



西安电子科技大学
XIDIAN UNIVERSITY

面向对象程序设计

Object Oriented Programming

实验报告

Experimental Report

学号	18130500143	姓名	陈凌灏
班级	1803052	任课教师	褚华
实验名称	面向对象程序设计“校园一卡通乘车模拟系统”		
实验学期	2019-2020 学年第二学期		
实验日期	2020.05.14	实验地点	课程群
报告成绩		评分教师	

一、实验环境

DELL G3 3579 PC 机;

Windows 10-64 位操作系统;

计算机内存: 8192MB;

中央处理器: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz(12CPUs) , ~2.2GHz;

集成开发环境: DEV C++。

二、实验内容

建立校园一卡通乘车模拟系统。系统可以管理三类人员信息: 教师、学生、家属, 且每人仅持有一张一卡通。相关人员可以通过系统进行乘车、充值、创建账户、注销账户等操作。在金额不足的情况下家属、学生不能乘车, 家属每月可以免费乘车 20 次。

此外, 需要模拟的操作包括公交车的行为。相关人员只能在规定的时段内进行乘车, 不能超过车上的限载人数, 且上车人员需要进行扣费(教师免费)。

通过面向对象的程序设计范型, 采用 C++编写程序。

三、实验步骤

3.1 系统设计与实验构思

• 代码版本管理

采用 Github (<https://github.com/LinghaoChan/Cpp-Final-Homework>)进行代码管理与版本控制, 避免由于编码格式、功能紊乱对系统开发的影响。

• 系统层次结构设计

系统主要采用“树形”的层次结构进行功能分支。整体的层次结构如图一所示(可放大查看)。

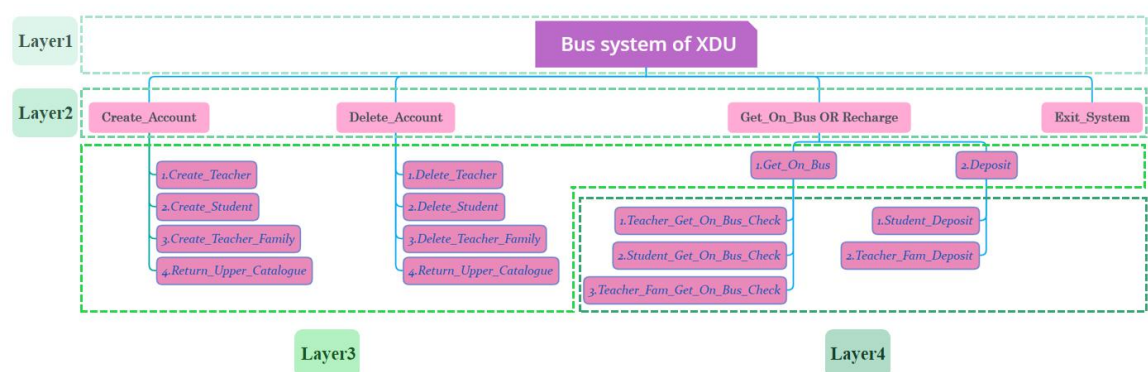


图 3.1 系统功能结构示意图

图一所示的系统分为 4 个大模块: 创建账户、注销账户、乘车或充值、退出系统。**创建账户**涉及的功能包括: 教师创建账户、学生创建账户、家属创建账户、返回上级;**注销账户**教师注销账户、学生注销账户、家属注销账户、返回上级;**乘车**涉及的功能包括: 教师乘车、学生乘车、家属乘车、返回上级;**充值**涉及的功能包括: 学生充值、家属乘车、返回上级。

创建账户的功能分支示意图如图 3.2 所示。在系统文件中,已存在用户基本信息文件(类似学籍信息),用户需要输入姓名、编号进行双重验证(避免重名的现象)。基本信息验证通过之后,方可在系统进行注册。用户需要输入 2 次密码对密码进行确认,避免输入出错对用户的影响。经过验证之后,用户信息创建成功返回上级目录。

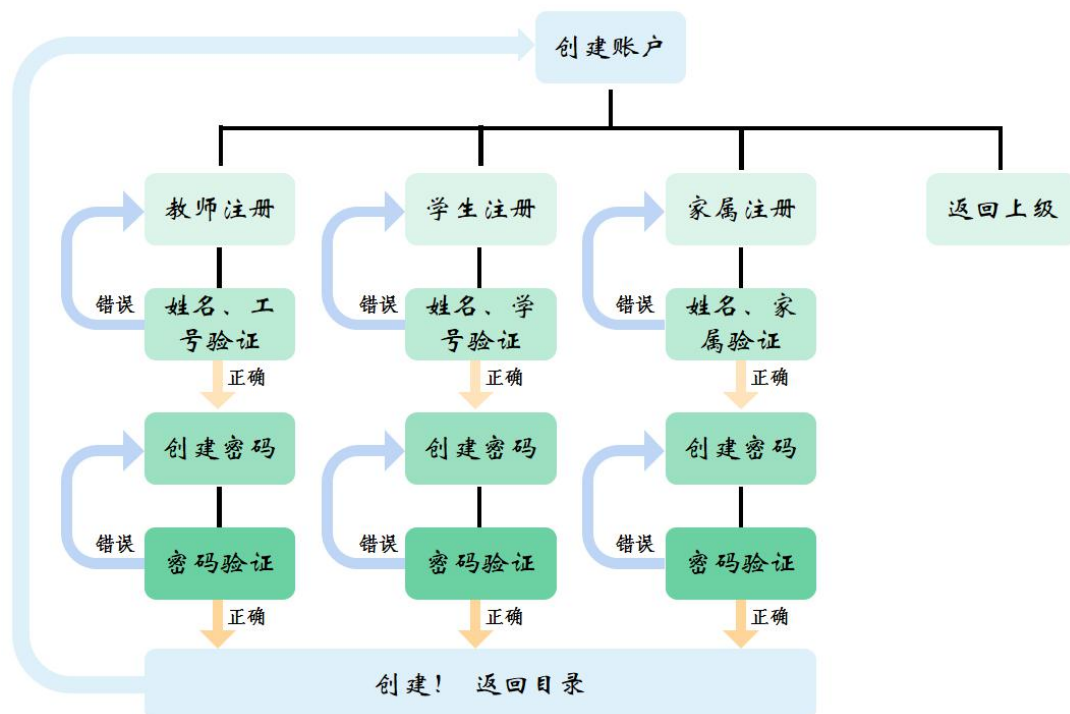


图 3.2 账户创建逻辑示意图

注销账户的功能分支示意图如图 3.3 所示。注销账户的用户逻辑要略显简单。相关人员需要通过输入姓名、账号对基本信息进行验证,通过输入密码对用户信息进行验证,以避免用户 A 在知道用户 B 的信息的前提下将 B 的账户进行注销,保证系统的安全性。

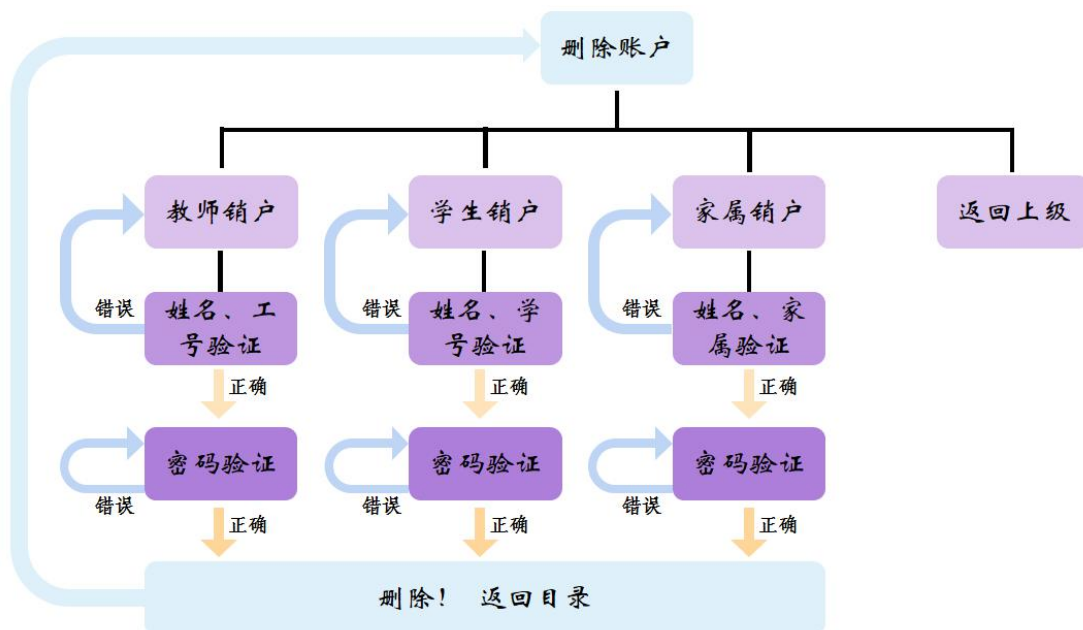


图 3.3 账户注销逻辑示意图

乘车或充值的功能分支示意图如图 3.4 所示。当用户选择乘车时，检查是否处于运营时段，若不满足直接返回上级目录。当处于运行时段时，用户可以选择当前位置，判断车辆是否到达、离开或处于当前站点。若处于当前站点，系统通过调用函数指针数组对不同用户的不同乘车方式函数进行调用，提高代码的简洁性。最后，车辆人数自增，相关人员扣费后返回上级目录。当用户选择充值时，通过验证信息和密码验证之后，可以使用函数指针数组对不同用户进行充值。

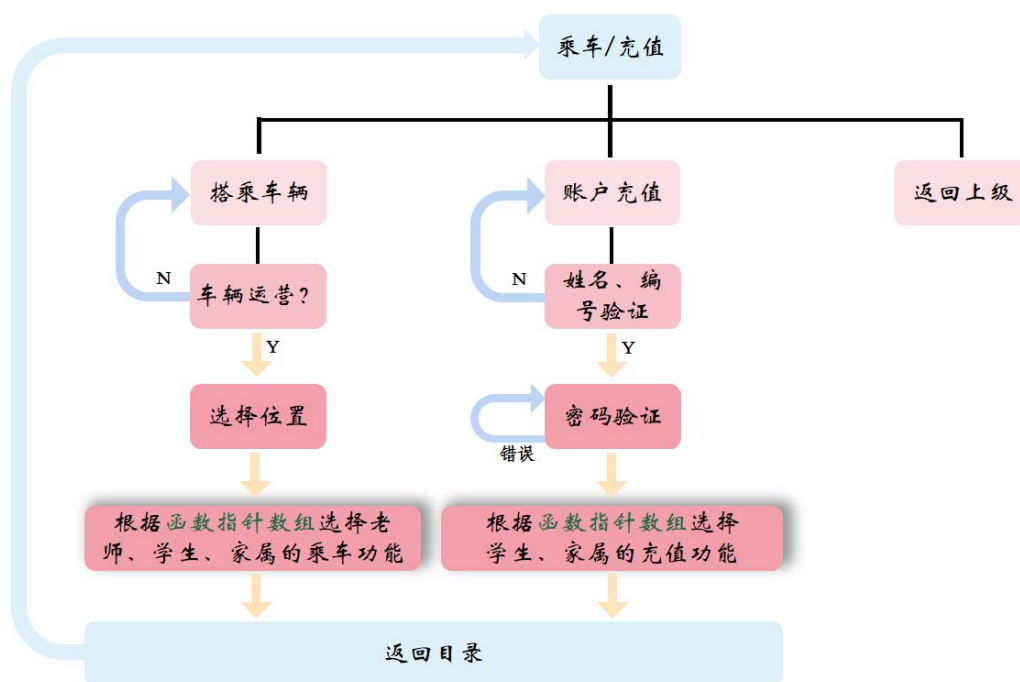


图 3.3 用户乘车或充值功能示意图

系统可采用按任意键返回的功能退出系统。

3.2 功能实现

• class Bus(公交车)

公交车的路线和停靠时间如图 3.4 所示。车辆从每小时 0 分出发，10~20min 内在教学楼停靠，35~45min 内在宿舍楼停靠。

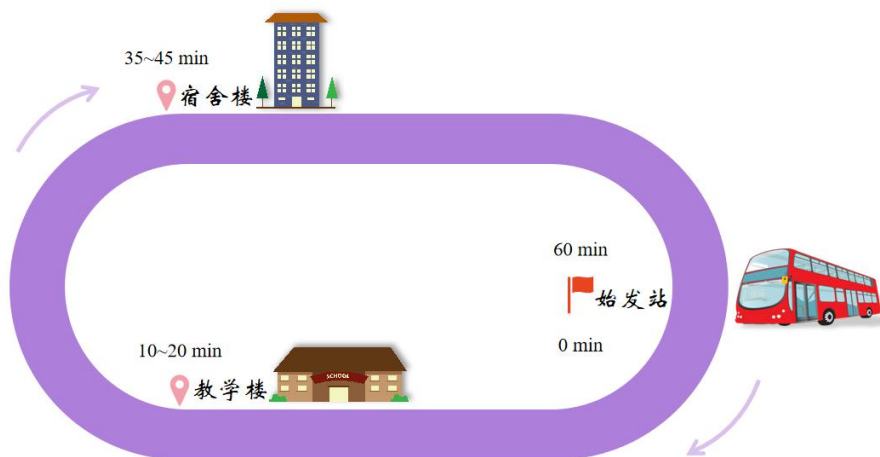


图 3.4 公交车停靠时间示意图

class Bus 包括 7 个属性和 8 个成员函数。其定义声明如表 3.1 所示。

表 3.1 class Bus 的成员列表

分类	名称	参数	type/返回值	功能、含义
属性 private	number_of_passengers	—	int	统计乘客数
	size	—	int	20/50 人车型
	count	—	int	车辆编号
	place	—	double	车辆位置
	id	—	string	车牌
	brand	—	string	车辆品牌
	driver	—	string	司机名字
函数 public	Bus	int un_permit	None	构造函数
	Get_Bus_Id	void	string	获取车牌
	Get_count	void	int	获取编号
	Get_number_of_passengers	void	int	获取车上人数
	Get_place	void	double	获取地点
	Show_Status	void	void	打印状态
	Get_On_One_Person	void	void	乘客人数增加
	Get_Status	void	bool	查看是否满员

系统所涉及的算法中，Bus 的构造函数较为复杂，其算法伪代码如 Algorithm1 中所示。由于在 Bus_Message.txt 中已经存在有 50 辆车的基本信息。在模拟行为的时候，由于不能初始化 2 辆同样的车，这里需要在初始化第二辆车的构造函数参数列表中输入不合适的车辆编号(指第一个 Bus 对象的车辆编号)。为保证一致性，在调用的时候，第一辆车的参数取 51 即可。同时还需要根据系统时间得到车辆信息。

Algorithm 1: Constructor of Bus

Input Parameter: 非法车辆编号 un_permit //第 1 辆车取 51, 第 2 辆车取第一辆车的编号

1: while ((choice = 0~50 的随机数) != un_permit):

2: : /*busy waiting, no commands*/

3: end while

4: for line in lines:

5: if line 编号 == choice):

6: 为对象属性赋值;

7: end if

8: line++;

9: end for

Construction Finish

• class People(人, 基类)

由于 Student、Teacher、Teacher_Fam(家属)三个类具有一些共同的属性，因此确定了 3 个共同属性：name、sex、id，均为 string 类型。People 类的三个派生类之间在系统内部的行为差别较大(如计费方式等)，故只重载了 People 的构造函数，便于 People 的派生类的构造函数调用。

由 People 类继承得到的派生类形成的继承树如图 3.5 所示。

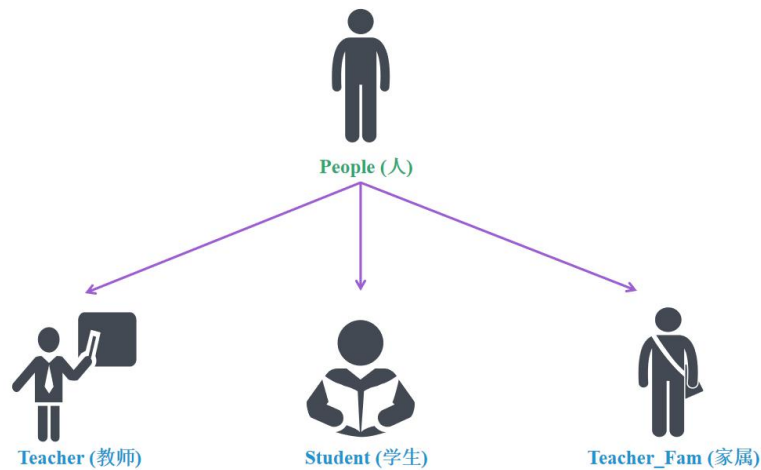


图 3.5 系统中类继承关系示意图

• class Student(学生, 派生类)

除继承 People 类已经具有的属性及方法外, Student 类定义的其他属性与方法如表 3.2 所示。

表 3.2 Student 类定义的成员列表

分类	名称	参数	type/返回值	功能、含义
属性 private	college	—	string	学院
	password	—	string	密码
	times	—	int	乘车次数
	money	—	double	账户余额
函数 public	Student	7 个 string	None	构造函数
	~Student	string	None	析构函数
	Student_Times_Plus	string	string	次数自增
	Stuent_Money_Decrease	string	string	乘车扣费
	Student_Deposit	double	string	充值
	Show_Status	void	void	打印状态
	Student_Clear_Times	void	static void	每月统计次数清零

由于统计的乘车次数为每月乘车次数, 因此要每月进行清零, 其算法流程如 Algorithm 2 所示。

Algorithm 2: Student_Clear_Times

Input Parameter:

if get_system_month != MONTH:

 MONTH = get_system_month;

 New_string = "";

//定义空串

 for line in lines:

 New_string += clear_time(line);

//清零后的学生信息字符串拼接

 end for

 New_string -> file

//写入文件

end if

Student_Clear_Times Finish

• class Teacher(教师, 派生类)

除继承 People 类已经具有的属性及方法外, Teacher 类定义的其他属性与方法如表 3.3 所示。

表 3.3 Teacher 类定义的成员列表

分类	名称	参数	type/返回值	功能、含义
属性 private	college	—	string	学院
	password	—	string	密码
	times	—	int	乘车次数
函数 public	Teacher	6 个 string	None	构造函数
	~Teacher	void	None	析构函数
	Teacher_Times_Plus	string	string	次数自增
	Show_Status	void	void	打印状态
	Student_Clear_Times	void	static void	每月统计次数清零

虽然 Teacher 类中部分函数的实现逻辑也较为复杂, 也和 Student 类的数据结构、文件读取方式有所不同, 但是算法较为相似, 本文不做详述。

• class Teacher_Fam(家属, 派生类)

除继承 People 类已经具有的属性及方法外, Teacher_Fam 类定义的其他属性与方法如表 3.3 所示。

表 3.3 Teacher_Fam 类定义的成员列表

分类	名称	参数	type/返回值	功能、含义
属性 private	money	—	double	学院
	password	—	string	密码
	times	—	int	乘车次数
函数 public	Teacher_Fam	6 个 string	None	构造函数
	~Teacher_Fam	void	None	析构函数
	Teacher_Fam_Times_Plus	string	string	次数自增
	Teacher_Fam_Money_Decrease	string	string	乘车扣费
	Get_Times	void	int	获取乘车次数
	Teacher_Fam_Deposit	double	string	充值
	Show_Status	void	void	打印状态
	Student_Clear_Times	void	static void	每月统计次数清零

虽然 Teacher_Fam 类中部分函数的实现逻辑也较为复杂, 也和 Student 类的数据结构、文件读取方式有所不同, 但是算法较为相似, 本文不做详述。

• class System(系统)

System 是系统类, 在 main 函数入口处调用静态方法检测是否月份更新, 若更新则用户本月的乘车次数清零。同时, 生成一个系统对象, 产生用户的系统入口。系统的成员变量只有 3 个, 一个为 MONTH(上一次系统关闭时的月份), 是 System 类的静态变量, 其余二者为函数指针, 负责函数跳转。由于系统函数的跳转和层级嵌套, 该类的成员函数较多, 本文只对其中实现功能复杂的成员函数进行说明。部分类成员可在附件中查看。System 类中的关键成员如表 3.4 所示。

表 3.4 System 类的部分成员列表

分类	名称	参数	type/返回值	功能、含义
属性 private	MONTH	—	static int	月份
	*Function_Get_On_Bus_pointer[3]	Bus&,Bus&	函数指针	乘车函数跳转
	*People_Deposit_pointer[2]	—	函数指针	充值函数跳转
函数 public	System_choice	void	void	功能选择
	Create_Teacher	void	void	创建教师账户
	get_password	void	string	获取密码
	Teacher_Fam_Get_On_Bus_Check	Bus&,Bus&	bool	家属乘车
	Teacher_Fam_Deposit	void	void	家属充值

对于**系统选项选择**的算法，主要的问题是字符串的读入，采用 getchar()函数进行读取的时候读入一个字符后不会立即停止读取输入，而是要在输入完一个字符串之后读入一个换行符才停止。为增强程序的健壮性，采用循环读入的方式，不需要用户按回车即可实现功能选择，增强用户体验感，更逼近图形界面的效果。具体算法如 Algorithm 3 所示。

Algorithm 3: System_choice

Input Parameter: void

```

1:初始化公交车信息
2: print 功能选项信息           //有 4 项选择
3: choice = getch();
4:while(choice ∉ {'1','2','3','4'}):
5:     choice = getch();
6:end while
7:功能跳转
8:clear 命令行界面

```

Finish System_choice

获取用户密码(get_password)的代码段主要实现难度在于用户密码长度控制和用户按退格键时的影响，此外还需考虑密码不可见性。本文设计的获取密码算法如 Algorithm 4 所示。

Algorithm 4: get_password

Input Parameter: void

```

1:初始化字符数组 str, 字符 c, 长度 count = 0
2: while((c = getch()) != 13){           //ASCII 码 13 对应回车
3:     if(c==8 && count>0):               //按字符读入, 需要考虑退格
4:         count--;    print:退格;    continue;
5:     else if c != 8:
6:         print:'*';    str[count] = c;    count++;
7:     end
8:     if count >= 20:
9:         提示长度过长;    break;
10:    end if
11:end while
12:str[count] = '\0';

```

Finish get_password

在**用户创建**的过程中, 仅仅依靠用户输入信息创建用户是不可靠的。因为在根本不存在一个学籍姓名叫张三, 学号是 100023 的用户的情况下, 本文的系统是不应该为他创建一卡通账户的。因此需要引入人群基本信息数据, 只有在这个数据集中的用户方可在校园公交系统中创建账户。创建账户的算法流程如 Algorithm 5 所示(以创建家属为例)。

Algorithm 5: Create_People

Input Parameter: void

```

1:  while (true):
2:      scan: 基本信息;
3:      for line1 in 基本信息数据集:
4:          if 输入的基本信息与数据集中信息匹配:
5:              构造对象;      break;
6:          end if
7:      end for
8:      if 未匹配上信息:
9:          print: 非法用户      break;
10:     end if
11:     for line2 in 一卡通用户数据集:
12:         if 匹配成功:
13:             print: 已注册
14:         end if
15:     end for
16: end while
17: while (true):
18:     scan: 密码 1;      scan: 密码 2;
19:     if(密码 1 == 密码 2):
20:         写入文件;      提示成功;
21:     else if:
22:         提示密码错误;
23:     end if
24: end while

```

Finish get_password

上述内容为各个 class 在设计过程中涉及的主要成员及算法说明。

3.3 优化功能

• 函数指针数组

在系统编码过程中, 一共采用了 2 个函数指针数组进行函数功能的跳转, 优化了代码的结构, 减少了 if-else 语句的多层嵌套。下面介绍这两个函数指针数组的功能。

①: bool (System :: *Function_Get_On_Bus_pointer[3])(Bus&, Bus&)

该函数指针数组的功能是数显不同人员的乘车功能的跳转, 传入 2 辆公共汽车的引用, 然后向用户展示车辆信息。数组的成员集合是: {&System :: Teacher_Get_On_Bus_Check, &System :: Student_Get_On_Bus_Check, &System :: Teacher_Fam_Get_On_Bus_Check}。这样

就能通过用户选择选项直接进行功能跳转了，降低代码的复杂度。

②: void (System :: *People_Deposit_pointer[2])(void)

该函数指针数组实现的是系统中不同人员充值功能的跳转。数组的数据成员集合为：
{&System :: Student_Deposit, &System :: Teacher_Fam_Deposit}。优化的原理同①相同。

• 公共汽车初始化问题改善

在编码的初期，我将公共汽车的初始化语句(构造函数)放在了人员乘车的函数模块中。由于初始化是随机的，会带来一个问题：当 A 用户上车之后返回上级时，B 用户再次上车时弹出的车辆信息和 A 使用时所弹出的不一致(因为第 2 次又初始化了一次)。

因此，改进后的系统在一开始的时候就进行了车辆信息的初始化，并采用引用参数传递的方式将公交车信息向调用的函数进行传递，以实现在 A 用户上车之后车上人数自增，B 在返回上级的情况下都能确保公交车还是之前初始化的 2 辆车。

3.4 Bug 改进及用户体验改善

• 密码长度限制

本系统采用了最多不超过 20 位的密码。主要基于两点考虑：一是不便于存储与处理、而是可能会出现数组越界。因此系统在调试的时候，增加了判定密码位数的情况，减少了程序出错的风险。在创建账户的时候，若初始密码设定超过了长度，将提示错误并重新获取输入。

• 公交车重复出现的避免

在初始化公交车的时候，由于随机模拟的原因，可能会出现 2 辆车信息相同的情况。虽然这种情况的出现概率极低，但是还是会出现的。在系统的车辆信息库中有 50 辆已经存在的公共汽车，编号为 0~50。第二辆车实例化时的非法编号为第一辆车的编号即可达到目的；为保证程序设计的一致性，第一辆车的非法编号设定为 51 即可。(具体算法见 Algorithm 1)。

• 用户提示

为改善用户体验，优化程序提示。系统通过采用系统提示音、提示语句以及“按任意键返回”来提醒用户并提供用户缓冲思考时间。

四、实验结果

4.1 用户基本功能选择实现

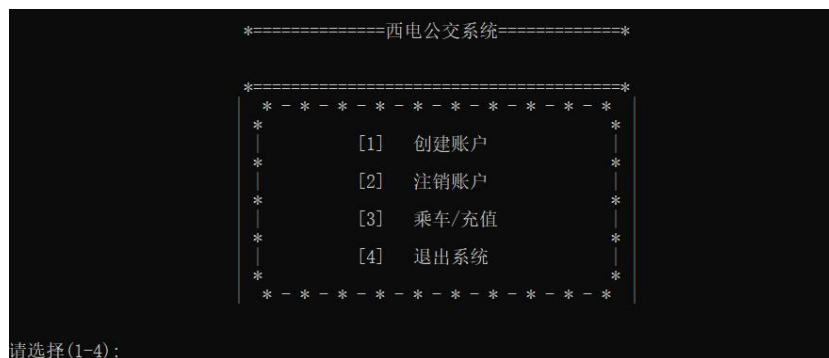


图 4.1 初始界面功能选择

4.2 教师账户创建

图 4.2(左)表示学校老师的基本信息管理, 图 4.3(右)表示在本文的系统中创建了账户的教师信息(包括姓名、编号、性别、学院、密码、本月乘车次数)。

刘坚 s2020024 m 计算机科学与技术学院	董洛兵 s2020006 m 计算机科学与技术学院 icml2020 11
刘伟 s2020025 m 计算机科学与技术学院	杜军朝 s2020007 m 计算机科学与技术学院 mobicom2020 2
刘宇鹏 s2020026 m 计算机科学与技术学院	范磊 s2020008 m 计算机科学与技术学院 focs2020 8
马继红 s2020027 f 计算机科学与技术学院	范莉莎 s2020009 f 计算机科学与技术学院 cikm2020 13
马玉麟 s2020028 m 计算机科学与技术学院	霍秋艳 s2020010 f 计算机科学与技术学院 kdd2020 23
上官红波 s2020029 m 计算机科学与技术学院	贾松卫 s2020011 m 计算机科学与技术学院 iccv2020 21
宋娟 s2020030 f 计算机科学与技术学院	蒋志平 s2020012 m 计算机科学与技术学院 sigir2020 15
宋胜利 s2020031 m 计算机科学与技术学院	张亮 s2020042 m 计算机科学与技术学院 1234 0
覃桂敏 s2020032 f 计算机科学与技术学院	鲍亮 s2020000 m 计算机科学与技术学院 infocom2020 3
王黎明 s2020033 m 计算机科学与技术学院	陈静玉 s2020001 f 计算机科学与技术学院 sigcom2020 6
王献青 s2020034 m 计算机科学与技术学院	陈平 s2020002 m 计算机科学与技术学院 nsdi2020 10
卫萌 s2020035 m 计算机科学与技术学院	樊凯 s2020054 m 计算机科学与技术学院 ccs2020 0
武波 s2020036 m 计算机科学与技术学院	邓岳 s2020004 m 计算机科学与技术学院 nips2020 11
徐悦胜 s2020037 m 计算机科学与技术学院	董大成 s2020005 m 计算机科学与技术学院 cvpr2020 53
严体华 s2020038 m 计算机科学与技术学院	
余春生 s2020039 f 计算机科学与技术学院	

图 4.2 教师基本信息与账户信息

4.3 教师信息与系统匹配

输入基本信息, 2 次输入密码“icde2020”, 显示创建成功跳转回上级目录。在账户信息文件中进行校正, 如图 4.4 所示, 增加了一条记录。

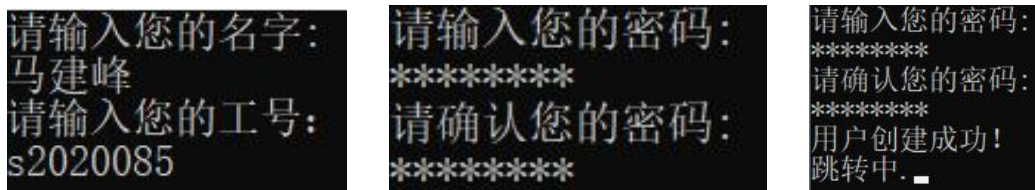


图 4.3 输入信息匹配流程结果

董洛兵 s2020006 m 计算机科学与技术学院 icml2020 11	董洛兵 s2020006 m 计算机科学与技术学院 icml2020 11
杜军朝 s2020007 m 计算机科学与技术学院 mobicom2020 2	杜军朝 s2020007 m 计算机科学与技术学院 mobicom2020 2
范磊 s2020008 m 计算机科学与技术学院 focs2020 8	范磊 s2020008 m 计算机科学与技术学院 focs2020 8
范莉莎 s2020009 f 计算机科学与技术学院 cikm2020 13	范莉莎 s2020009 f 计算机科学与技术学院 cikm2020 13
霍秋艳 s2020010 f 计算机科学与技术学院 kdd2020 23	霍秋艳 s2020010 f 计算机科学与技术学院 kdd2020 23
贾松卫 s2020011 m 计算机科学与技术学院 iccv2020 21	贾松卫 s2020011 m 计算机科学与技术学院 iccv2020 21
蒋志平 s2020012 m 计算机科学与技术学院 sigir2020 15	蒋志平 s2020012 m 计算机科学与技术学院 sigir2020 15
张亮 s2020042 m 计算机科学与技术学院 1234 0	张亮 s2020042 m 计算机科学与技术学院 1234 0
鲍亮 s2020000 m 计算机科学与技术学院 infocom2020 3	鲍亮 s2020000 m 计算机科学与技术学院 infocom2020 3
陈静玉 s2020001 f 计算机科学与技术学院 sigcom2020 6	陈静玉 s2020001 f 计算机科学与技术学院 sigcom2020 6
陈平 s2020002 m 计算机科学与技术学院 nsdi2020 10	陈平 s2020002 m 计算机科学与技术学院 nsdi2020 10
樊凯 s2020054 m 计算机科学与技术学院 ccs2020 0	樊凯 s2020054 m 计算机科学与技术学院 ccs2020 0
邓岳 s2020004 m 计算机科学与技术学院 nips2020 11	邓岳 s2020004 m 计算机科学与技术学院 nips2020 11
董大成 s2020005 m 计算机科学与技术学院 cvpr2020 53	董大成 s2020005 m 计算机科学与技术学院 cvpr2020 53
	马建峰 s2020085 m 计算机科学与技术学院 icde2020 0

图 4.4 用户创建前后信息对比

如图 4.5 所示, 当用户不存在信息库中时, 不允许在本文的系统中创建账户。

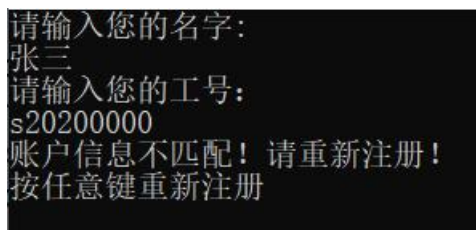


图 4.5 错误信息不允许创建账户

4.4 账户信息的删除

如图 4.6 所示, 输入账户姓名、学号、密码即可删除用户。如图 4.7 所示, 系统的记录

确实减少了该同学的信息记录。

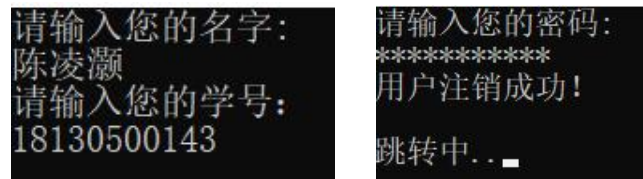


图 4.6 学生用户信息的注销

王子元 f 18140100226 计算机科学与技术学院 ziyuewang 0 0.0	王子元 f 18140100226 计算机科学与技术学院 ziyuewang 0 0.0
李佳乐 m 18130500017 电子工程学院 jialeli 0 0.0	李佳乐 m 18130500017 电子工程学院 jialeli 0 0.0
刘荣樵 m 18130500125 电子工程学院 rongqiaoliu 0 0.0	刘荣樵 m 18130500125 电子工程学院 rongqiaoliu 0 0.0
陈佳灵 f 18130500155 电子工程学院 jialingchen 0 0.0	陈佳灵 f 18130500155 电子工程学院 jialingchen 0 0.0
王秋知 m 18040500115 机电工程学院 zhiqiuwang 0 0.0	王秋知 m 18040500115 机电工程学院 zhiqiuwang 0 0.0
潘逸豪 f 18140100012 人工智能学院 yihaopan 0 15.720000	潘逸豪 f 18140100012 人工智能学院 yihaopan 0 15.720000
熊雨舟 f 18140100204 人工智能学院 222333 0 34.520000	熊雨舟 f 18140100204 人工智能学院 222333 0 34.520000
李悦涵 m 18140100198 人工智能学院 yuehanli 0 11.120000	李悦涵 m 18140100198 人工智能学院 yuehanli 0 11.120000
陈凌灏 m 18130500143 通信工程学院 linghaochen 7 22.400000	邹思远 m 18130500209 通信工程学院 siyuanzhou 1 7.000000
邹思远 m 18130500209 通信工程学院 siyuanzhou 1 7.000000	

图 4.7 删除前后系统信息的比对

4.5 乘车功能

图 4.8 表示在超出运营时间范围后，不允许乘车。

当日车辆已停运
早上8点为第一班车，晚上18点为最后一班车
按任意键重返回

图 4.8 超出时间不能乘车

图 4.9、图 4.10 表示用户乘车成功，并打印相关信息。

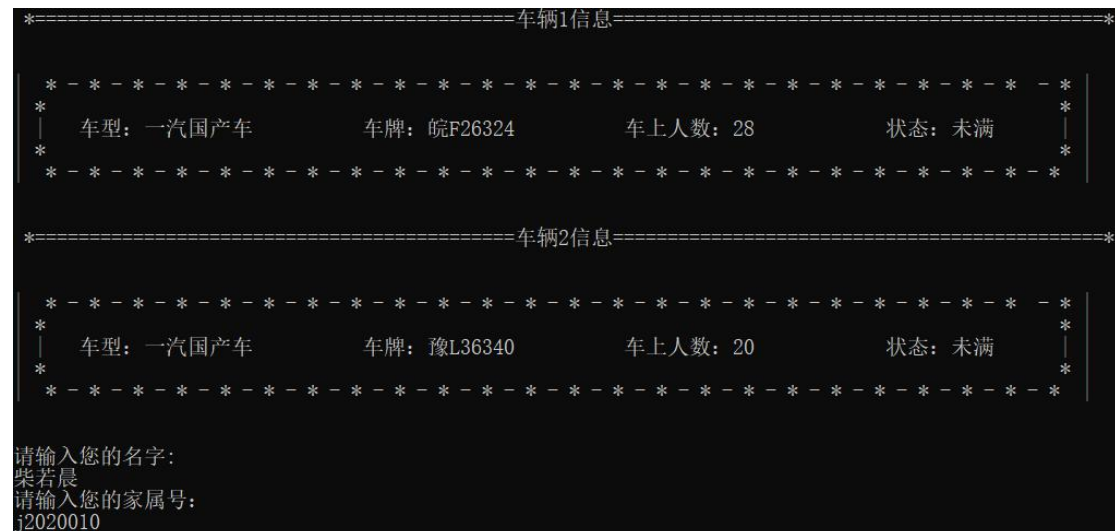


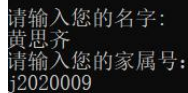
图.9 车辆信息展示及用户信息输入



前后账户金额、乘车次数变化如图 4.11 所示。

柴若晨 fj2020010 2.200000 1 xz2b4|

图 4.12、图 13 表示账户余额不足 2 元，不允许乘车。图 4.12 与图 4.9 对比车辆 1 上的人数增加 1 人。



```

=====车辆1信息=====
*
* - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - *
* |   车型：  一汽国产车           车牌：  皖F26324           车上人数： 29           状态： 未满           |
*
* - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - *
=====车辆2信息=====
*
* - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - *
* |   车型：  一汽国产车           车牌：  豫L36340           车上人数： 20           状态： 未满           |
*
* - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - *
请输入您的名字：
黄思齐
请输入您的家属号：
j2020009
账户余额不足2.0元，请充值
按任意键返回

```

图 4.14 显示车辆 1 达到 30 人满员，故只能上第二辆车。

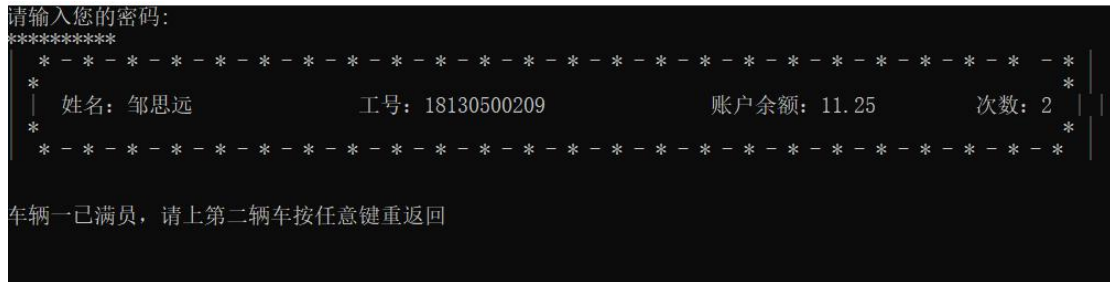


图 4.14 车辆满员情况截图

4.6 充值功能

图 4.15 表示充值过程、

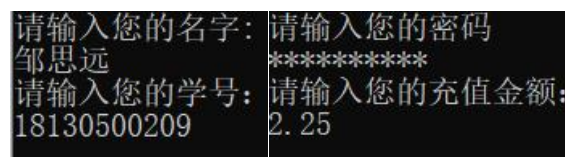


图 4.15 账户充值结果截图

图 4.16 表示充值前后账户的变化对比, 金额多了 2.25 元。

邹思远 m 18130500209 通信工程学院 siyuanzhou 1 7.000000

邹思远 m 18130500209 通信工程学院 siyuanzhou 1 9.250000

图 4.16 充值前后账户变化

图 4.17 表示不在账户中的人无法充值。

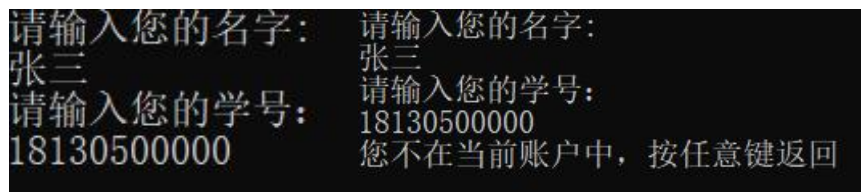


图 4.17 用户无法充值的情况截图

由于教师、学生、家属一些操作在用户层面看来区别不大, 故仅做上述演示。另有一些效果是截图无法体现出来的, 需要实际运行。

五、实验总结

5.1 面向对象的程序设计范型和封装

这次编码的过程在 main 函数外没有任何一个全局变量、完全暴露函数。主函数外只有类和方法的定义。在编程过程中, 我不断地思考: 有什么 C++ 的特性可以使用, 以此来优化代码。其中最让我印象深刻的就是函数指针数组, 简化了分支结构。这令我不得不大呼: 函数指针大法好。

由于长期使用 Python 做代码复现, coding 的思维已有些许僵化, 在声明变量, 函数返回的时候有些许不习惯。但是这次编发也让我真正认识到 C++ 的优势所在。

5.2 用户信息生成

这次编码的个性在于: 我的系统需要由一个本来就存在的信息库(如学籍信息库)提供基

础信息,以避免一个根本不是校内的人或者根本不存在的人进行注册,然后建立系统的数据。那么问题来了,这个原本就存在的一个信息库从哪里来呢?我的工作如下:

教师信息直接从教师主页上抓取用 python 进行处理,将已有的班级学生名单做分割,一部分成为学生、一部分成为家属。那么问题解决了吗?看似解决了,但是公交车的信息从哪里来呢?我通过各省简称生成公交车牌、通过常用的姓氏进行组合,生成司机名字。得到文件: Teacher_Fam_Message.txt、Student_Message.txt、Teacher_Message.txt、Bus_Message.txt。而本文设计的校园一卡通系统则存储于以下的文件中: Teacher_Fam_Account_Message.txt、Student_Account_Message.txt、Teache_Accountr_Message.txt。

因为这是模拟工程,不是真实工程,真实的用户场景下这些工作肯定是可以避免的,但是这些工作也锻炼了我的能力。

5.3 代码耦合性问题

最近在学习 Software Engineering,这里面强调了代码之间低耦合的优势便是便于调试。本文的系统是“树形”的层次结构,在调试很小的一个函数的时候,往往要退出并重新进行多次选择才能到达我想要的那个功能入口。因此单元的测试十分重要。

但是单元测试有函数之间的依赖性,这就使得单元测试也比较复杂。我在设计的后期才发现了一点,并有意地降低代码的耦合性,加速了进程。

曾经以为 Software Engineering 这些东西都是形而上学的方法论,经过这次大作业才发现,这才是所谓的“理论指导实践”、“实践与认识的相互作用”。

5.4 版本管理与设计

曾经一度将 Github 作为百度网盘和同学之间互传文件。这次实验,总计 20 余次 commits,便利了我的编码过程。以后无论是论文复现还是 oj 的刷题要多用 Github。Xmind 软件也在软件设计绘制结构图的时候起到了不小的作用。

本次作业已经在 Github 开源: <https://github.com/LinghaoChan/Cpp-Final-Homework>。