阳	90	90	92	64	90	16.5	2	90.43	18				
20	41926011	吴俊豪	89	90	93	0	89	14.5	0	90.14	19				
21	41926065	朱克迪	88	90	91	0	94	14.5	0	90.03	20				
22	41926039	李响	88	90	93	0	88	14.5	0	89.74	21				
23	41926067	丁亚桥	89	90	91	0	89	14.5	0	89.71	22				

欢迎进入学生成绩管理系统!

请输入您的账号: 41926081

请输入您的密码: 41926081

登录成功, 欢迎您: 41926081

get 成绩

姓名: 唐凯悦 学号: 41926081

以下是您的课程成绩:

课程: 高数 学分: 6.0 成绩: 98.00 平均分: 78.37

课程: 大基 学分: 2.5 成绩: 95.00 平均分: 91.12

课程: 英语 学分: 4.0 成绩: 90.63 平均分: 88.70

课程: 选修 学分: 2.0 成绩: 93.75 平均分: 79.55

课程: 化学 学分: 2.0 成绩: 85.00 平均分: 78.92

您修得的总学分为: 16.5 其中选修的学分为: 2.0

您的加权分数为: 93.66

get 大基 排名

大基: 95 排名: 17

get 加权分 排名

加权分: 93.66 排名: 3

exit

感谢您的使用!

八、总结与心得体会

黄德希: 经过本次的学习, 我对面向对象有了更深的认识, 并掌握了许多算法, 在处理 java 题的时候思维也更进一步, 还有实验报告的排版。然后接下来我学习 java 的心得体会: 首先基础是最重要的, 只有基础牢固才有可能对后面的学习有更加深刻的认识。再来就是要有足够的练习, 多多练习才能更好的运用 java 的各种方法, 最后就是要理清思路, 把思路弄清晰比写更多的 java 程序更重要, 然后这次的大作业让我对学好 java 的信心有所增加, 希望在未来的学习中再接再厉。

卢冠男: 这次的大作业让我对 java 的学习又有了新的认识: 通过过这次练习, 我在编程的理论知识上有所巩固和深化, 积累了一笔宝贵的经验。我学会了如何团队合作, 共同开发。独木难支, 光靠一个人的力量是远远不够的。只有管理组织好一个团队, 才能够最终完成项目的开发, 或许在编程技术水平上团队成员有高有低, 但是, 每个人都有自己的闪光点, 总可以在团队的找到适合自己的位置, 每个人在自己的团队中做的贡献, 创造出不可磨灭的重大价值。

最后感谢宋老师一个学期以来的悉心教导!

九、附录

Main类:

```
package managementSystem;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;

public class Main {
    public static List<Student> stuList = new ArrayList<Student>(); //创建
    学生组
    public static int courseNum = 0;
    static {
        FileReader fr = null;
        BufferedReader bfr = null;
        File f = null;
        try {
            fr = new FileReader("importcourses.txt");
            bfr = new BufferedReader(fr);
            String line;
            while((line = bfr.readLine())!=null) {
                String[] s = line.split(",");
                addStudentList(s[0],Double.valueOf(s[1]),s[2]);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                bfr.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    //从importcourses.txt获取需要初始化添加的课程
    for(int i=0;i<stuList.size();i++) {
```

```

LoginData.Register(stuList.get(i).getID(),stuList.get(i).getID());
    }
    //初始化成绩及登录名单
}
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    Login.Login();
    //登录部分
    String cmd = null;
    do {
        cmd = in.nextLine();
        String[] temp = null;
        if (cmd.contains(" ")) {
            temp = cmd.split(" ");
        }else {
            temp =new String[]{cmd};
        }
        if(Login.inAccount.equals("teacher")) {//教师端
            Teacher t = new Teacher();
            switch(temp[0].toLowerCase()) {
                case "set":
                    t.setStudent(temp);
                    break;
                case "import"://添加新科目成绩txt
                    t.importCourse();
                    break;
                case "avg"://获得平均分
                    t.getAvg(temp);
                    break;
                case "get":
                    t.getInfomation(temp);
                    break;
                case "sort"://对某一个科目或加权分进行排序
                    t.sortCourse(temp[1]);
                    break;
                case "export":
                    t.exportCsv();
            }
        }else {//学生端
            String ID = Login.inAccount;
            int t = 0;
            for(int i =0;i<stuList.size();i++) {
                if(stuList.get(i).getID().equals(ID)) {
                    t = i;

```

```

        break;
    }
}
if(temp.length==2) {
    switch(temp[1]) {
        case "成绩":
            System.out.println(stuList.get(t));
    }
} else if(temp.length==3) {
    switch(temp[2]) {
        case "排名":
            stuList.get(t).getRank(temp);
    }
}
}
System.out.println();
}while(!cmd.equals("exit"));
System.out.println("感谢您的使用! ");
in.close();
}

public static void info(String str) {
    System.out.println(str);
}

public static void addStudentList(String courseName,double hour,String
path) {
    courseNum +=1;
    FileReader fr = null;
    BufferedReader br = null;
    File f = new File(path);
    String[][] str;
    int i=0;
    try {
        fr = new FileReader(f);
        br = new BufferedReader(fr);
        String line;
        while(br.readLine()!=null) {
            i++;
        }
        //txt行数读取

        str = new String [i][3];
        i=0;
        fr = new FileReader(f);

```

```

        br = new BufferedReader(fr);
        while((line = br.readLine())!=null) {
            str[i] = line.split("\t");
            i++;
        }
        //读取信息至二维数组
        if(stuList.isEmpty()){
            for(int t=1;t<str.length;t++) {
                Student stu = new Student(str[t][0],str[t][1]);
                stuList.add(stu);
                stuList.get(t-
1).addCourse(courseName,hour,Double.parseDouble(str[t][2]));
            }
        }else{
            for(int t=1;t<str.length;t++) {
                for(int a=0;a<stuList.size();a++) {
                    if(stuList.get(a).getID().equals((str[t][1]))) {

                        stuList.get(a).addCourse(courseName,hour,Double.parseDouble(str[t][2]))
                    }
                }
            }
            for(int t=1;t<str.length;t++) {
                Boolean temp = true;
                for(int a=0;a<stuList.size();a++) {
                    if(stuList.get(a).getID().equals((str[t][1]))) {
                        temp=false;
                    }
                }
                if(temp) {
                    Student stu = new Student(str[t][0],str[t][1]);
                    stuList.add(stu);
                    stuList.get(t-
1).addCourse(courseName,hour,Double.parseDouble(str[t][2]));
                }
            }
        }
        //录入所有学生成绩

        for(int a=0;a<stuList.size();a++) {
            if(stuList.get(a).getCourseList().size()<courseNum) {
                stuList.get(a).addCourse(courseName,hour,0);
            }
        }
    }
}

```

Login类:

13

```

        if(account.equals(LoginData.list.get(i).getAccount())) {
            if(password.equals(LoginData.list.get(i).getPassword()))
        {
            System.out.println("登录成功, 欢迎您:"+account);
            temp = true;
            inAccount =account;
            break;
        }else {
            System.out.println("密码错误");
            break;
        }
    }
}

if((!account.equals(LoginData.list.get(i).getAccount())||!password.equals(LoginData.list.get(i).getPassword()))&&i==LoginData.list.size()-1) {
    System.out.println("登录失败");
}
}
}while(!temp);
}
}

```

LoginData类:

```

package managementSystem;

import java.util.ArrayList;
import java.util.List;

public class LoginData {

    public static List<SignIn> list = new ArrayList<SignIn>();
    //初始化教师登录 账号: teacher 密码: teacher
    static {
        list.add(new SignIn("teacher","teacher"));
    }

    //注册
    public static void Register(String account,String password) {
        SignIn s = new SignIn(account,password);
        list.add(s);
    }
}

//用户账号密码记录

```

```
class SignIn{
    private String account;
    private String password;
    public SignIn() {
        super();
    }
    public SignIn(String account, String password) {
        super();
        this.account = account;
        this.password = password;
    }
    public String getAccount() {
        return account;
    }
    public void setAccount(String account) {
        this.account = account;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

Student类: **package** managementSystem;

```
import java.util.ArrayList;
import java.util.List;
```

```
public class Student extends SortCourse{
    private String name;
    private String ID;
    public List<Course> courseList = new ArrayList<Course>();//科目列表
    private double totalHour;//总学时
    private double electiveCourseHour;//选修学时
    private double weightedScore;//加权分
    public static String sortCourse;//需要排名的课程
    private int rank;
    public Student() {
        super();
    }

    public Student(String name, String id) {
```



```
        super();
        this.name = name;
        ID = iD;
    }

    public String getID() {
        return ID;
    }
    public void setID(String iD) {
        ID = iD;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }

    public List<Course> getCourseList() {
        return courseList;
    }
    //常规的get set方法

    public void addCourse(String name,double hour,double score) {
        Course course = new Course(name,hour,score);
        courseList.add(course);
    }
    //添加课程

    public String getCourseScore(String courseName) {
        StringBuffer sb = new StringBuffer();
        for(int i=0;i<courseList.size();i++) {

            if(courseList.get(i).getScore()!=0&&courseList.get(i).getName().equals(
courseName)) {
                sb.append("课程: "+courseList.get(i).getName()+" \t学分:
"+courseList.get(i).getHour()+"\t 成绩: "+String.format("%.2f",
courseList.get(i).getScore())+" \t 平均分: "+String.format("%.2f",
courseList.get(i).getAverage())+"\n");
                return sb+"";
            }
        }
        return "未查到您该科目的成绩";
    }
}
```

```
}  
//查询单科成绩  
  
public String getCourse() {  
    StringBuffer sb = new StringBuffer();  
    for(int i=0;i<courseList.size();i++) {  
        if(courseList.get(i).getScore()!=0) {  
            sb.append("课程: "+courseList.get(i).getName()+" \t学分:  
"+courseList.get(i).getHour()+"\t 成绩: "+String.format("%.2f",  
courseList.get(i).getScore())+"    平均分: "+String.format("%.2f",  
courseList.get(i).getAverage())+"\n");  
        }  
    }  
    return sb+"";  
}  
//得到所有科目学分及成绩  
  
public String getCourseAverage(String courseName) {  
    StringBuffer sb = new StringBuffer();  
    for(int i=0;i<courseList.size();i++) {  
  
        if(courseList.get(i).getScore()!=0&&courseList.get(i).getName().equals(  
courseName)) {  
            sb.append("课程: "+courseList.get(i).getName()+" \t学分:  
"+courseList.get(i).getHour()+"\t 成绩: "+String.format("%.2f",  
courseList.get(i).getScore())+"    平均分: "+String.format("%.2f",  
courseList.get(i).getAverage())+"\n");  
            return sb+"";  
        }  
    }  
    return "未查到您该科目的成绩";  
}  
//得到单科的平均分  
  
public double getTotalHour() {  
    double temp=0;  
    for(int i=0;i<courseList.size();i++) {  
        if(courseList.get(i).getScore()!=0) {  
            temp += courseList.get(i).getHour();  
        }  
    }  
}
```

```

        this.totalHour = temp;
        return totalHour;
    }
    //得到总学分

    public double getElectiveCourseHour() {
        double temp=0;
        for(int i=0;i<courseList.size();i++) {
            if(courseList.get(i).getName().equals("选修
")&&courseList.get(i).getScore()!=0) {
                temp =courseList.get(i).getHour();
                break;
            }
        }
        this.electiveCourseHour = temp;
        return electiveCourseHour;
    }
    //得到选修学分

    public double getWeightedScore() {
        double temp1=0;
        double temp2=getTotalHour()-getElectiveCourseHour();
        for(int i=0;i<courseList.size();i++) {
            if((!courseList.get(i).getName().equals("选修
"))&&courseList.get(i).getScore()!=0) {
                temp1
+=courseList.get(i).getScore()*courseList.get(i).getHour();
            }
        }
        this.weightedScore = temp1/temp2;
        return weightedScore;
    }
    //得到加权分

    public String toString() {
        return "姓名: "+name+" 学号: "+ID+"\n以下是您的课程成绩:
\n"+getCourse()+"您修得的总学分为: "+getTotalHour()+" 其中选修的学分为:
"+getElectiveCourseHour()+
        "\n您的加权分数为: "+String.format("%.2f",
getWeightedScore());
    }

```



```

    }
    //获取排名

    public String toRow() {
        StringBuffer sb = new StringBuffer();
        for(int i =0;i<Main.stuList.size();i++) {
            if(Main.stuList.get(i).getName().equals(name)) {
                this.rank = i+1;
            }
        }
        for(int i=0;i<courseList.size();i++) {
            sb.append(","+String.format("%.0f",
courseList.get(i).getScore()));
        }
        return ID + "," + name + sb+ "," +
getTotalHour()+","+getElectiveCourseHour()+","+String.format("%.2f",
getWeightedScore()+","+rank;
    }
}

```

Course类:

```
package managementSystem;
```

```

public class Course {
    private String name;
    private double hour;
    private double score;
    public double average;

    public Course() {
        super();
    }
    public Course(String name, double score) {
        super();
        this.name = name;
        this.score = score;
    }
    public Course(String name, double hour, double score) {
        super();
        this.name = name;
        this.hour = hour;
        this.score = score;
    }
    public String getName() {

```

```

        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getHour() {
        return hour;
    }
    public void setHour(double hour) {
        this.hour = hour;
    }
    public double getScore() {
        return score;
    }
    public void setScore(double score) {
        this.score = score;
    }
    //常规的get、set方法

    public double getAverage() {
        double temp1=0;//统计总分数
        int temp2=0;//统计总人数
        for(int i=0;i<Main.stuList.size();i++) {
            for(int t=0;t<Main.stuList.get(i).getCourseList().size();t++) {

                if(Main.stuList.get(i).getCourseList().get(t).getName().equals(name)&&Main.stuList.get(i).getCourseList().get(t).getScore()!=0) {

                    temp1+=Main.stuList.get(i).getCourseList().get(t).getScore();
                    temp2+=1;
                }
            }
        }
        this.average=temp1/temp2;
        return average;
    }
}

```

Teacher类:

```

package managementSystem;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;

```

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.util.Collections;
import java.util.Scanner;
import java.awt.Desktop;

public class Teacher extends SortCourse{

    public void exportCsv(){
        File csvFile = null;
        BufferedWriter csvWriter = null;
        try {
            csvFile = new File("成绩单.csv");
            csvFile.createNewFile();
            csvWriter = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream(csvFile), "GB2312"), 1024);
            StringBuffer sb = new StringBuffer();
            for(int t=0;t<Main.stuList.get(0).getCourseList().size();t++)
{

                sb.append(", "+Main.stuList.get(0).getCourseList().get(t).getName());
            }
            csvWriter.write("学号,姓名"+sb+",总学时,选修学时,加权分,排名");
            csvWriter.newLine();
            for(Student stu :Main.stuList) {
                csvWriter.write(stu.toRow());
                csvWriter.newLine();
            }
            csvWriter.flush();
            System.out.println("导出成功! ");
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            try {
                csvWriter.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

        try {
            // 打开当前文件
            Desktop.getDesktop().open(new
File(csvFile.getAbsolutePath()));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    //导出成绩

    public void getInfomation(String[] temp) {
        Boolean a = false;
        Boolean b = true;
        if (temp.length == 3) {
            for (int i = 0; i < Main.stuList.size(); i++) {
                if (temp[1].equals(Main.stuList.get(i).getName())
                    || temp[2].equals(Main.stuList.get(i).getName()))
                {
                    for (int t = 0; t <
Main.stuList.get(i).getCourseList().size(); t++) {
                        if
(Main.stuList.get(i).getCourseList().get(t).getScore() != 0
                            &&
(temp[1].equals(Main.stuList.get(i).getCourseList().get(t).getName())
                                ||
temp[2].equals(Main.stuList.get(i).getCourseList().get(t).getName())))
                        {
                            a = true;
                        }
                    }
                }
            }
            if (temp[2].equals("加权分")) {
                b = false;
                for (int i = 0; i < Main.stuList.size(); i++) {
                    if (temp[1].equals(Main.stuList.get(i).getName()) ||
temp[1].equals(Main.stuList.get(i).getID())) {
                        System.out.println("姓名: " +
Main.stuList.get(i).getName() + "\t学号: " + Main.stuList.get(i).getID()
                            + "\n已修学科如下: ");
                        for (int t = 0; t <
Main.stuList.get(i).getCourseList().size(); t++) {
                            if
(Main.stuList.get(i).getCourseList().get(t).getScore() != 0) {

```



```

        System.out.print(Main.stuList.get(i).getCourseList().get(t).getName(
) + ":"
                                + String.format("%.0f",
Main.stuList.get(i).getCourseList().get(t).getScore())
                                + "\t");
    }
}
    System.out.println("加权分: " +
String.format("%.2f",Main.stuList.get(i).getWeightedScore()));
}
}
} else if (temp[2].equals("不及格")) {
    int num =0;//不及格人数
    b =false;
    for (int i = 0; i < Main.stuList.size(); i++) {
        for (int t = 0; t <
Main.stuList.get(i).getCourseList().size(); t++) {
            if
(temp[1].equals(Main.stuList.get(i).getCourseList().get(t).getName())
                &&
Main.stuList.get(i).getCourseList().get(t).getScore() != 0
                &&
Main.stuList.get(i).getCourseList().get(t).getScore() <= 60) {
                num+=1;
            }
        }
    }
    //判断是否有不及格

    if(num!=0) {
        System.out.println(temp[1]+"的不及格人数为: "+num);
        System.out.println("姓名\t学号\t
成绩");
        for (int i = 0; i < Main.stuList.size(); i++) {
            for (int t = 0; t <
Main.stuList.get(i).getCourseList().size(); t++) {
                if
(temp[1].equals(Main.stuList.get(i).getCourseList().get(t).getName())
                    &&
Main.stuList.get(i).getCourseList().get(t).getScore() != 0
                    &&
Main.stuList.get(i).getCourseList().get(t).getScore() <= 60) {

```

```

        System.out.println(Main.stuList.get(i).getName() + "\t" +
Main.stuList.get(i).getID() + "\t"
                                + String.format("%.0f",
Main.stuList.get(i).getCourseList().get(t).getScore())
                                + "\t");

    }
    }
    }
    }else {
        System.out.println("您好, "+temp[1]+"没有不及格人员");
    }
} else if(a&&b){
    for (int i = 0; i < Main.stuList.size(); i++) {
        if (temp[1].equals(Main.stuList.get(i).getName())
            ||
temp[2].equals(Main.stuList.get(i).getName())) {
            System.out.print("姓名: " +
Main.stuList.get(i).getName() + "\t 学号: "
                                + Main.stuList.get(i).getID() + "    ");
            b = false;
            for (int t = 0; t <
Main.stuList.get(i).getCourseList().size(); t++) {
                if
(Main.stuList.get(i).getCourseList().get(t).getScore() != 0
                    &&
(temp[1].equals(Main.stuList.get(i).getCourseList().get(t).getName())
                        ||
temp[2].equals(Main.stuList.get(i).getCourseList().get(t).getName())))
                {

                    System.out.println(Main.stuList.get(i).getCourseList().get(t).getNam
e() + ":"
                                + String.format("%.0f",
Main.stuList.get(i).getCourseList().get(t).getScore()));
                }
            }
        }
    }
}
}
}
if(b) {
    System.out.println("您输入的指令有误! ");
}

```

```

    }
    //get 获取信息

    public void setStudent(String temp[]) {
        Boolean b = true;
        if(temp.length==3) {
            Student stu = new Student(temp[1],temp[2]);
            Main.stuList.add(stu);
            System.out.println("已添加学生: "+temp[1]+"      "+temp[2]);
            for(int t=0;t<Main.stuList.get(0).getCourseList().size();t++)
        {
            Main.stuList.get(Main.stuList.size()-
1).addCourse(Main.stuList.get(0).getCourseList().get(t).getName(),
Main.stuList.get(0).getCourseList().get(t).getHour(), 0);
            b = false;
            //初始化成绩都为0
        }
        }else if(temp.length==4) {
            try {
                double score = Double.valueOf(temp[3]);
                for1:for(int i=0;i<Main.stuList.size();i++) {

                    if(temp[1].equals(Main.stuList.get(i).getName())||temp[1].equals(Main
.stuList.get(i).getID())) {
                        for(int
t=0;t<Main.stuList.get(i).getCourseList().size();t++) {

                            if(temp[2].equals(Main.stuList.get(i).getCourseList().get(t).getName
())) {

                                Main.stuList.get(i).getCourseList().get(t).setScore(score);
                                System.out.println("学生:
"+Main.stuList.get(i).getName()+" 课程: "+temp[2]+" 分数已设为:
"+temp[3]);

                                    b = false;
                                    break for1;
                                }
                            }
                        }
                    }
                }
            }catch(NumberFormatException e){
                //加入保险机制, 防止输入非double型
            }
        }
    }
}

```

```

        if(b) {
            System.out.println("您输入的指令有误! ");
        }
    }
    //设定学生

    public void getAvg(String temp[]) {
        Boolean b = true;
        if(temp.length==2) {
            for(int i=0;i<Main.stuList.get(0).getCourseList().size();i++)
{
                if(temp[1].equals(Main.stuList.get(0).getCourseList().get(i).getName
                ())) {
                    System.out.println("课程: "+temp[1]+"\t平均分:
                    "+String.format("%.2f",Main.stuList.get(0).getCourseList().get(i).getAv
                    erage()));
                    b = false;
                }
            }
            for(int i=0;i<Main.stuList.size();i++) {

                if(temp[1].equals(Main.stuList.get(i).getName())||temp[1].equals(Mai
                n.stuList.get(i).getID())){
                    b = false;
                    int temp1 = 0;
                    int temp2 = 0;
                    System.out.println("姓名:
                    "+Main.stuList.get(i).getName()+"\t学号:
                    "+Main.stuList.get(i).getID()+"\n已修学科如下: ");
                    for(int
                    t=0;t<Main.stuList.get(i).getCourseList().size();t++) {

                        if(Main.stuList.get(i).getCourseList().get(t).getScore()!=0) {
                            temp1 +=
                            Main.stuList.get(i).getCourseList().get(t).getScore();
                            temp2 += 1;

                            System.out.print(Main.stuList.get(i).getCourseList().get(t).getName(
                            )+": "+String.format("%.0f",
                            Main.stuList.get(i).getCourseList().get(t).getScore())+"\t");
                        }
                    }
                }
            }
        }
    }
}

```

```

        System.out.println("平均分: "+(temp1/temp2));
    }
}
}
if(b) {
    System.out.println("您输入的指令有误! ");
}
}
//得到平均分

public void importCourse() {
    Scanner in = new Scanner(System.in);
    System.out.print("请输入课程名: ");
    String temp1 = in.next();
    boolean b = false;
    for(int i=0;i<Main.stuList.get(0).getCourseList().size();i++) {

        if(Main.stuList.get(0).getCourseList().get(i).getName().equals(temp1
    )) {
            b = true;
        }
    }
    if(b) {
        System.out.print("请输入课程学分: ");
        double temp2 = in.nextDouble();
        System.out.print("请输入文本路径: ");
        String temp3 = in.next();
        String temp4 = null;
        if(temp3.endsWith(".txt")&&temp3.contains("\\")) {
            temp4 = temp3.replace("\\", "\\");
            File f1 = null;
            File f2 = null;
            FileInputStream fis = null;
            FileOutputStream fos = null;
            FileWriter fw = null;
            PrintWriter pw = null;
            try {
                f1 = new File(temp4);
                fis = new FileInputStream(f1);
                fos = new FileOutputStream(temp1+".txt");
                int data;
                while((data = fis.read())!=-1) {
                    fos.write(data);
                }
            }
        }
    }
}

```

```

        Main.addStudentList(temp1,temp2,temp1+".txt");
        System.out.println("已添加课程: "+temp1);
        f2 = new File("importcourses.txt");
        fw = new FileWriter(f2,true);
        pw = new PrintWriter(fw);
        pw.print("\n"+temp1+","+temp2+","+temp1+".txt");
        pw.flush();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }finally {
        try {
            fis.close();
            fos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
} else {
    System.out.println("路径格式不正确");
}
} else {
    System.out.println("该课程已经添加! ");
}
}
//添加新课程成绩

public void sortCourse(String course) {
    Boolean b = true;

    if(course.equals("加权分")) {
        rankCourse("加权分");
        b = false;
    } else {
        for(int i
=0;i<Main.stuList.get(0).getCourseList().size();i++) {

            if(course.equals(Main.stuList.get(0).getCourseList().get(i).getName(
))) {

                rankCourse(course);
                b = false;
            }
        }
    }
}

```

```

        if(b){
            System.out.println("未找到"+course);
        }
    }
}
//排序对应科目

public void rankCourse(String courseName) {
    sortCourse = courseName;
    Collections.sort(Main.stuList, new sortRule());
    System.out.print("名次\t姓名\t学号\t");
    for (int a = 0; a < Main.stuList.get(0).getCourseList().size();
a++) {
        System.out.print("\t" +
Main.stuList.get(0).getCourseList().get(a).getName());
    }
    System.out.println("\t总学分\t加权分");
    for (int i = 0; i < Main.stuList.size(); i++) {
        System.out.print((i + 1) + "\t" +
Main.stuList.get(i).getName() + "\t" + Main.stuList.get(i).getID());
        for (int t = 0; t <
Main.stuList.get(i).getCourseList().size(); t++) {
            System.out.print("\t" + String.format("%.0f",
Main.stuList.get(i).getCourseList().get(t).getScore()));
        }
        System.out.print("\t" + Main.stuList.get(i).getTotalHour() +
"\t"
            + String.format("%.2f",
Main.stuList.get(i).getWeightedScore()));
        System.out.println();
    }
}
}
}

```

SortCourse类:

```

package managementSystem;

import java.util.Collections;
import java.util.Comparator;

```

```

public class SortCourse {
    public static String sortCourse;

    public void rank(String courseName) {
        sortCourse = courseName;
        Collections.sort(Main.stuList, new sortRule());
    }
}

```

//排序规则：分数从高到低排，分数相同则按学号从低到高排

```

class sortRule implements Comparator{

    @Override
    public int compare(Object o1, Object o2) {
        Student stu1 = (Student) o1;
        Student stu2 = (Student) o2;

        for(int i=0;i<stu1.courseList.size();i++) {
            if(SortCourse.sortCourse.equals("加权分")) { //加权分的排名
                if(stu1.getWeightedScore()>stu2.getWeightedScore()) {
                    return -1;
                }else if(stu1.getWeightedScore()==stu2.getWeightedScore()) {

                    if(Integer.parseInt(stu1.getID())<Integer.parseInt(stu2.getID())) {
                        return -1;
                    }else {
                        return 1;
                    }
                }else {
                    return 1;
                }
            }else {

                if(stu1.courseList.get(i).getName().equals(SortCourse.sortCourse)) { //
                某课程的排名

                    if(stu1.courseList.get(i).getScore()>stu2.courseList.get(i).getScore())
                    {
                        return -1;
                    }else
                    if(stu1.courseList.get(i).getScore()==stu2.courseList.get(i).getScore()) {

                        if(Integer.parseInt(stu1.getID())<Integer.parseInt(stu2.getID())) {

```



```
        return -1;
    }else {
        return 1;
    }
    }else {
        return 1;
    }
    }
    }
    }
    return 0;
}
}
```