

**软件学院课程设计任务书**

题 目： 学生信息管理系统

专 业： 软件工程

班 级： 软件工程1807班

姓 名： 翟翊君

学 号： 8002118184

完成人数： 1人

起讫日期： 2019. 6. 3 — 2019. 6. 23

任课教师： 华 鑫 职称： 讲师

部分管主任： 夏灵林

完成时间： 2019年6月23日前

|  |
| --- |
| **大作业的要求和内容：（包括题目选择范围、技术要求、递交时间、考核方法等）**  一、实验目的  通过设计性、综合性的C++语言程序设计，提高学生编写程序、调试程序的能力，加深C++语言知识的理解和把握，初步掌握面向对象和泛型编程的基本思想以及使用第三方类库进行软件构建能力。  二、实验内容和要求  要求学生从老师给定的选题或者自定选题（要得到老师的允许），1人1组，根据需求描述（或自己撰写需求描述），独自完成需求分析，程序模块设计以及程序的编写、调试和测试。  要求学生根据需求描述独自完成需求分析、程序模块设计以及程序的编写、调试和测试。包括应用程序体系结构的确定、类和对象的设计与实现、模板容器以及算法的选用、文件操作、控制台或GUI的输入输出等。必须满足如下要求：   1. 采用**面向对象程序设计，**要运用类的封装、继承、多态等机制； 2. 具备**文件读写**功能，程序中的数据要以文件方式保存； 3. 程序中要具备对**STL容器**的运用； 4. 程序中要求具备**异常处理**； 5. 必须体现**多文件结构**的思想，类定义和类实现进行分离； 6. 程序要求做到界面友好，对于用户的输入操作要有明确的**提示信息**，并对输入的数据进行**合法性和正确性**的检查； 7. 做到工程名、类名、变量名、代码格式等合乎**规范**，代码中必须对所设计的类的功能或函数功能进行规范的**注解**。 8. 要求具备**菜单功能**，根据用户的输入，执行相应的操作。在**菜单选项**中，必须包含“关于”一项，该菜单项用来显示课程设计的基本信息，包括程序名称、版本、完成时间、姓名、班级、学号等信息；   三、递交时间  在设计完成之后书写设计说明书（实验报告），按规定报告的格式书写，在2019年6月23日前与设计相关文档（包括源代码与可执行程序）一并上交。  四、答辩方式  通过平时成绩汇总，再由源代码实现程度及课程设计完成度进行综合评价，综合成绩排名前10%的同学必须答辩，学习情况异常和排名靠后的同学由教师提供名单进行答辩，其他同学自愿答辩。  五、课程设计格式及要求  要求说明：  1. 完成需求陈述中所提到的所有功能；  2. 要求撰写需求分析、软件设计文档；  3. 设计体会和课程建议；  4. 设计报告严禁抄袭，否则按不及格论。  最终提交的作业，除提交完整的程序工程代码外，还必须以大作业报告的形式阐述整个程序的实现过程，要求报告内容中必须包括：   1. **技术路线**（要详细说明所使用的主要技术和实现原理）3分； 2. **需求分析**（项目介绍、功能需求）5分； 3. **系统分析与设计**（阐述项目中需解决的关键技术问题，同时要以类关系图、程序流程图、功能模块示意图等辅助系统设计的描述）； 4. **程序设计与实现**（首先要描述代码工程的文件组织结构，然后以文字的方式阐述代码中每个类的设计意图、主要功能、运用到的关键知识点，主要成员函数的设计意图及功能介绍，对于代码实现中有特色的算法或有创新的地方，可以着重描述（这将视为课程设计的加分点）。如果必须给出实现代码才能更好地说明问题时，也必须先有相关的文字叙述，然后才是代码，代码只是作为例证。）10分； 5. **系统测试**（该部分不能只简单展示几幅程序运行时的截图就完事！要设计几组有代表性的数据来验证程序的各个功能是否达到设计需求。也就是测试是否有不正确或者是遗漏了的功能，在接口上，输入能否正确的接受，能否输出正确的结果；是否有数据结构错误或者外部信息访问错误；性能上是否满足要求；是否有初始化或终止性错误等。建议以表格的形式展示）5分； 6. **个人小结**（课程设计心得体会不少于800字, 该部分必须包含课程设计项目的工作量、开发过程中遇到的困难以及如果克服困难、个人收获、得到的启示或教训等等，切忌空洞无实际内容或千篇一律的敷衍文字）5分； 7. **参考文献**(该部分给出整个程序从选题、需求分析、系统设计到代码实现过程中所参考的书籍、网上资料等）2分。   格式说明：   1. 目录请采用Word自动生成的方式生成，目录格式采用“正式”，“目录”两字小三号宋体加粗，目录内容小四号宋体，页码数字对齐。 2. 页面设置：上2.54cm，下2.54cm，左3.67cm，右2.67cm，页眉1.5cm,页脚1.75cm，行间距1.35倍。 3. 页眉和页码：页眉为相应内容的标题，页码中目录用罗马数字(I，II，III……)编排，从正文第一章开始按照阿拉伯数字（1，2，3……）编排。 4. 全文中文字体采用宋体，英文字体采用Times New Roman字体，一级标题使用小三号字体，二级、三级标题均使用四号字体，正文内容采用小四号字体 5. 图表：图表内容五号宋体。 6. 参考文献：“参考文献”四字四号宋体，参考文献内容小四号宋体，其中英文用小四号Times New Roman 体。 |
| 教师小结：  成绩：  教师 签名：  系部负责人：  学生 签名： |



**NANCHANG UNIVERSITY**

**C++程序设计课程设计报告**



**题 目:** 学生信息管理系统

**学 院：** 软件学院

**专 业：** 软件工程

**班 级：** 软件工程1807班

**完成人数：** 1人

**人 员：**

**起讫日期：** 2019. 6. 3 — 2019. 6. 23

**任课教师：** 华 鑫 职称： 讲 师

**完成时间：** 2019年6月23日前

**填表日期：** 2019 年 6 月 20 日

**目 录**

一、 项目技术路线说明 1

二、 项目需求分析 2

2.1 项目介绍 2

2.2 功能需求 2

三、 系统分析与设计 3

3.1 本程序需解决的关键技术问题 3

3.2 程序流程 4

3.3 功能模块 4

3.4 类的设计 5

四、 程序设计与实现 9

4.1 工程文件组织结构 9

4.2 程序设计 9

4.3 程序实现 10

五、 系统测试 14

5.1 测试概要 14

5.2 测试环境与配置 14

5.3 测试方法和工具 15

5.4 系统功能分解 15

5.5 测试内容 15

六、 个人小结 32

七、 参考文献 34

# 项目技术路线说明

学生信息管理系统所使用的编程语言是C++语言。C++语言具有面向对象的特点，给编写程序带来了极大地方便。学生信息管理系统程序设计通过抽象、封装、继承和多态使程序代码达到了很大限度的可重用和可扩展。而程序中的多种多样的类是此次程序设计的重要组成部分，利用这些类可以实现数据的封装，隐蔽。通过这些类的派生与继承，能够实现对问题的抽象描述。除此之外，此次程序设计所运用到的泛型程序设计思想、STL标准模板库、I/O流类库、文件操作以及异常处理是本程序设计的核心。

在程序设计中，数据的共享与保护显得尤为重要。设置常成员、常对象、常引用和常指针等对数据进行保护，增强程序代码的安全性可靠性。

在程序的组织上，采用多文件结构。将类的声明和定义在不同文件中实现，可以对不同文件进行单独编写、编译，最后再连接。同时利用类的封装性，在调试程序、修改代码时，只对某一个类的定义和实现进行修改，可以提高程序编写效率。

在程序中适当使用多态机制。声明抽象类，为整个类族提供了统一的对外接口。使用指向派生类的指针就可以对派生类的对象进行统一操作。

在程序中巧用泛型程序设计及STL标准模板库，可以提高程序编写效率。泛型程序设计是一种重标要的程序设计方法，在程序中合理利用泛型程序设计，提高了代码可重用性。STL准模板库是C++标准库的一部分，协调使用其中的容器，迭代器，函数对象以及算法，会使学生信息管理系统程序的编写更容易。

学生信息的管理实质就是数据的管理，数据的管理离不开文件操作和输入输出。因此，在程序中合理使用文件操作和输入输出，能给数据处理带来极大得方便。

数据处理会出现许多极端情况，在程序中增加异常处理，能提高程序的容错能力。

学生信息管理系统算法的实现主要依赖于对泛型程序设计思想、STL标准模板库、I/O流类库、文件操作以及异常处理的充分理解和恰当使用。利用STL标准模板库及文件操作设计了一个简单的数据库，极大得方便了数据的的存储和访问，也方便了学生信息的格式化输出。



# 项目需求分析

## **项目介绍**

学生信息管理系统是针对学校人事处的大量业务处理工作而开发的管理软件，主要用于学校信息管理，总体任务是实现学生信息关系的系统化、科学化、规范化和自动化，其主要任务是用计算机对学生各种信息进行日常管理，如信息的录入、浏览、排序、查询、统计、修改、删除，针对这些要求设计了学生信息管理系统。

## **功能需求**

2.2.1 管理员对系统的管理功能

管理员可以通过账号登入系统，系统主要对学生信息进行收集整理和存储。管理员在学生基本信息上可以进行添加、修改和删除数据。除此之外，管理员可以对系统进行升级和维护。

2.2.2 学生和教师对系统的查看、查询和信息录入等功能

(1) 学生登入系统后可以查看自己的相关信息：

学生个人信息：姓名、学号、性别、年龄、出生年月、籍贯、家庭住址、电话号码、邮箱、班级和政治面貌。

1. 教师登入系统后可以查看和操作学生的相关信息

可以查看：姓名、学号、性别、年龄、出生年月、籍贯、家庭住址、电话号

码、邮箱、班级和政治面貌等学生基本信息。

可以录入、浏览和查询学生信息，也可以对学生信息进行排序、统计、删除和修改等。

# 系统分析与设计

## **本程序需解决的关键技术问题**

* + 1. 个人账号注册

学生信息管理系统应具有账户注册功能，账户由账号和密码组成，其中需要解决的关键技术问题有：

1. 如何存储账号和密码，使得二次登陆时，可以通过验证账号和密码登入系统。
2. 当注册账户和登陆账户时，如何实现隐藏输入密码，保证信息安全。
3. 如何将固定账户与固定学生信息进行绑定，使得每次使用账户登入系统，学生信息都存在且可以直接进行操作。
4. 一个账户应具有修改密码功能，如何实现密码修改，使得与账号绑定的学生信息不变。
   * 1. 学生信息管理

学生信息管理系统主要实现对学生信息的存储和整理，其中需要解决的技术问题有：

1. 学生信息管理系统可以对学生信息进行录入，格式化录入学生信息以及存储学生信息，便是主要问题。其中存储学生信息本质上是对数据的存储，如何设计数据结构，使得内存可以动态分配，数据动态存储。
2. 学生信息管理系统可以对学生信息进行不同程度的排序，按不同排序标准对学生信息排序，是当下面临的问题。
3. 学生信息管理系统可以对学生信息进行广度和深度查询，实现按不同查询标准对学生信息进行精准查询，是写程序所面临的问题之一。
4. 学生信息管理系统提供了分类统计学生信息功能，如何运用分类统计的思想，实现对学生信息的分类统计。
5. 学生信息管理系统还可以对学生信息进行修改和删除，如何实现对学生信息进行修改和删除。
6. 如何在基于以上操作的情况下，当再次进入固定账户时，学生信息依然可以同步。

## **程序流程**

程序流程如图流程图 图3-2所示

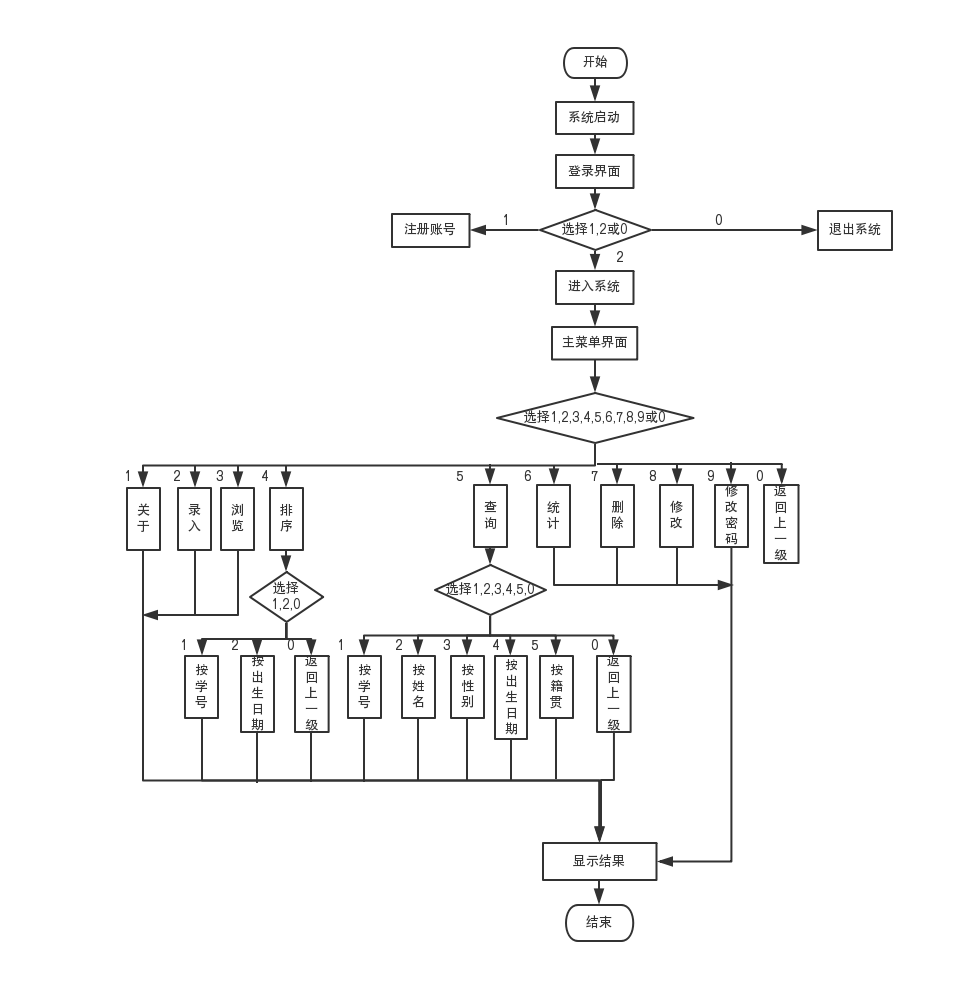


图3-2

## **功能模块**

学生信息管理系统功能模块分为：

1. 账户注册：用户在学生信息管理系统上注册一个属于自己的账号，用此账号管理学生信息。
2. 信息录入：允许用户录入学生信息，并对学生信息分类保存到磁盘文件中。当用户再次登入系统时，磁盘文件信息会自动被读取到内存中，起到同步作用。
3. 信息浏览：可对学生信息进行浏览。
4. 信息排序：允许用户按学号、出生日期对学生进行排序。
5. 查询信息：用户可以按学号、姓名、性别、出生日期和籍贯进行信息查询。
6. 统计信息：允许用户按性别、出生日期、籍贯、班级和政治面貌进行统计学生情况，并计算出所占比例，输出统计表。
7. 删除信息：允许用户输入学号，删除学号对应的所有信息。
8. 修改信息：用户可以按学号，修改学号对应的学生信息。
9. 修改密码：当用户输入正确的账号和密码时，才能修改本账号的密码。修改密码会有确定密码进行验证，防止用户手误输错。

## **类的设计**

依据2.2功能需求分析以及3.1需要解决的关键技术问题进行类的设计。

3.4.1 账户类设计

用户可以在学生信息管理系统上注册一个属于自己的账号，用此账号管理学生信息。

首先，可以设计一个账户类Account来表示账户，类内含账号和密码两个主要数据成员。通常，我们在注册账号时，设置密码会有两个环节，一个是设置密码，另一个是输入确认密码，确认密码有效防止了用户手误输错。因此，可以用一个数组code[2]存储密码，数组元素code[1]表示确认密码。

为了保证用户账号密码安全，可以设计一个函数hide Password来隐藏输入密码，用\*号代替密码，同时支持退格重新输入。

当用户在进入系统和修改密码时，需要验证账号信息，才能进行相关操作，这样可以有效保证账户信息安全。为此设计函数verify Information来验证账号信息。函数需要返回一个布尔值，返回值为真表示用户之前在系统上注册过此账号，返回值为假表示用户当前注册了次账号同时进行登录或账号不存在。函数返回值为后续操作反馈一个信息，这个信息用来通知程序，将与账号相关的学生信息从磁盘文件中读入到内存中来。

当用户对所设置的账号和密码有疑问时，可以通过重新设置账号和密码解决疑问。为此，设计一个函数Change Password来修改账号和密码。

以上对账号和密码的存储，是基于对文件的操作来实现的。设计函数save\_account将内存中数据保存到文件中，函数input\_account将数据从文件读入到内存中来。无论是保存数据到文件，还是从文件读数据到内存中，都涉及到查找文件，设计一个查找函数find\_filename来实现文件查找功能。

3.4.2 学生类设计

对学生群体进行属性和行为抽象可以得到Student类，Student类对姓名、性别和籍贯等数据进行封装形成新的数据类型。 类内包含姓名、学号、性别、政治面貌、籍贯、家庭住址、电话号码、电子邮箱、出生日期、年龄以及班级11个数据成员。类内有获取学生相关信息和设置学生相关信息的函数成员。同时，重载流插入运算符和流提取运算符为Student类的友元函数。这样，就可以对整个类对象进行输入输出操作，为后续程序输出提供便捷，增加程序可读性。

3.4.3 管理类设计

对学生信息的基本操作由管理类Manage来表示。类内数据成员包括多重映射，向量，以及两个一维数组。

一般情况下，对于学生信息的查询、删除以及修改等操作都是基于学号，学号和学生信息一一对应，可以用多重映射来表示这种关系。多重映射的键类型为string，值的类型为Student。这样，就可以存储多个学生的学生信息，根据学号来处理学生信息。多重映射可以实现动态内存分配，这样可以实现运行时多个学生的信息的存储。向量容器可以充当一维数组使用，不同的向量元素都用来存储不同的统计数据种类。向量的内置类型为多重映射，多重映射的键的类型为string，值为int；键用来表示统计学生信息时的标准（例如：按籍贯统计，按出生日期统计等），值表示这一标准下的学生数量。向量和多重映射的配合使用，极大方便了学生信息的统计。

设计record函数录入学生信息；函数browse可浏览学生信息，但是，我们不仅在浏览学生信息时需要用到browse，在输出学生信息统计结果时也要用到browse,两函数处理数据的算法相同，处理的数据类型不尽相同，可以使用函数模板browse来解决这一问题，不仅提高了代码可读性，而且提高了代码的可重用性。

对于学生信息排序，可以用函数sorts来实现，函数核心使用了标准模板库的排序函数sort,问题是sort函数仅支持vector、string以及deque的排序，而由红黑树实现的set、map、multimap等容器[2]，元素本身有序（即映射按键值递增），故不支持sort排序, 而我们采用multimap来存储学生信息，想要按学号降序排序，按出生日期排序便很难实现。由于sort支持对vector的排序，所以可以使用vector和pair的混合使用来解决这一问题。pair作为vector的内置数据类型，它的第一个数据的类型为string，第二个数据类型为Student。不难发现，pair的数据类型和multimap的键值数据类型相同。在排序时，可以将multimap中的数据转存到pair中，然后使用排序函数sort对vector排序。

当标准模板库中的sort函数的内置类型为自定义类型或容器时，其函数对象便要由使用者提供。为此我们需要设计两个函数对象CmpByValueDate和CmpByKeyNumber来决定sort递增或递减[2]。

设计函数find实现查询功能；函数deletes实现信息删除；函数change实现修改信息的功能；函数save可以保存信息到磁盘文件；函数input可以从文件读取信息到内存中。

本类还设计了辅助函数chart输出统计表；函数edit编辑学生信息；函数sort\_data按出生日期排序；函数sort\_number按学号进行排序。

3.4.4 菜单类、时间类和控制台类设计

对于菜单的设计，用Menu类表示。Time类表示启动界面和显示当前时间。Console类表示对控制台的颜色、尺寸、光标等控制。

图3-4通过UML图形表示了设计的类及其相互关系。

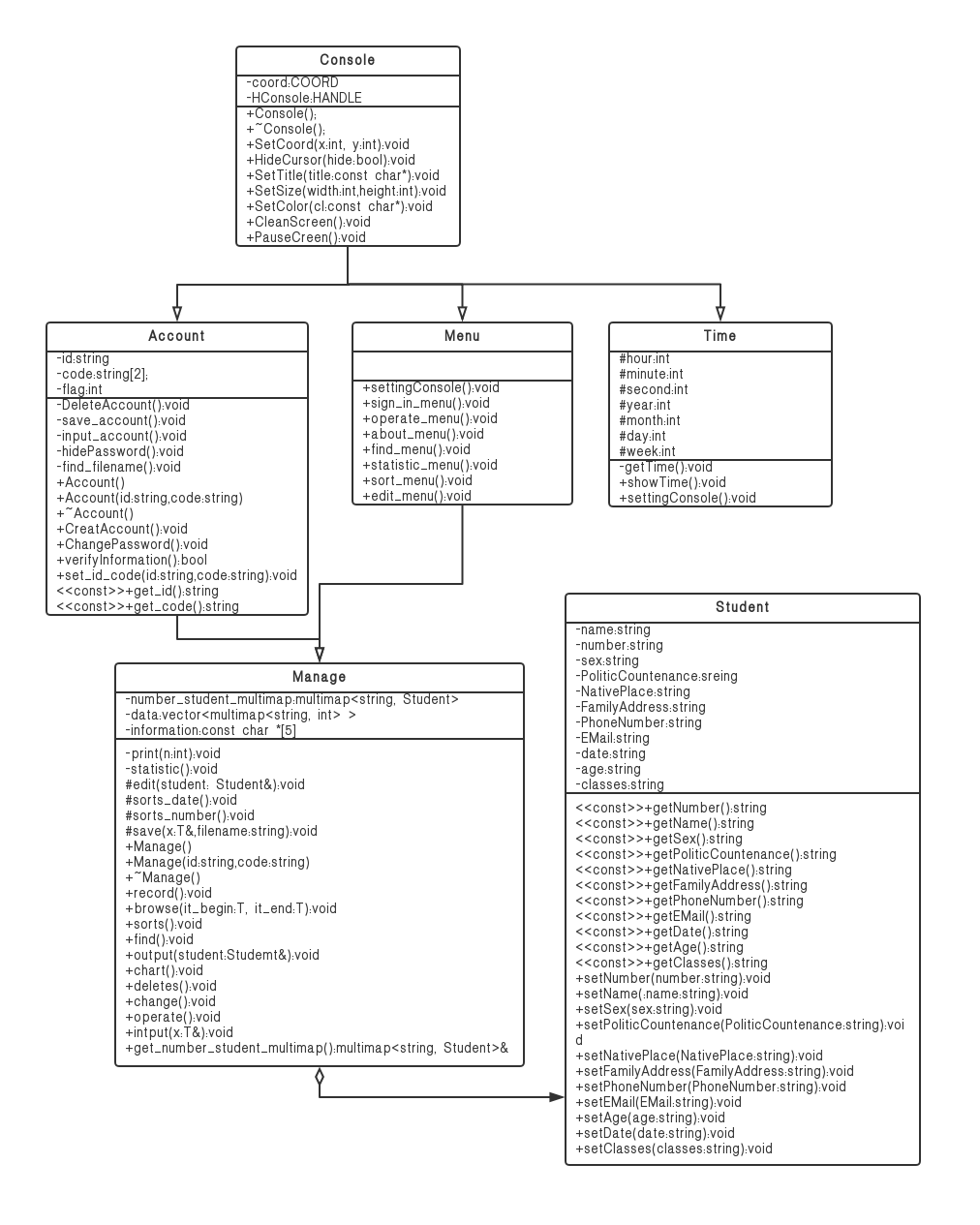


图 3-4

# 程序设计与实现

## **工程文件组织结构**

Student information Management System工程文件由SAccount.h、SConsole.h、SManage.h、SMenu.h、SStudent.h、STime.h 6个头文件和SAccount.cpp、SConsole.cpp、SManage.cpp、SMenu.cpp、SStudent.cpp、STime.cpp、main.cpp 7个源文件组成。

SAccount.cpp、SConsole.cpp、SManage.cpp、SMenu.cpp、SStudent.cpp、STime.cpp包含对应的头文件SAccount.h、SConsole.h、SManage.h、SMenu.h、SStudent.h、STime.h。

SMenu.h、STime.h包括SConsole.h头文件。

SManage.h包括SStudent.h、SAccount.h、SMenu.h头文件。

## **程序设计**

4.2.1 Account类

设计意图：解决3.1.1的关键问题技术。 运用到的关键知识点：I/O流类库和文件操作。主要函数：函数CreatAccount可以创建账户；函数hidePassword可以隐藏输入密码，用\*代替密码，支持退格重新输入；函数save\_account可以保存账号信息到本地磁盘；函数input\_account从磁盘将信息读入；函数DeleteAccount可以删除账户的本地磁盘和内存中的信息；函数ChangePassword可以修改密码；函数verifyInformation用来验证账户信息；函数find\_filename可以用来查找文件名。

4.2.2 Menu类

设计意图：提供菜单，使得界面更加友好。主要函数：函数settingConsole可以设置控制台；函数sign\_in\_menu登录菜单；函数operate\_menu操作菜单；函数about\_menu关于菜单；函数find\_menu查询菜单；

函数statistic\_menu统计菜单；函数sort\_menu排序菜单；函数edit\_menu修改菜单；

4.2.3 Student类

设计意图: 对学生群体进行属性和行为抽象可以得到Student类，Student类对姓名、性别和籍贯等数据进行封装形成新的数据类型。运用到关键的知识点：类与对象，抽象和封装，数据共享与保护。主要函数：重载运算符为类的友元函数operator<<和operator>>对整个类对象进行输入输出操作，为后续程序输出提供便捷，增加程序可读性。

4.2.4 Time类

设计意图：显示当前时间和日期，显示启动界面。主要函数：getTime函数可以获得系统当前时间；showTime函数可以连续显示当前时间以及程序启动界面。

4.2.5 Console类

设计意图：统一定制控制台界面的名称，颜色，尺寸，位置等信息。主要函数：SetCord设置输出的位置、HideCursor光标的开启与关闭、SetTitle设置标题名称、SetSize设置控制台大小、SetColor设置颜色、CleanScreen清屏、PauseScreen冻结屏幕。

4.2.6 Manage类

设计意图：实现对学生信息的录入、浏览、排序、查询、删除、

修改以及统计。

运用到的关键知识点：文件操作、I/O流类库、函数模板、STL标准模板库以及异常处理。

主要函数：函数record录入信息、函数browse浏览信息、函数find查询信息、函数chart输出统计报表、函数deletes删除信息、

函数change修改信息、函数input从磁盘文件导入信息、函数save保存信息到磁盘文件、函数sort\_date按出生日期排序、函数sort\_number按学号进行排序、函数statistic统计学生信息。

## **程序实现**

4.3.1 隐藏输入密码，用\*号代替密码，支持退格重新输入

该功能主要由函数\_getch()和putchar()函数实现，\_getch()可以使用户的输入不显示在屏幕上，其包含在conio.h头文件。而putchar()函数可以将指定字符输出到屏幕上。因此二者的结合使用，用\_getch()将用户的输入读走，同时用putchar()将“\*”输出到屏幕上。当\_getch()读到回车键的时候结束。

对于退格再输入的实现，可以将\_getch()读到的字符与退格符比较，如果\_getcah()读到是退格符，则用putchar()输出一个退格符。此时光标位于字符“\*”之前，再用putchar()输出一个空格符，将字符“\*”覆盖，以达到删除效果。然后，用putchar()输出退格符回退一格等待输入。

代码实现如下：

void Account::hidePassword(string& password) {

char c;

int count = 0;

while ((c = \_getch()) != '\r') { //回车

if (c == 8) { //退格

if (count == 0) {

continue;

}

putchar('\b'); //回退一格

putchar(' '); //输出一个空格将原来的\*隐藏

putchar('\b'); //回退一格等待输入

password.pop\_back(); //同时删除密码的尾字符

count--;

}

if ((c >= 'a'&& c <= 'z') || (c >= 'A'&&c <= 'Z') || (c >= '0'&&c <= '9')) {//密码只能是字母和数字

putchar('\*');

password += c;

count++;

}

}

cout << endl;

}

4.3.2 查找文件名

查找文件名主要由结构体struct\_finddata\_t和\_findfirst、\_findnext以及\_findclose函数实现。包含在io.h头文件中。

结构体struct\_finddata\_t用来存储文件的各种信息：文件属性的存储位置，文件最后一次访问时间，文件第一次被修改时间，文件大小以及文件名等。 \_findfirst、\_findnext以及\_findclose合理搭配使用实现文件查找[3]。\_findfirst会返回一个long型,如果返回成功，将返回一个long型的唯一的查找用的句柄，这个句柄将在\_findnext中被使用。若失败，则返回-1。\_findnext函数若成功返回0，失败返回-1。\_findclose函数关闭由\_findfirst函数创建的句柄[3]。

以上知，可以先用\_findfirst查找第一个文件，若成功则用返回的句柄调用\_findnext函数查找其他文件，当查找完毕后，用\_findclose关闭由\_findfirst函数创建的句柄，结束查找[3]。

代码实现如下：

bool Account::find\_filename(string filename) {

bool flag = false; //"没有找到账户!请重新注册账户"

string s;

long lf;

\_finddata\_t File; //\_finddata\_t存储文件各种信息的结构体，io.h中的结构体

//搜索路径中符合的名称，“.txt”".

if ((lf = \_findfirst(".\\Password\\\*.txt", &File)) == -1) //\*.txt代表任意后缀为.txt的文件...

flag = false; //"没有找到账户!请重新注册账户"

else {

do{

if ((filename+".txt") == File.name) {//filename加上.txt后缀

flag = true;

break;

}

} while (\_findnext(lf, &File) == 0);

}

\_findclose(lf); //关闭由\_findfirst函数创建的一个搜索句柄

return flag;

4.3.3 排序

对于学生信息排序，可以用函数sorts来实现，函数核心使用了标准模板库的排序函数sort,问题是sort函数仅支持vector、string以及deque的排序，而由红黑树实现的set、map、multimap等容器，元素本身有序（即映射按键值递增），故不支持sort排序, 而我们采用multimap来存储学生信息，想要按学号降序排序，按出生日期排序便很难实现。由于sort支持对vector的排序，所以可以使用vector和pair的混合使用来解决这一问题。pair作为vector的内置数据类型，它的第一个数据的类型为string，第二个数据类型为Student。不难发现，pair的数据类型和multimap的键值数据类型相同。在排序时，可以将multimap中的数据转存到pair中，然后使用排序函数sort对vector排序。

当标准模板库中的sort函数的内置类型为自定义类型或容器时，其函数对象便要由使用者提供。为此我们需要设计两个函数对象CmpByValueDate和CmpByKeyNumber来决定sort递增或递减。

代码实现如下：

//给pair声明别名

typedef pair<string, Student> PAIR;

//定义函数对象，按出生年月进行比较,递减

struct CmpByValueDate{

bool operator()(PAIR& lhs, PAIR& rhs) {

return lhs.second.getDate() > rhs.second.getDate();

}

};

//定义函数对象，按学号进行比较,递减

struct CmpByKeyNumber{

bool operator()(PAIR& lhs, PAIR& rhs) {

return lhs.first > rhs.first;

}

};

//按学号进行排序

void Manage::sorts\_number() {

//将multimap中的元素转存到vector中

vector<PAIR> number\_student\_vec(number\_student\_multimap.begin(), number\_student\_multimap.end());

//在STL标准容器中只有vector、string、deque可以使用sort

sort(number\_student\_vec.begin(), number\_student\_vec.end(), CmpByKeyNumber());

//浏览信息

browse(number\_student\_vec.begin(), number\_student\_vec.end());

}

//按出生年月日排序

void Manage::sorts\_date() {

//将multimap中的元素转存到vector中

vector<PAIR> number\_student\_vec(number\_student\_multimap.begin(), number\_student\_multimap.end());

//在STL标准容器中只有vector、string、deque可以使用sort

sort(number\_student\_vec.begin(), number\_student\_vec.end(), CmpByValueDate());

//浏览信息

browse(number\_student\_vec.begin(), number\_student\_vec.end());

}

# 系统测试

## **测试概要**

在Visual Studio 2017环境下进行测试，并配置相关工具。在测试中尽可能的选用大量典型数据以及边界数据去测试系统，使测试范围更广，测试数据更全面。将系统分解为多个功能模块，逐一对各个功能模块进行测试。通过5-6个测试人员对系统进行测试，减小测试偶然性，增加系统的可靠性。

## **测试环境与配置**

所使用的测试环境是Visual Studio 2017，配置是实时调试器、VC++2017v141工具集（X86,X64）、C++分析工具、Visual C++ATL支持、用于CMake的Visual C++工具等。

## **测试方法和工具**

使用大量典型数据去测试系统，以及使用一些边界数据去测试系统，记录测试结果，分析测试结果。

## **系统功能分解**

将系统功能分解为以下11个部分：

1. 启动界面：模拟程序启动以及显示当前时间。
2. 登录界面：模拟注册系统账户和登录系统。
3. 主菜单: 包括关于、录入、排序、查询、统计、删除、修改等选项。
4. 录入学生信息：录入学生学生信息并保存到文件
5. 浏览学生信息：浏览所有学生信息。
6. 排序学生信息：按学号或出生年月排序学生。
7. 查询学生信息：按学号、姓名、性别、出生年月、籍贯查询。
8. 统计学生信息：按性别、出生年月、籍贯、班级、政治面貌统计学生。
9. 删除学生信息：删除指定学号的学生所有信息，同时删除文件中对应的信息。
10. 修改学生信息：修改对应学号的学生的信息。
11. 修改密码: 可以修改账户的账号和密码，且账户内学生信息不会变更。

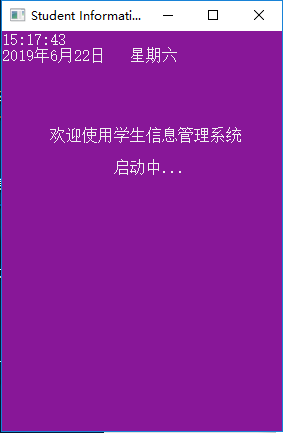
## **测试内容**

1. 功能性测试

（1）模块名：启动界面

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 启动界面 | 无 | 启动成功 | PASS |  |
| 显示当前时间 | 无 | 显示成功 | PASS |  |

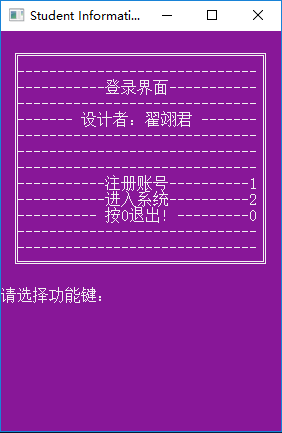
针对启动界面的测试，选择测试时间为15:17:43，时间显示正常，程序启动功能正常，



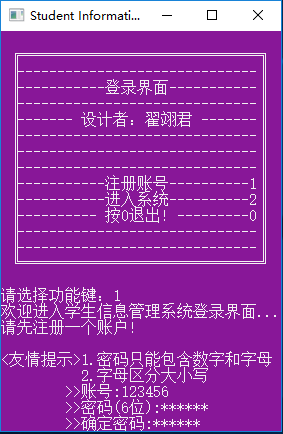
1. 模块名：登录界面

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 注册账户 | 数字或字母 | 注册成功 | PASS |  |
| 登录系统 | 数字或字母 | 登录成功 | PASS |  |

登录界面如下：

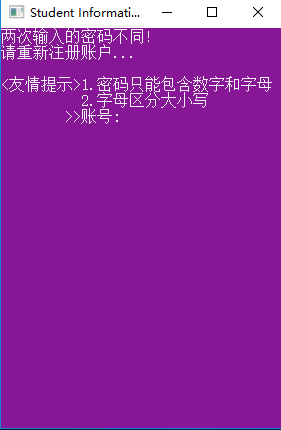


对于注册账号功能的测试：注册账户时，程序会给用户友情提示，防止用户输入无效字符，当输入无误情况下，程序运行如下：



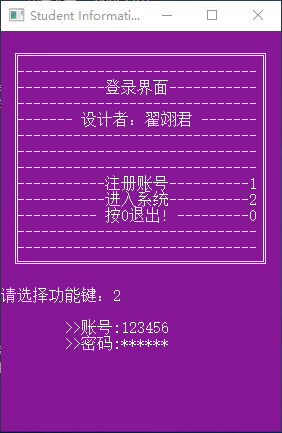
在注册账户时，用户可能往往会手误输错密码，导致当前密码与确认密码不一致，针对这一问题做如下测试：

当输入有误时（例如:密码和确认密码不一致），程序运行如下：

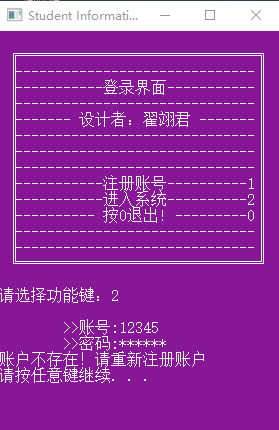


对于登录系统功能正的测试：

当输入账号存在，密码正确时，系统正常登录，如下图:



当用户进行输入时，往往会手误输错，使得输入账号不存在，此时程序会反馈“账户不存在”，提醒用户重新注册账户，程序运行如下:



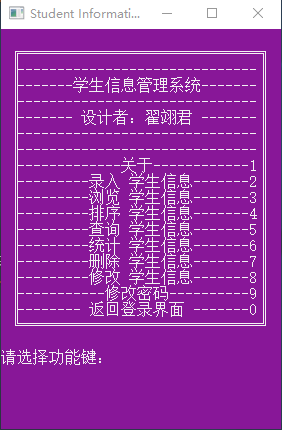
当输入密码不正确时，程序反馈“账号或密码不正确”，提醒用户重新输入，程序运行如下：



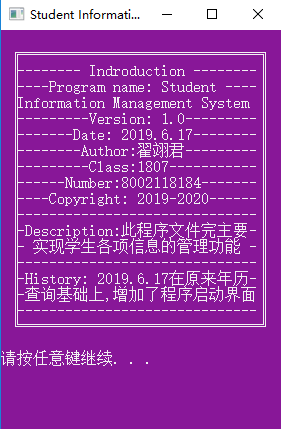
（3）模块名：主菜单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 显示主要功能 | 数字0-9 | 显示相应功能 | PASS |  |
| 关于 | 1 | 显示程序基本信息 | PASS |  |

显示功能正常，如下：



输入1，程序运行如下：



（4）模块名：录入学生信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 录入学生信息并保存到文件中 | 学生基本信息 | 录入成功 | PASS |  |

对于录入学生信息的测试如下：



（5）模块名：浏览学生信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 浏览学生信息 | 无 | 所有学生信息表 | PASS |  |

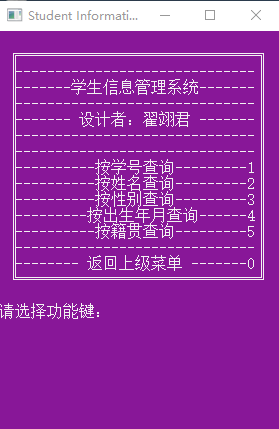
对于浏览学生信息的测试如下：



（6）模块名：查询学生信息

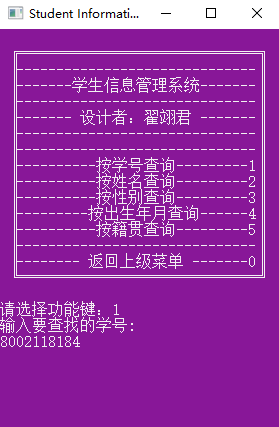
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 查询学生信息 | 0-5 | 待查询学生信息 | PASS |  |

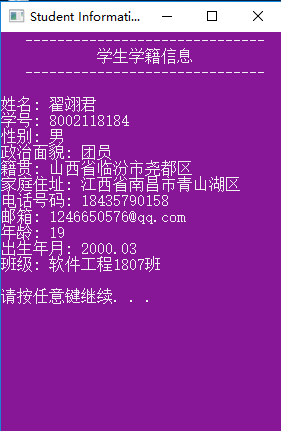
查询菜单界面如下：



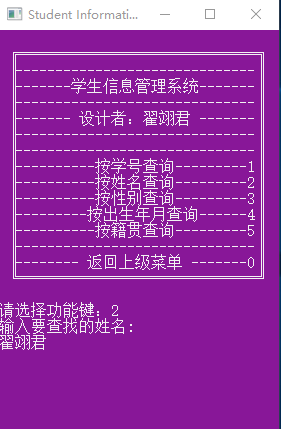
以下是针对不同查询方式所做的测试

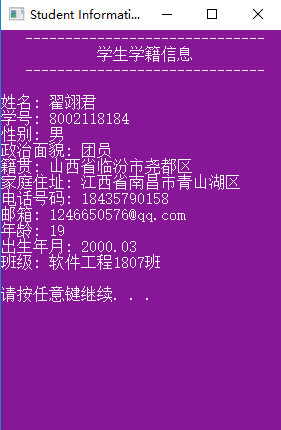
输入1，程序运行如下：



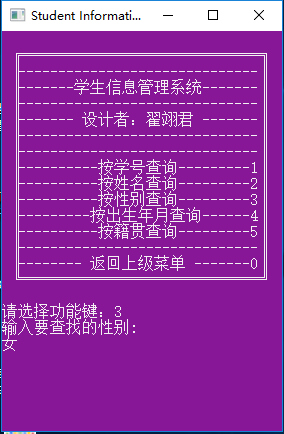


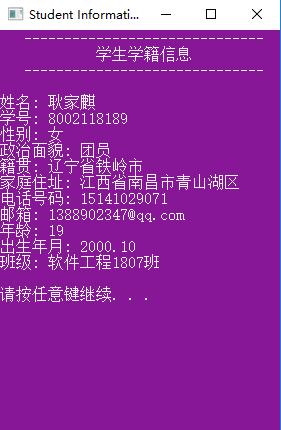
输入2，程序运行如下：



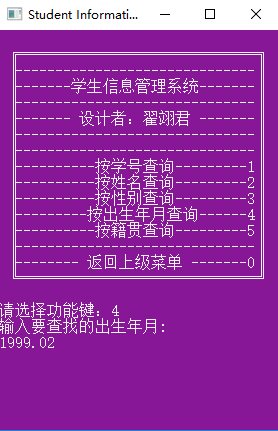


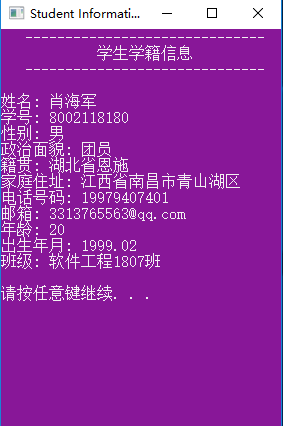
输入3，程序运行结果如下



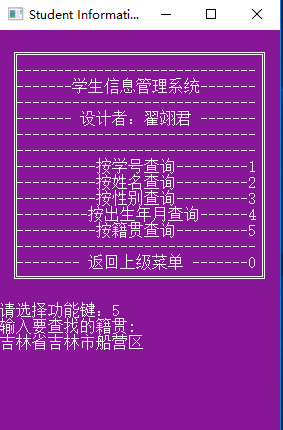


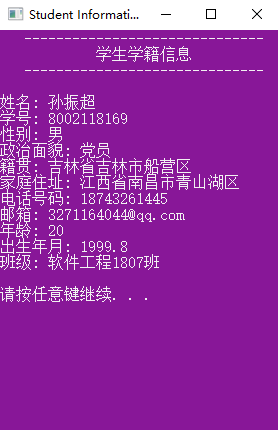
输入4，程序运行如下：



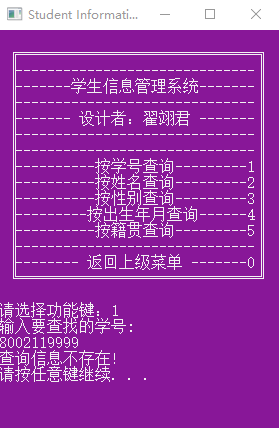


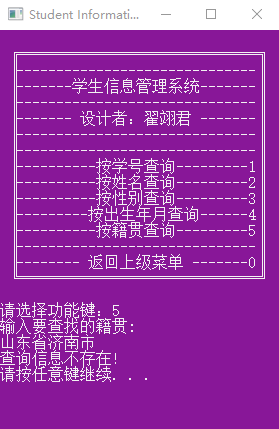
输入5，程序运行如下：





以上是所查找信息均存在的情况，当我们所查找信息不存在时，程序会进行异常处理，反馈信息“查询信息不存在”（以学号和籍贯查询为例，其他情况也一样），程序运行如下：

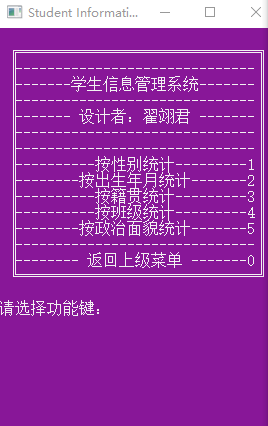




（7）模块名：统计学生信息

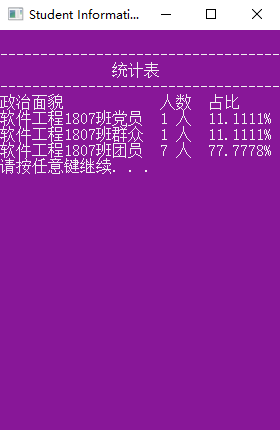
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 统计学生信息，输出统计表和占比 | 0-5 | 统计学生信息 | PASS |  |

查询菜单界面如下：



按选定的标准统计学生信息，程序会统计出满足这一标准的所有学生信息，计算出占比，输出统计报表（这里仅以按出生年月和按政治面貌统计为例，其他情况也一样）

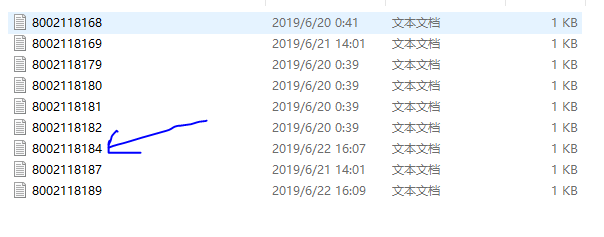




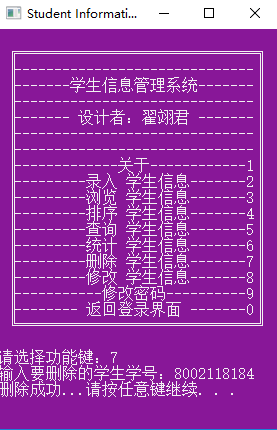
（7）模块名：删除学生信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 删除学生信息 | 学生学号 | 删除成功/或学号不存在 | PASS |  |

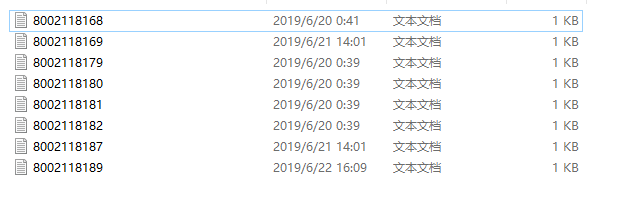
删除学生信息这一模块界面比较简单，只需输入需要删除的学号，一旦删除便无法找回。在删除内存中的信息的同时，也会删除对应文件中的信息。以删除学号为8002118184和删除8002119999。前者学号信息存在，而后者学号信息不存在，当输入学号不存在时，程序会进行异常处理。以下是二者的程序运行图以及磁盘文件的变化。删除8002118184，其对应磁盘文件存在如图：



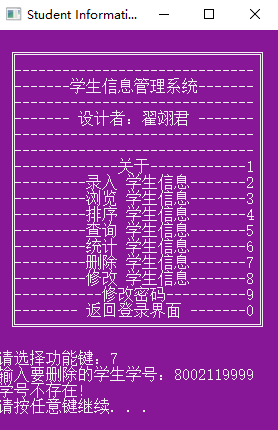
删除该学号信息，程序反馈“删除成功”，程序运行图如下



删除该学号信息后，学号对应磁盘文件同时也被删除。



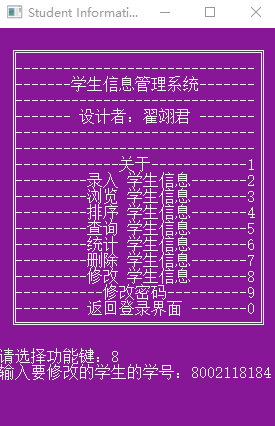
当我们删除的学号8002119999不存在时，程序反馈“学号不存在”程序运行图如下

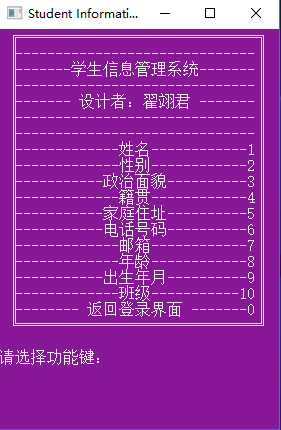


（8）模块名：修改学生信息

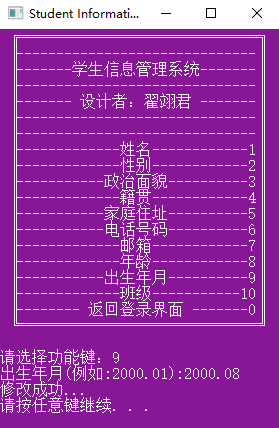
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 修改学生信息 | 学生学号 | 修改成功/或学号不存在 | PASS |  |

此功能与删除学生信息一样，也需要输入学号。因此输入正确的学号，才能修改学号对应的信息，但输入的学号不正确，程序依然会进行异常处理，反馈“学号不存在”。输入错误学号的情况这里不再介绍，以学号8002118168为例，程序进入“修改菜单”。





这里只以修改出生日期为例（其他情况也一样处理），将出生日期修改为2000.08，程序会反馈“修改成功”



在修改日期的同时，磁盘文件对应的日期也会被修改，如下



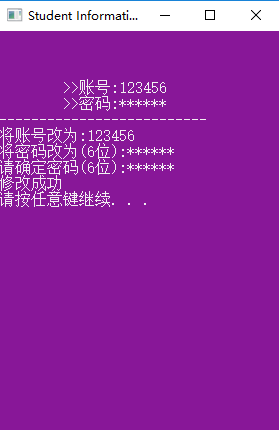
（8）模块名：修改密码

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **预期输入** | **预期输出** | **实际结果** | **备注** |
| 修改账号密码 | 数字或字母 | 修改成功 | PASS |  |

为了确保账号信息安全，在修改密码前需要验证账户的账号和密码（如下图）。同样，当输入账号或密码不正确时，程序反馈“账号或密码错误”，提醒用户重新输入。

修改密码的操作和注册密码相同，这里不再介绍。





1. 性能测试

压力测试：具体测试数据见： 8002118184-软件工程1807班-翟翊君-C++程序设计课程任务书及报告/Student Information Management System/ReadMe/Test Data的文本文件。

根据测试数据显示，系统在不断施加压力的情况下，系统依然能够进行良好运行；当系统接受一些非法输入或边界数据时，系统依然能够提供服务。这有赖于系统具有良好的容错能力。

稳定性测试：系统在持续运行过程中系统无异常情况发生，无突发事件发生，在意外断电、事件中断等情况下系统完备性良好。

# 个人小结

在用C++语言编写学生信息管理系统的过程中，感受颇深，获益良多。体会到了面向对象的编程思想和泛型编程的思想对C++程序设计的重要性。深刻意识到C++语法与C语言语法之间的诸多区别和诸多相同之处。领会到了抽象、封装、继承以及多态在C++程序设计中占有举足轻重的地位。最后重点学习了STL标准模板库、文件操作、I/O流类库以及异常处理等知识。

将上次设计的万年历查询系统与本次设计的学生信息管理系统相比较，不难发现，前者的难点侧重于算法的实现，而后者的难点侧重于数据结构的实现。因此，前者在程序设计上侧重于类的实现与组织，而后者在程序设计上侧重于对STL标准模板库、文件操作以及I/O流类库等知识的灵活应用。

正如郑莉老师所言“写程序总不能一直从刀耕火种开始”，要学会使用已有的程序，这样可以极大得提高我们写程序的效率。本次对学生信息管理系统的设计，便应用了上次万年历所设计的时间类和控制台类。对这些类的使用，提高了我编写学生信息管理系统的效率，让我深刻感受到类的设计所体现的抽象，封装的作用。抽象和封装不仅是面向对象的程序设计的重要特性，也是代码重用的一种方式。

本次程序设计的难点是数据结构的设计以及数据的处理。从刚开始拿到选题的不知所措，到一点一点的摸索，到最后程序的实现，其实是经历了许多的绊脚石。第一个绊脚石是关于类的设计和组织，学生信息管理系统是一个实时系统，因此需要一个账户类，账户类不仅可以注册账号，还应具有登录系统、修改密码等账号管理功能。这样学生信息管理系统的第一部分的框架便搭建完成了。其中遇到困难是如何实现密码隐藏输入和查找文件名。通过阅读和查找相关资料，得到了与输入输出和文件操作的相关的知识，然后经过精心的算法设计，解决了这两个困难。这一过程不仅解决了我所面临的困难，同时也弥补了我所欠缺的知识。

第二个绊脚石是关于数据结构的设计。学生信息管理系统所处理的对象是学生。因此，对学生这一群体进行属性和行为抽象，可以设计一个学生类来表示学生。然而，学生信息管理系统是能够对学生信息进行录入、查询以及删除等操作的。因此，设计一个管理类表示这些功能。管理类的函数所处理的对象是学生，所以学生类和管理类进行组合，学生类对象作为管理的对象。学生信息管理系统并针对某一个学生的信息，而是针对许多学生的信息。故，选用容器来存储学生类对象比较合适。可是，选择哪种容器更合适。不难发现，我们用字符串来表示学生的各种基本信息，而学号是学生的唯一标识，即学号与学生所有信息一一对应。这是一种简单的映射关系。因此，用映射容器表示这一关系便无可厚非了。

第三个难点是关于文件操作。通过不断查阅C++手册，了解关于文件操作的许多知识。同时也解决所面临的问题，使得编写程序得以进行。

虽然编写程序会遇到许多难点，有可能会失败，也有可能会成功。但只有从一次又一次的实践到一次又一次失败，不断解决问题，才有可能成功。在这一过程中自己会获益良多。坚持是一种解决问题的方法。当然，写程序也是一件很快乐和幸福的事。

# 参考文献

1. 郑莉,董渊,何江舟.C++语言程序设计(第四版)[M].北京.清华大学出版社,2010.
2. 胡凡,曾磊.算法笔记(第一版)[M].北京.机械工业出版社.2016.7
3. C++ reference.[OL].<http://www.cplusplus.com/reference/>