

电子科技大学信息与软件工程学院

实 验 报 告

(五)

学 号 2018091618008

姓 名 袁昊男

(实验) 课程名称 面向对象程序设计 Java

理论教师 周帆

实验教师 何中海

电子科技大学

实验报告

学生姓名：袁昊男 学号：2018091618008 指导教师：周帆

实验地点：信软楼 304

实验时间：2019. 11. 24、12. 01

一、 实验名称：学生成绩管理系统设计与实现实验

二、 实验学时：8 学时

三、 实验目的：

通过实现对“学生成绩管理系统”设计和编程，锻炼学生对对象模型设计及软件模型的理解和分析能力实验。

四、 实验原理：

1、 面向对象

Java 语言具有真正的面向对象语言的特点，除了数值、布尔和字符三种基本的数据类型外，其它类型都是对象。它支持封装、多态性和继承。封装就是将对象内的数据和代码联编起来，形成一个对象；多态性是指一个接口，有多个内在实现形式表示；继承是指某一对象直接使用另一对象的所有属性和方法的过程。

对程序员来说，这意味着要注意应中的数据 and 操纵数据的方法（method），而不是严格地用过程来思考。在一个面向对象的系统中，类（class）是数据和操作数据的方法的集合。数据和方法一起描述对象（object）的状态和行为。每一对象是其状态和行为的封装。类是按一定体系和层次安排的，使得子类可以从超类继承行为。在这个类层次体系中有一个根类，它是具有一般行为的类。Java 程序是用类来组织的。

2、 编译和解释性

Java 编译程序生成字节码（byte-code），而不是通常的机器码。Java 字节码提供对体系结构中性的目标文件格式，代码设计成可有效地传送程序到多个平台。Java 程序可以在任何实现了 Java 解释程序和运行系统（run-time system）的系统上运行。

在一个解释性的环境中，程序开发的标准“链接”阶段大大消失了。如果说 Java 还有一个链接阶段，它只是把新类装进环境的过程，它是增量式的、轻量级的过程。因此，Java 支持快速原型和容易试验，它将导致快速程序开发。这是一个与传统的、耗时的“编译、链接和

测试”形成鲜明对比的精巧的开发过程。

五、 实验内容：

1、 对象及属性的创建

学生成绩管理系统中所涉及的对象及属性有：

- (1) 学生基本信息：学号、姓名、性别、出生年月、学院、专业；
- (2) 教师基本信息：职工编号、姓名、学院、系；
- (3) 课程基本信息：课程编号、课程名称、学分、学时数；
- (4) 成绩基本信息：成绩单号、课程、教师、学生、成绩；
- (5) 教务员基本信息：职工编号、姓名、学院。

2、 对象信息的获得与修改

基于对学生成绩管理系统功能分析，创建对象的行为方法，实现学生成绩管理系统业务功能。

学生、教师、课程、成绩、教务员对象行为有：

- (1) 信息编辑；
- (2) 登录账号的维护；
- (3) 信息查询；
- (4) 综合统计、排名及报表打印。

3、 系统角色

系统的使用者通过各功能实现相关功能。其中使用者有：

- (1) 学生：个人信息维护、课程及成绩查询；
- (2) 教师：个人信息维护、课程及成绩登录；
- (3) 教务员：个人帐号维护、课程管理、综合统计、排名及报表打印；
- (4) 系统管理员：系统及数据管理。

六、 实验器材（设备、元器件）：

配置了 JDK 环境、安装有 Eclipse 软件的个人电脑一台。

七、 实验步骤：

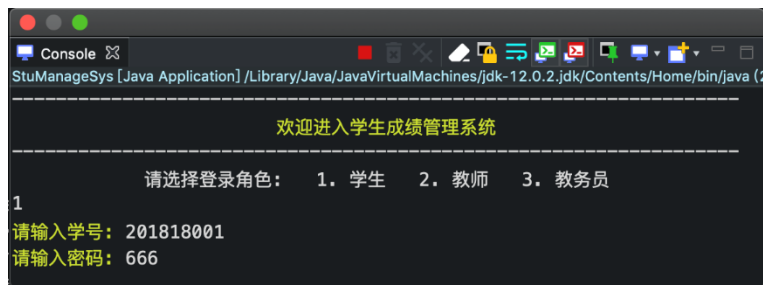
- 1、 分析所要编写的学生成绩管理系统的各功能需求，确定系统的基本框架和编写思路；
- 2、 根据系统需求，建立学生、教师、课程、成绩对象及其相关属性；
- 3、 基于对学生成绩管理系统功能分析，创建对象的行为方法，实现学生成绩管理系统业务功能；
- 4、 对系统进行测试，重点测试各系统角色的登录与相关功能的实现，以及文件流的读写；
- 5、 分析实验结果，撰写实验报告。

八、实验结果与分析（含重要数据结果分析或核心代码流程分析）

1、系统运行截图

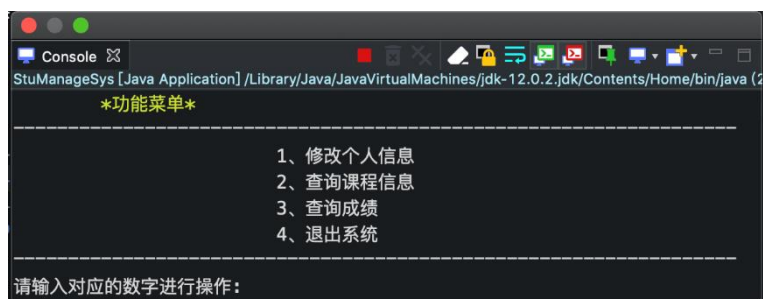
(1) 学生角色

a) 学生登录界面

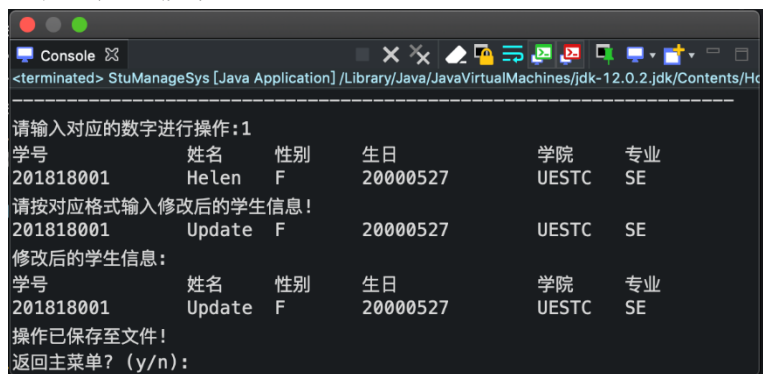


（注：为测试系统功能，此处默认登录学生学号为 201818001）

b) 学生菜单界面



c) 学生个人信息修改界面



d) 修改后输出至 Student.txt 文件保存

学号	姓名	性别	生日	学院	专业
201818001	Update	F	20000527	UESTC	SE
201818003	Tommy	M	20000627	UESTC	SE
201818004	Dean	M	20001011	UESTC	SE
201818005	George	M	20000329	UESTC	SE
201818006	Oscar	M	20000117	UESTC	SE
201818007	Taylor	F	20000209	UESTC	SE
201818008	David	M	20000808	UESTC	SE

e) 学生查找课程信息界面



f) 学生查询成绩界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2
请输入对应的数字进行操作:3
成绩单号      学号      姓名      课程      教师      成绩
1111001      201818001      Peter      JavaProg      Zhou      68
1111002      201818001      Peter      EnglishB      Li      65
1111003      201818001      Peter      Calculus      Jia      65
返回主菜单? (y/n):
y

```

(2) 教师角色

a) 教师登录界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2
-----
                欢迎进入学生成绩管理系统
-----
                请选择登录角色:   1. 学生   2. 教师   3. 教员
2
请输入教工号: 1001
请输入密码: 666

```

(注: 为测试系统功能, 此处默认登录教工号为 1001)

b) 教师菜单界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2
-----
                *功能菜单*
-----
                1、修改个人信息
                2、添加课程信息
                3、添加成绩信息
                4、退出系统
-----
请输入对应的数字进行操作:

```

c) 教师个人信息修改界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Hc
<terminated>
请输入对应的数字进行操作:1
编号      姓名      学院      系
1001      Jia      UESTC      ST
请按对应格式输入修改后的教师信息!
1001      NEW      UESTC      ST
修改后的教师信息:
编号      姓名      学院      系
1001      NEW      UESTC      ST
操作已保存至文件!
返回主菜单? (y/n):

```

d) 教师新增课程界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Hc
<terminated>
请输入对应的数字进行操作:2
请按对应格式输入新增的课程信息!
888801800      NewNew      6      24
新增的课程信息:
课程号      名称      学分      学时
888801800      NewNew      6      24
操作已保存至文件!
返回主菜单? (y/n):

```

e) 新增课程输出至 Course.txt 文件保存

ID	Course Name	Credits	Score
888801874	Java Prog	2	48
888801876	English8	2	78
888801878	Calculus	6	78
888801877	Calculus8	4	46
888801800	NewNew	5	24

f) 教师录入成绩界面

```

请输入对应的数字进行操作:3
请按对应格式录入成绩信息!
1111004 201818001 Peter NEWNEW Teacher 100
录入的成绩信息:
成绩单号      学号      姓名      课程      教师      成绩
1111004      201818001  Peter    NEWNEW    Teacher    100
操作已保存至文件!
返回主菜单? (y/n):
  
```

g) 录入成绩输出至 Grade.txt 文件保存

ID	Student ID	Student Name	Course	Teacher	Score
1111001	201818001	Peter	JavaProg	Zhou	68
1111001	201818002	Helen	JavaProg	Zhou	88
1111001	201818003	Tommy	JavaProg	Zhou	99
1111001	201818004	Dean	JavaProg	Zhou	100
1111001	201818005	George	JavaProg	Zhou	69
1111001	201818006	Oscar	JavaProg	Zhou	34
1111001	201818007	Taylor	JavaProg	Zhou	78
1111001	201818008	David	JavaProg	Zhou	92
1111002	201818001	Peter	EnglishB	Li	65
1111002	201818002	Helen	EnglishB	Li	76
1111002	201818003	Tommy	EnglishB	Li	86
1111002	201818004	Dean	EnglishB	Li	23
1111002	201818005	George	EnglishB	Li	88
1111002	201818006	Oscar	EnglishB	Li	95
1111002	201818007	Taylor	EnglishB	Li	92
1111002	201818008	David	EnglishB	Li	77
1111003	201818001	Peter	Calculus	Jia	65
1111003	201818002	Helen	Calculus	Jia	76
1111003	201818003	Tommy	Calculus	Jia	86
1111003	201818004	Dean	Calculus	Jia	23
1111003	201818005	George	Calculus	Jia	85
1111003	201818006	Oscar	Calculus	Jia	95
1111003	201818007	Taylor	Calculus	Jia	88
1111003	201818008	David	Calculus	Jia	77
1111004	201818001	Peter	NEWNEW	Teacher	100

(3) 管理员角色

a) 管理员登录界面

```

欢迎进入学生成绩管理系统

请选择登录角色:  1. 学生  2. 教师  3. 教务员
3
请输入管理员账号: 8888
请输入密码: 666
  
```

(注: 为测试系统功能, 此处默认登录管理员账号为 8888)

b) 管理员菜单界面

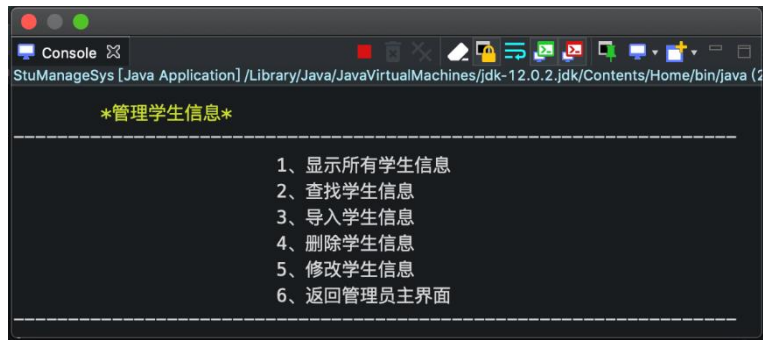
```

*功能菜单*

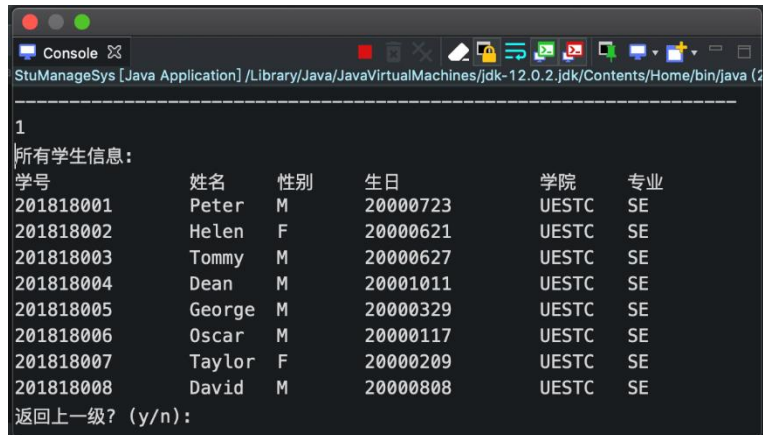
1、管理学生信息
2、管理教师信息
3、管理课程信息
4、打印成绩单
5、退出系统

请输入对应的数字进行操作:1
  
```

c) 管理学生信息菜单



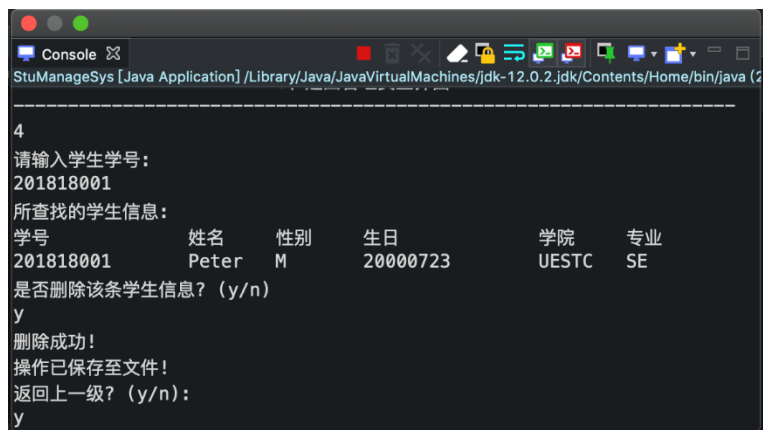
d) 显示所有学生信息界面



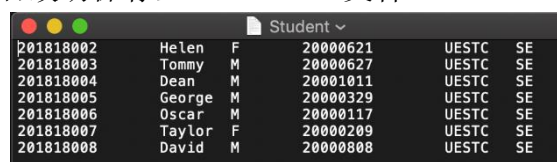
e) 查找学生界面



f) 删除学生界面



g) 删除学生后改动保存至 Student.txt 文件



h) 修改学生信息界面

```

<terminated> StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java
5
请输入学生学号:
201818002
所查找的学生信息:
学号      姓名      性别      生日      学院      专业
201818002  Helen    F        20000621  UESTC    SE
请按对应格式输入修改后的学生信息!
201818002  New      F        20000621  UESTC    SE
修改后的学生信息:
学号      姓名      性别      生日      学院      专业
201818002  New      F        20000621  UESTC    SE
操作已保存至文件!
返回上一级? (y/n):

```

i) 修改后的学生信息输出至 Student.txt 保存

201818002	New	F	20000621	UESTC	SE
201818003	Tommy	M	20000627	UESTC	SE
201818004	Dean	M	20001011	UESTC	SE
201818005	George	M	20000329	UESTC	SE
201818006	Oscar	M	20000117	UESTC	SE
201818007	Taylor	F	20000209	UESTC	SE
201818008	David	M	20000808	UESTC	SE

j) 显示所有教师信息界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2
-----
1
所有教师信息:
编号      姓名      学院      系
1001      Jia       UESTC     IS
1003      Zhou      UESTC     ST
1002      Li        UESTC     ST
返回上一级? (y/n):
y

```

k) 查找教师界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2
所查找的教师信息:
编号      姓名      学院      系
1003      Zhou      UESTC     ST
返回上一级? (y/n):
y

```

l) 删除教师界面

```

StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2
4
请输入教师编号:
1001
所查找的教师信息:
编号      姓名      学院      系
1001      Jia       UESTC     IS
是否删除该条教师信息? (y/n)
y
删除成功!
操作已保存至文件!
返回上一级? (y/n):
y

```

(注: 删除教师后改动保存至 Teacher.txt 截图略)

m) 修改教师界面

```

Console
<terminated> StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home
5
请输入教师编号:
1002
所查找的教师信息:
编号 姓名 学院 系
1002 Li UESTC ST
请按对应格式输入修改后的教师信息!
1002 New UESTC ST
修改后的教师信息:
编号 姓名 学院 系
1002 New UESTC ST
操作已保存至文件!
返回上一级? (y/n):

```

(注: 修改后的教师信息输出至 Teacher.txt 保存截图略)

n) 成绩排名、计算平均分、打印成绩单界面

```

Console
StuManageSys [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Contents/Home/bin/java (2
请输入对应的数字进行操作:4

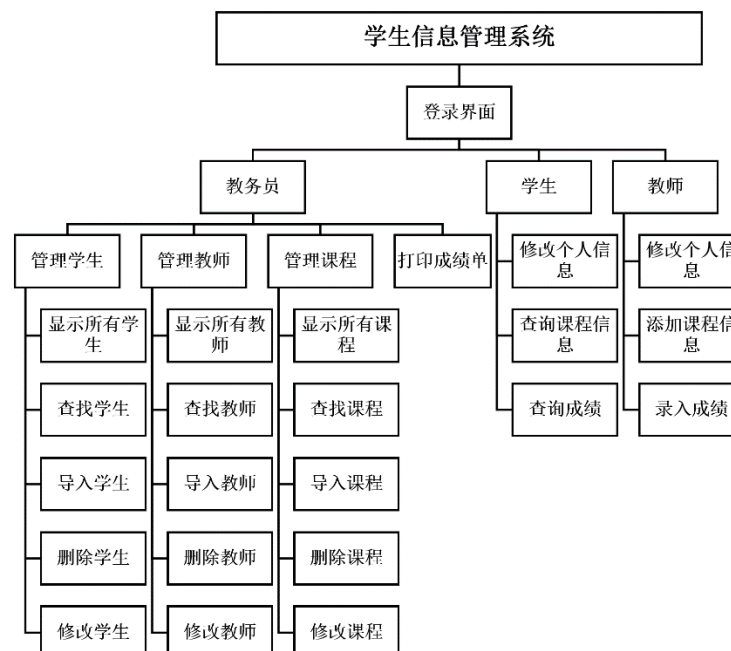
*打印成绩单*

请输入待打印的成绩单编号:
1111001
编号为1111001的成绩单信息:
成绩单号 学号 姓名 课程 教师 成绩
1111001 201818004 Dean JavaProg Zhou 100
1111001 201818003 Tommy JavaProg Zhou 99
1111001 201818008 David JavaProg Zhou 92
1111001 201818002 Helen JavaProg Zhou 88
1111001 201818007 Taylor JavaProg Zhou 78
1111001 201818005 George JavaProg Zhou 69
1111001 201818001 Peter JavaProg Zhou 68
1111001 201818006 Oscar JavaProg Zhou 34
平均分: 78.0
返回主菜单? (y/n):

```

o) 管理课程信息界面与管理学生、教师界面类似, 此处省略

2、系统模块关系图



3、对象及属性的创建

(1) Student 类

```
1. package StuManageSys;
2.
3. public class Student implements Comparable<Student> {
4.     private int sNo;
5.     private String sName;
6.     private String sSex;
7.     private String sBirth; //YYYY.MM.DD
8.     private String sSchool;
9.     private String sMajor;
10. }
```

说明：Student 类定义了由关键字 private 修饰的成员变量：学生学号 sNo、学生姓名 sName、学生性别 sSex、学生生日 sBirth、学生所属学院 sSchool 以及学生专业 sMajor，这些变量只能由 Student 类私有访问。

(2) Teacher 类

```
1. package StuManageSys;
2.
3. public class Teacher implements Comparable<Teacher>{
4.     private int tNo;
5.     private String tName;
6.     private String tSchool;
7.     private String tDepartment;
8. }
```

说明：Teacher 类定义了由关键字 private 修饰的成员变量：教职工号 tNo、教师姓名 tName、教师所属学院 tSchool 以及教师所在系别 tDepartment，这些变量只能由 Teacher 类私有访问。

(3) Course 类

```
1. package StuManageSys;
2.
3. public class Course implements Comparable<Course>{
4.     private int cNo;
5.     private String cName;
6.     private int cCredit;
7.     private int cHour;
8. }
```

说明：Course 类定义了由关键字 private 修饰的成员变量：课程号 cNo、课程名称 cName、学分 cCredit 以及课时 cHour，这些变量只能由 Course 类私有访问。

(4) Grade 类

```
1. package StuManageSys;
2.
3. public class Grade implements Comparable<Grade>{
```

```

4.     private int gNo;
5.     private int sNo;
6.     private String cName;
7.     private String tName;
8.     private String sName;
9.     private int gPoint;
10.}

```

说明：Grade 类定义了由关键字 `private` 修饰的成员变量：成绩单号 `gNo`、学生学号 `sNo`、课程名称 `cName`、教师姓名 `tName`、学生姓名 `sName` 以及成绩 `gPoint`，这些变量只能由 Grade 类私有访问。

4、对象信息获取、行为方法创建（举例说明，完整代码见附件）

(1) 对象信息获取

```

1. public Student() {
2. }
3.     public Student(int sNo, String sName, String sSex, String sBirth, String sSchool, String sMajor) {
4.         super();
5.         this.sNo=sNo;
6.         this.sName=sName;
7.         this.sSex=sSex;
8.         this.sBirth=sBirth;
9.         this.sSchool=sSchool;
10.        this.sMajor=sMajor;
11.    }
12.    public void setNo(int no) {
13.        this.sNo=no;
14.    }
15.    public int getNo() {
16.        return sNo;
17.    }
18.    public void setName(String name) {
19.        this.sName=name;
20.    }
21.    public String getName() {
22.        return sName;
23.    }
24.    public void setSex(String sex) {
25.        this.sSex=sex;
26.    }
27.    public String getSex() {
28.        return sSex;
29.    }
30.    public void setBirth(String birth) {
31.        this.sBirth=birth;
32.    }
33.    public String getBirth() {
34.        return sBirth;
35.    }
36.    public void setSchool(String school) {
37.        this.sSchool=school;
38.    }
39.    public String getSchool() {
40.        return sSchool;

```

```

41.     }
42.     public void setMajor(String major) {
43.         this.sMajor=major;
44.     }
45.     public String getMajor() {
46.         return sMajor;
47.     }
48.     public String toString() {
49.         re-
         turn sNo+"\t"+sName+"\t"+sSex+"\t"+sBirth+"\t"+sScho
         ol+"\t"+sMajor;
50.     }
51.     public int compareTo(Student stu) { //按学号排
        序
52.         return this.sNo-stu.getNo();
53.     }

```

说明：由 Student 类的对象信息获取为例：定义了构造器、以及对学号、姓名、性别、生日、学院、专业这几个变量的初始化方法和获取信息方法；重写了 toString 方法，便于在 print 时美观的按照格式输出各字段。

另 Student 类使用了 implements 关键字来实现一个名为 Comparable 类中的接口，我们在字符串比较中见到过 CompareTo 方法，Java 中所有的 compareTo 方法都源于一个共同的接口，那就是 Comparable，这个接口只有一个方法，那就是 CompareTo。所有要实现比较功能的类，都建议实现这个接口，而非是自己定义这个功能，这体现了 Java 面向对象的概念（将具有相同功能的事物抽象到一个共同的类或接口），并且为了多态也建议通过实现接口来进行向上转型，通过接口来操作具体实现。实现 compareTo 方法能按 sNo 升序排列。

(2) 对象行为方法创建

```

1. public static void stuMenu() {
2.     System.err.println("\n\t*功能菜单*");
3.     System.out.println("-----");
4.     System.out.println("1、修改个人信息");
5.     System.out.println("2、查询课程信息");
6.     System.out.println("3、查询成绩");
7.     System.out.println("4、退出系统");
8.     System.out.println("-----");
9.     System.out.print("请输入对应的数字进行操作:");
10. }

```

说明：主程序调用 stuMenu 方法，在屏幕上打印出可选择的功能菜单，根据用户输入的功能选择数字不同，采用 switch-case 语句跳转到相应的功能模块。

```

1. public static void sInfo() throws IOException {
2.     int stu=0;
3.     for(int j=0;j<StuList.size();j++) {
4.         if(Stu-
List.get(j).getNo()==201818001) {
5.             System.out.println("学号\t姓名\t性别\t生日\t学院
\t专业");
6.             System.out.println(StuList.get(j).toString());
7.             stu=j;
8.             } break;
9.         }
10. System.out.println("请按对应格式输入修改后的学生信
息! ");
11. Scanner line=new Scanner(System.in);
12. String update=line.nextLine();
13. String split[]=update.split("\t");
14. int no[]=new int[100];
15. no[0]=Integer.valueOf(split[0]).intValue();
16. line.close();
17. StuList.get(stu).setNo(no[0]);
18. StuList.get(stu).setName(split[1]);
19. StuList.get(stu).setSex(split[2]);
20. StuList.get(stu).setBirth(split[3]);
21. StuList.get(stu).setSchool(split[4]);
22. StuList.get(stu).setMajor(split[5]);
23. System.out.println("修改后的学生信息: ");
24. System.out.println("学号\t\t姓名\t性别\t生日
\t\t学院\t专业");
25. System.out.println(StuList.get(stu).toString());
26. if(outputTxt(1)==1)
27.     System.out.println("操作已保存至文件!");
28. System.out.println("返回主菜单? (y/n): ");
29. char yn = sc.next().charAt(0);
30. if (yn == 'y')
31.     sStatus = 1;
32. }

```

说明：此方法用于学生修改自己的个人信息。在定义方法时注意添加 throws IOException 的声明，说明此方法可能抛出读写异常。由于在一开始时定义了以下语句：

```

1. static List<Student> StuList = new LinkedList<Stu-
dent>();
2. readTxt(1);

```

其中 readTxt 方法将 Student.txt 文件中的信息读入 StuList 链表中（见后文），于是使用 for 语句在 StuList 链表中查找待修改的学生结点；查找成功后提示用户输入修改后的学生信息，用一个 Scanner 变量读入用户输入的一行字符串，调用 Scanner 的 split 方法，以一个制表符\t（也可以用其他分隔符）将字符串划分为几个字段，其中学号为 int 型变量，在 Student 类中的 setNo 方法参数设置的同样为 int 型变量，因此将第一个字段调用 Interger.valueOf 方法从字符串转换为 int 型变量，后调用 Student 类的 set 型方法

重置该学生信息，并调用 `outputTxt` 方法将链表保存至 `Student.txt` 文件，实现信息更新效果。

```
1. public static int manaLogin() {
2.     System.err.print("请输入管理员账号: ");
3.     int id = sc.nextInt();
4.     System.err.print("请输入密码: ");
5.     int pwd = sc.nextInt();
6.     if(id == 8888 && pwd == 666)
7.         return 1;
8.     else
9.         return 0;
10. }
```

说明：此方法为角色登录控制方法。在本实验中，简单用了明文比对的方式来进行。但 Java 自带了有关账号、密码输入登录的类。如果是控制台程序，可以调用 `Console` 类的 `readPassword` 方法提示用户输入密码并自动隐藏为“*”，如下：

```
1. Console console = System.console();
2.     if (console == null) {
3.         System.out.println("Error");
4.         System.exit(0);
5.     }
6.     char passwordArray[] = console.readPass-
7. word("Enter your secret password: ");
8.     console.printf("Password en-
9. tered was: %s\n", new String(passwordArray));
```

但这个方法在使用 Eclipse 运行时抛出 `NullPointerException` 的异常，查阅资料后发现是由于 Eclipse 的进程占用控制台导致的；如果是窗体程序，可以编写一个 `Frame` 类继承 `swing` 类，调用 `textfield` 的 `getText` 方法即可实现在输入框中的密码自动隐藏为“*”。

```
1. public static void teaView() {
2.     if (TechList.isEmpty() == true) {
3.         System.out.println("无教师信息!");
4.     } else {
5.         System.out.println("所有教师信息: ");
6.         System.out.println("编号\t姓名\t学院\t系");
7.         for (int j = 0; j < TechList.size(); j++) {
8.             Collections.sort(TechList);
9.             System.out.println(TechList.get(j).toString());
10.        }
11.    }
12. }
```

说明：与修改学生个人信息类似，使用 `for` 循环遍历 `TechList`，在遍历的同时，调用 `Collections` 类中的 `sort` 方法，根据在 `Teacher` 类中重写的 `compareTo` 方法对教工号进行升序排列，后调用 `toString` 方法，将 `Teacher` 类的各字段格式化 `print`。

```

1. public static int teaFind() {
2.     int j=0;
3.     if (TechList.isEmpty() == true) {
4.         System.out.println("无信息! ");
5.         return -1;
6.     } else {
7.         System.out.println("请输入教师编号: ");
8.         int no=sc.nextInt();
9.         for (j = 0; j < Tech-
List.size(); j++) {
10.            if(TechList.get(j).getNo()==no)
11.                break;
12.        }
13.        if(j==TechList.size()) {
14.            System.out.println("无待查教师信息! ");
15.            return -2;
16.        } else {
17.            System.out.println("所查找的教师信息: ");
18.            System.out.println("编号\t姓名\t学院\t系");
19.            System.out.println(TechList.get(j).toString());
20.            return j;
21.        }
22.    }
23. }

```

说明：该方法的返回值为 int，当 TechList 链表为空时返回-1，查询失败时返回-2，查询成功时返回该结点编号。调用该方法的函数根据返回值的不同，进行对应的相应。

```

1. public static void manaFuncGrade( ) {
2.     System.err.println("\n\t*打印成绩单*");
3.     System.out.println("-----");
4.     System.out.println("请输入待打印的成绩单编号: ");
5.     int j=0;
6.     int no=sc.nextInt();
7.     for (j = 0; j < GradeList.size(); j++) {
8.         if(GradeList.get(j).getNo()==no)
9.             break;
10.    }
11.    if(j==GradeList.size()) {
12.        System.out.println("无待打印成绩单编号! ");
13.    } else {
14.        int sum = 0, i=0, count=0;
15.        System.out.println("编号为
"+GradeList.get(j).getNo()+"的成绩单信息: ");
16.        System.out.println("成绩单号\t\t学号\t\t姓名\t课程
\t\t教师\t成绩");
17.        for(i = 0; i < GradeList.size(); i++) {
18.            Collections.sort(GradeList);
19.            if(GradeList.get(i).getNo()==no) {
20.                System.out.println(GradeList.get(i).toString());
21.                sum+=GradeList.get(i).getPoint();
22.                count++;
23.            }
24.        }
25.        System.out.println("平均分: "+(float)(sum/count));
26.    }
27.    System.out.println("返回主菜单? (y/n): ");
28.    char yn = sc.next().charAt(0);

```

```

29.         if (yn == 'y')
30.             mStatus = 1;
31.     }

```

说明:该方法使用 for 循环查找 GradeList 中某一科目的成绩信息,并能调用 Collections 类中的 sort 方法,根据在 Grade 类中重写的 compareTo 方法对成绩进行降序排列,同时计算平均分。必要时,可以将结果保存至文件后打印。

(3) 文件读写

```

1. public static int readTxt(int i) throws FileNot-
   FoundException { //0 学生, 1 教师, 2 课程, 3 成绩
2.     switch(i) {
3.         case 1: {
4.             Scanner file=new Scan-
               ner(new File("src/stuManageSys/Student.txt"));
5.             while(file.hasNextLine()) {
6.                 String temp=file.nextLine();
7.                 String split[]=temp.split("\t");
8.                 int no[]=new int[100];
9.                 no[0]=Integer.valueOf(split[0]).intValue();
10.                StuList.add(new Stu-
                   dent(no[0],split[1], split[2], split[3], split[4], s
                     plit[5]));
11.            }
12.            file.close();
13.            break;
14.        }
15.        case 2: {...//省略}
16.        case 3: {...//省略}
17.        case 4: {...//省略}
18.    }
19.    return 1;
20. }

```

说明:由于本实验要读入的文件信息为格式化的纯文本文件,因此选用 Scanner 类来实现。在定义方法时注意添加 throws IOException 的声明,说明此方法可能抛出读写异常。用一个名为 temp 的 String 暂存从文件读入的一行字符串信息,调用 Scanner 的 split 方法,以一个制表符\t(也可以用其他分隔符)将字符串划分为几个字段,其中学号为 int 型变量,且 Student 类中的构造器对于学号的参数类型同样也是 int,因此需要将第一个字段调用 Integer.valueOf 方法将其转换为 int 型参数,后调用构造器,并添加至 StuList 中。

```

1. public static int outputTxt(int i) throws IOExcep-
   tion {
2.     switch(i) {
3.         case 1: {
4.             File ofile=new File("src/stuManag-
               eSys/Student.txt");
5.             if (!ofile.isFile()) {

```



```

6.         ofile.createNewFile();
7.     }
8.     BufferedWriter writer = new Buff-
        eredWriter(new FileWriter("src/stuManageSys/Stu-
        dent.txt"));
9.     for (Student stu:StuList){
10.         writer.write(stu.toString() + "\n");
11.     }
12.     writer.close();
13.     break;
14. }
15. case 2: {...//省略}
16. case 3: {...//省略}
17. case 4: {...//省略}
18. }
19. return 1;
20. }

```

说明：本方法将链表中的数据变化输出至相应文件保存，在定义方法时注意添加 throws IOException 的声明，说明此方法可能抛出读写异常。写文件选择 BufferedWriter 字符缓冲输出流构造函数，将每个对应 List 中的结点以 toString 的方法格式化地输出至 txt 文件中，在 txt 文件不存在时，直接创建新空白文件。

九、 总结及心得体会：

本实验在于考察对 Java 面向对象特性的掌握及应用，锻炼学生对对象模型设计及软件模型的理解和分析能力。通过本次实验，我掌握了 Java 程序编写的一般思路与方法，锻炼了小型软件、系统的理解、分析、编码能力，熟悉了 Eclipse 软件各项功能的使用，提升了自我的实验能力。

在大一学年，《数据结构》课程曾要求使用 C 语言编写一个学生成绩管理系统，要求实现的功能类似于本实验教务员的功能，我查看了其代码量为 700 行左右，而本实验在完成三个登录角色的相关功能要求下，代码量为 1000 行左右。从这一点可以看出 Java 面向对象这一特性和编程思想对程序员工作量的减少。

但由于第一次使用 Java 完整的编写这样的系统，对面向对象这一特性还没有得以足够深入的了解与内化，导致很多地方还是出现了不必要的代码冗余，没有很好地利用继承、重写这些技术方法。另对 Java GUI 编程了解还不是很多，加之时间比较紧，很遗憾没有把这个系统实现 GUI 界面。希望这些不足与遗憾能在以后的学习中加以弥补。

十、 对本实验过程及方法、手段的改进建议：

实验指导书中对某些细节的描述还不是很准确，导致在设计系统时需求分析阶段比较模糊。另本实验给出的实验学时相对较少，希望能增加必要的实验学时，使得实验完成的更加完善。

报告评分：

指导教师签字：