

大 連 工 業 大 学

设 计 性 实 验 报 告

题 目： 课程综合项目：学生成绩管理系统

专业： 计算机科学与技术

班级学号： 计算机 182-07

学生姓名： 梁宇龙

指导教师： 苏波

实验室名称： 综 B322

所属学院： 信息科学与工程学院

实验日期： 2020 年 5 月 22 日

课程综合项目：学生成绩管理系统

1 实验内容设计

本项目基于本课程实验基础上进行。

程序应用场景：模拟教务处成绩管理系统控制台程序

程序功能：学生端完成该学生各学科成绩查询和 GPA 计算功能，教师端完成学生成绩录入和修改功能。

用户使用流程（基于文件读写进行）：

（1）学生端成绩查询：学生输入姓名→系统进入控制台检索（若学生不存在，提示并返回首页）→若学生存在，提示输入密钥（密钥不正确进行提示，并返回主页）→密钥正确，系统检索该生所修全部课程并输出课程名称及成绩，并计算 GPA（平均成绩加权绩点）→返回首页

（2）教师端成绩录入与修改：教师输入工号→工号存在，检索教师姓名进行问好，并提示输入密钥（不存在提示，返回首页）→密钥正确，提示输入课程号→检索该课程是否为该老师授课→若该学科是该老师授课则进入修改成绩界面（不是则提示：您无修改权限）→输入学生姓名，平时成绩和期末成绩→修改完成，返回首页

2 程序记录

（未记录文件读写方法 FileTool）

//主函数，进入程序

```
public class OOP_Experiment {  
    public static void main(String[] args) {  
        System.out.println("请选择服务项目");  
        Service.serviseTip();  
        int service = Console.inputInt();  
        Service.servise(service);  
    }  
}
```

//服务部分

```
public class Service {  
    //从文件中获取学生信息
```

```

public static Student SearchStu(){
    System.out.println("\n 请输入学生姓名:");
    String StuName = Console.inputString();
    Student stu = Manager.stuManager(StuName);
    return stu;
}
//服务界面
public static void servise(int service){
    while(service!=0){
        switch (service) {
            case 1:
                Student stu1 = SearchStu();
                if(stu1==null){
                    System.out.println("\n 无该学生成绩! 请重新选择操作: ");
                    serviseTip();
                    service = Console.inputInt();
                }
                else{
                    System.out.println("\n 请输入查询密钥: ");
                    String password = Console.inputString();
                    if(stu1.getPassword().equals(password)){
                        System.out.println("\n 该学生成绩为: ");
                        Manager.stuGrade(stu1.getStuNum());
                        GPAManager.getGPA(stu1);
                        System.out.println("\n 查询结束, 请选择操作: ");
                        serviseTip();
                        service = Console.inputInt();
                    }
                    else{
                        System.out.println("\n 密钥错误, 请重新选择操作: ");
                        serviseTip();
                        service = Console.inputInt();
                    }
                }
            break;
            case 2:

```

```

        System.out.println("\n 请输入您的工号: ");
        String teacherNum = Console.inputString();
        Teacher teacher1 = Manager.getTeacher(teacherNum)
;

        if(teacher1!=null){
            System.out.println("\n"+teacher1.getTeacherName()+"老师您好, 请输入您的密码: ");
            String password = Console.inputString();
            if(password.equals(teacher1.getPassword())){
                int tf = ChangeGrade(teacher1.getTeacherNum());

                if(tf == 1){
                    System.out.println("\n 更改成功, 请选择操作: ");

                    serviseTip();
                    service = Console.inputInt();
                }
                else if(tf==2){
                    System.out.println("\n 您无权修改该学科成绩, 请选择操作: ");

                    serviseTip();
                    service = Console.inputInt();
                }
                else{
                    System.out.println("\n 更改失败, 请选择操作: ");

                    serviseTip();
                    service = Console.inputInt();
                }
            }
            else{
                System.out.println("\n 密码输入错误, 请选择操作: ");

                serviseTip();
                service = Console.inputInt();
            }
        }
        else{

```

```

        System.out.println("\n 未查询到您的信息，请选择操
作：");

        serviseTip();
        service = Console.inputInt();
    }
    break;
default:
    System.out.println("\n 非法输入，请重新选择");
    serviseTip();
    service = Console.inputInt();
    break;
}
}
System.out.println("正在退出系统...");
}
//服务提示界面
public static void serviseTip(){
    System.out.println("1.成绩查询");
    System.out.println("2.成绩修改");
    System.out.println("0.退出系统");
}
//更改成绩函数
public static int ChangeGrade(String teacherNum){
    int tf;
    System.out.println("\n 请输入课程号");
    String CourseNum = Console.inputString();
    if(teacherNum.equals(Manager.getAimTeacher(CourseNum))){
        Student stu2 = SearchStu();
        System.out.println("\n 请输入平时成绩");
        int grade1 = GradeConsole.inputInt();
        System.out.println("\n 请输入期末成绩");
        int grade2 = GradeConsole.inputInt();
        int grade = (int)(grade1 * 0.4 + grade2 * 0.6);
        double Gredit;
        if(grade>=90){
            Gredit = 4;
        }
        else if(grade>=80&&grade<90){
            Gredit = 3;

```

```

    }
    else if(grade>=70&&grade<80){
        Gredit = 2;
    }
    else if(grade>=60&&grade<70){
        Gredit = 1;
    }
    else{
        Gredit = 0;
    }
    tf = Manager.ChangeGrade(stu2.getStuNum(),CourseNum,grade
,Gredit);
    }
    else{
        tf=2;
    }
    return tf;
}
}

```

//构建学生类

```

public class Student {
    private String stuNum;           //学号
    private String stuName;          //学生姓名
    private String password;
    //构造方法
    public Student(){

    }
    public Student(String stuNum){
        this.stuNum = stuNum;
    }
    public Student(String stuNum,String stuName){
        this.stuNum = stuNum;
        this.stuName = stuName;
    }
    public Student(String stuNum,String stuName,String password){
        this.stuNum = stuNum;
        this.stuName = stuName;
        this.password = password;
    }
}

```

```

    }
    //getter 和 setter 方法
    public String getStuNum(){
        return stuNum;
    }
    public String getStuName(){
        return stuName;
    }
    public String getPassword(){
        return password;
    }
    public void setStuNum(String stuNum){
        this.stuNum = stuNum;
    }
    public void setStuName(String stuName){
        this.stuName = stuName;
    }
}
//构建老师类
public class Teacher {
    private String teacherNum;
    private String teacherName;
    private String password;
    public Teacher(String teacherNum, String teacherName, String password){
        this.teacherNum = teacherNum;
        this.teacherName = teacherName;
        this.password = password;
    }
    public String getTeacherNum(){
        return teacherNum;
    }
    public String getTeacherName(){
        return teacherName;
    }
    public String getPassword(){
        return password;
    }
}

```

//构建课程类

```
public class Course {
    private String courseNum;
    private String courseName;
    private String teacher;
    private double credit;
    //构造方法
    public Course(String courseNum){
        this.courseNum = courseNum;
    }
    public Course(String courseNum,String courseName,double credit){
        this.courseNum = courseNum;
        this.courseName = courseName;
        this.credit = credit;
    }
    public Course(String courseNum,String courseName,String teacher,d
ouble credit){
        this.courseNum = courseNum;
        this.courseName = courseName;
        this.teacher = teacher;
        this.credit = credit;
    }
    //各个变量的 getter 和 setter
    public String getCourseNum(){
        return courseNum;
    }
    public String getCourseName(){
        return courseName;
    }
    public double getCredit(){
        return credit;
    }
    public String getTeacherNum(){
        return teacher;
    }
    public void setCourseNum(String courseNum){
        this.courseNum = courseNum;
    }
    public void setCourseName(String courseName){
```



```

        this.courseName = courseName;
    }
    public void setCredit(double credit){
        this.credit = credit;
    }
}
//输入整型和字符型数值方法
import java.util.Scanner;
public class Console {
    public static String inputString(){
        Scanner scanner = new Scanner(System.in);
        String str = scanner.next();
        return str;
    }
    public static int inputInt(){
        Scanner scanner = new Scanner(System.in);
        int num = scanner.nextInt();
        return num;
    }
}
//通过重写 Console 中输入整型数值方法，使输入成绩时保证成绩在 0~100 间。
public class GradeConsole extends Console{
    static int num;
    @Override
    public static int inputInt(){
        num = super.inputInt();
        while(num<0||num>100){
            System.out.println("输入信息有误，请核对后重新输入：");
            num = super.inputInt();
        }
        return num;
    }
}
//构建信息检索相关方法
public class Manager {
    static Student[] students;
    static SC[] scs;
    static Course[] courses;
    //通过学生姓名检索该学生对象

```

```

public static Student stuManager(String StuName){
    String content = FileTool.read("stu-lab.txt");
    String[] lines = content.split("\r\n");
    students = new Student[lines.length];
    for(int i=0;i<lines.length;i++){
        String[] items = lines[i].split(",");
        students[i] = new Student(items[0],items[1],items[2]);
        if(students[i].getStuName().equals(StuName)){
            return students[i];
        }
    }
    return null;
}
//通过学生学号检索学生各学科成绩
public static void stuGrade(String StuNum){
    String content = FileTool.read("SC-lab.txt");
    String[] lines = content.split("\r\n");
    scs = new SC[lines.length];
    for(int i=0;i<lines.length;i++){
        String[] items = lines[i].split(",");
        scs[i] = new SC(items[0],items[1],Integer.parseInt(items[
2]),Double.valueOf(items[3]));
        if(scs[i].getSNo().equals(StuNum)){
            System.out.println(getCourseName(scs[i].getCNo())+"\t
"+scs[i].getGrade());
        }
    }
}
//通过课程号检索课程名称
public static String getCourseName(String CourseNum){
    String content = FileTool.read("Course-lab.txt");
    String[] lines = content.split("\r\n");
    courses = new Course[lines.length];
    for(int i=0;i<courses.length;i++){
        String[] items = lines[i].split(",");
        courses[i] = new Course(items[0],items[1],items[2],Double
.valueOf(items[3]));
        if(courses[i].getCourseNum().equals(CourseNum)){
            return courses[i].getCourseName();

```

```

    }
}
return null;
}
//修改成绩函数，成功返回值为1，不成功返回数值0
public static int ChangeGrade(String StuNum,String CourseNum,int
Grade,double Gredit){
    int tf = 0;
    String content = FileTool.read("SC-lab.txt");
    String[] lines = content.split("\r\n");
    scs = new SC[lines.length];
    for(int i=0;i<lines.length;i++){
        String[] items = lines[i].split(",");
        scs[i] = new SC(items[0],items[1],Integer.parseInt(items[
2]),Double.valueOf(items[3]));
        if(scs[i].getSNo().equals(StuNum)){
            if (scs[i].getCNo().equals(CourseNum)) {
                scs[i].setGrade(Grade);
                scs[i].setGredit(Gredit);
                tf = 1;
            }
        }
    }
    String contents = "";
    for (SC sc : scs) {
        contents += sc.getSNo() + "," + sc.getCNo() + "," + sc.ge
tGrade() + "," + sc.getGredit() + "\r\n";
    }
    if(tf==0){
        contents += StuNum+","+CourseNum+","+Grade+","+Gredit+"\r
\n";
        tf=1;
    }
    FileTool.write("SC-lab.txt", contents);
    return tf;
}
public static Teacher[] teachers;
//通过老师工号检索该老师对象。
public static Teacher getTeacher(String TeacherNum){

```

```

String content = FileTool.read("Teacher-lab.txt");
String[] lines = content.split("\r\n");
teachers = new Teacher[lines.length];
for(int i=0;i<teachers.length;i++){
    String[] items = lines[i].split(",");
    teachers[i] = new Teacher(items[0],items[1],items[2]);
    if(teachers[i].getTeacherNum().equals(TeacherNum)){
        return teachers[i];
    }
}
return null;
}
//通过课程号检索该课程授课教师工号
public static String getAimTeacher(String CourseNum){
    String content = FileTool.read("Course-lab.txt");
    String[] lines = content.split("\r\n");
    courses = new Course[lines.length];
    for(int i=0;i<courses.length;i++){
        String[] items = lines[i].split(",");
        courses[i] = new Course(items[0],items[1],items[2],Double
        .valueOf(items[3]));
        if(courses[i].getCourseNum().equals(CourseNum)){
            return courses[i].getTeacherNum();
        }
    }
    return null;
}
}
import java.math.*;
//通过继承 Manager 类中相关方法及变量完成计算学生 GPA 的操作
public class GPAManager extends Manager{
    static Student stu1 = new Student();
    static double gpa;
    public static void getGPA(Student stu1){
        gpa = figureGPA(stu1.getStuNum());
        //实现绩点保留小数点后 2 位数
        BigDecimal b = new BigDecimal(gpa);
        double f1 = b.setScale(2, BigDecimal.ROUND_HALF_UP).doubleValue();
    }
}

```

```

        System.out.println("该生平均成绩绩点为: "+ f1);
    }
    //计算出该学生 GPA
    public static double figureGPA(String StuNum){
        double sum = 0,ave = 0;
        String content = FileTool.read("SC-lab.txt");
        String[] lines = content.split("\r\n");
        scs = new SC[lines.length];
        for(int i=0;i<lines.length;i++){
            String[] items = lines[i].split(",");
            scs[i] = new SC(items[0],items[1],Integer.parseInt(items[
2]),Double.valueOf(items[3]));
            if(scs[i].getSNo().equals(StuNum)){
                sum += scs[i].getGredit() * getCourseCredit(scs[i].ge
tCNo());
                ave += getCourseCredit(scs[i].getCNo());
            }
        }
        sum = sum/ave;
        return sum;
    }
    //获取课程所对应学分
    public static double getCourseCredit(String CourseNum){
        String content = FileTool.read("Course-lab.txt");
        String[] lines = content.split("\r\n");
        courses = new Course[lines.length];
        for(int i=0;i<courses.length;i++){
            String[] items = lines[i].split(",");
            courses[i] = new Course(items[0],items[1],Double.valueOf(
items[2]));
            if(courses[i].getCourseNum().equals(CourseNum)){
                return courses[i].getCredit();
            }
        }
        return 0;
    }
}

```

3 结果记录

run:

请选择服务项目

1.成绩查询

2.成绩修改

0.退出系统

1

请输入学生姓名:

Xianyu

请输入查询密钥:

1805010104

进行密码试错操作

密钥错误, 请重新选择操作:

1.成绩查询

2.成绩修改

0.退出系统

1

请输入学生姓名:

Yulong

请输入查询密钥:

1805010207

该学生成绩为:

大学物理 2 78

电工学 90

离散数学 83

数据结构 95

该生平均成绩绩点为: 3.29

学生成绩查询操作

查询结束, 请选择操作:

1.成绩查询

2.成绩修改

0.退出系统

2

请输入您的工号:

01

Liuna 老师您好, 请输入您的密码:

099

密码试错

密码输入错误, 请选择操作:

1.成绩查询

2.成绩修改

0.退出系统

2

请输入您的工号:

01

Liuna 老师您好, 请输入您的密码:

csliuna01

请输入课程号

01

修改学科权限试错

您无权修改该学科成绩, 请选择操作:

1.成绩查询

2.成绩修改

0.退出系统

2

请输入您的工号:

01

Liuna 老师您好, 请输入您的密码:

csliuna01

请输入课程号

099

请输入学生姓名:

Yulong

请输入平时成绩

98

请输入期末成绩

91

修改成绩操作

更改成功，请选择操作：

1.成绩查询

2.成绩修改

0.退出系统

1

请输入学生姓名：

Yulong

请输入查询密钥：

1805010207

该学生成绩为：

大学物理 2 78

电工学 90

离散数学 83

数据结构 93

数据结构成绩从 95 分改为
93 分

该生平均成绩绩点为：3.29

查询结束，请选择操作：

1.成绩查询

2.成绩修改

0.退出系统

0

正在退出系统...

成功构建 (总时间: 1 分钟 30 秒)

4 结论

本次课程项目，利用 Java 语言编写模拟教务处成绩管理系统的控制台程序，通过类的概念对学生，老师，课程，及学生-课程-成绩进行封装，并写出相应方法，通过对方法的继承和改写进一步扩充程序功能，通过文件读写方法进行信息查询检索和修改功能。

得到结果如下：

在学生端程序中，验证学生姓名和密钥，可以查询该生已修课程成绩并计算 GPA。

在教师端程序中，验证老师工号，密钥，并确定有权修改该学科成绩后可对某学生成绩进行写入和修改。

结果表明：

在 Java 控制台程序中，可以运用文件读写进行对象检索和相应值的修改。

项目未来设想：

在本项目基础上仍可继续进行完善：

- ①可以将密钥检测，修改权限检错用 try-catch 方法实现。
 - ②可以将文件读写转换成对数据库的查询和修改，也可以建立服务器将系统上传，在终端安装客户端进行操作。
 - ③可使用前端开发或移动端开发建立 UI 界面，实现交互操作。
- 我会将本项目在未来继续完善，并证据利用到实际生活中。