



|  |  |  |  |
| --- | --- | --- | --- |
| **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** |  |

西安电子科技大学计算机科学与技术学院

1. **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++。

1. **实验内容**

建立校园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间段内进行乘车，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。

1. **实验步骤**

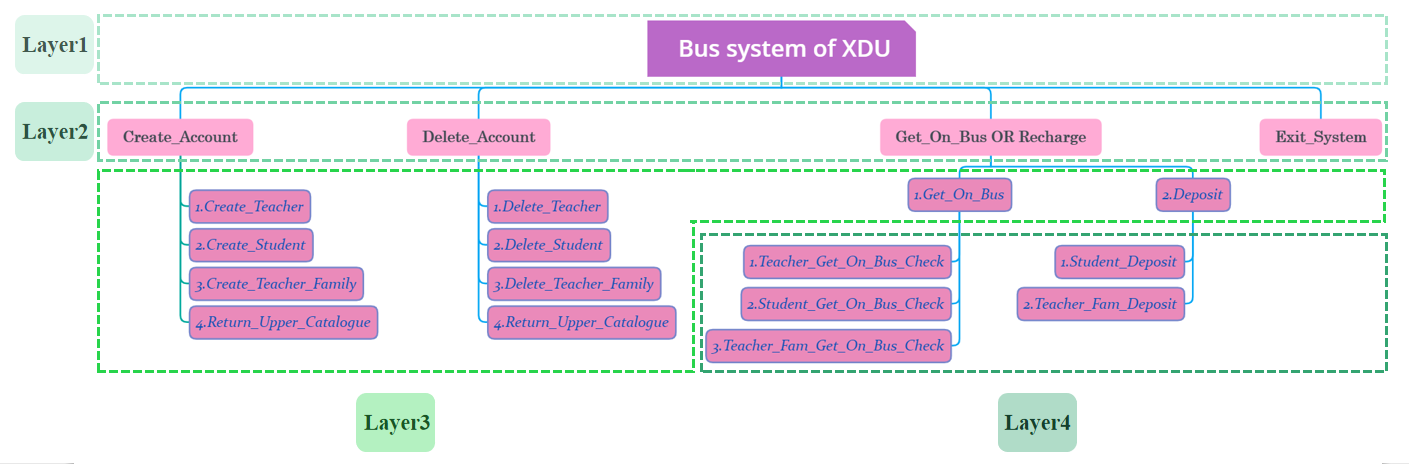
**3.1 系统设计与实验构思**

**·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**

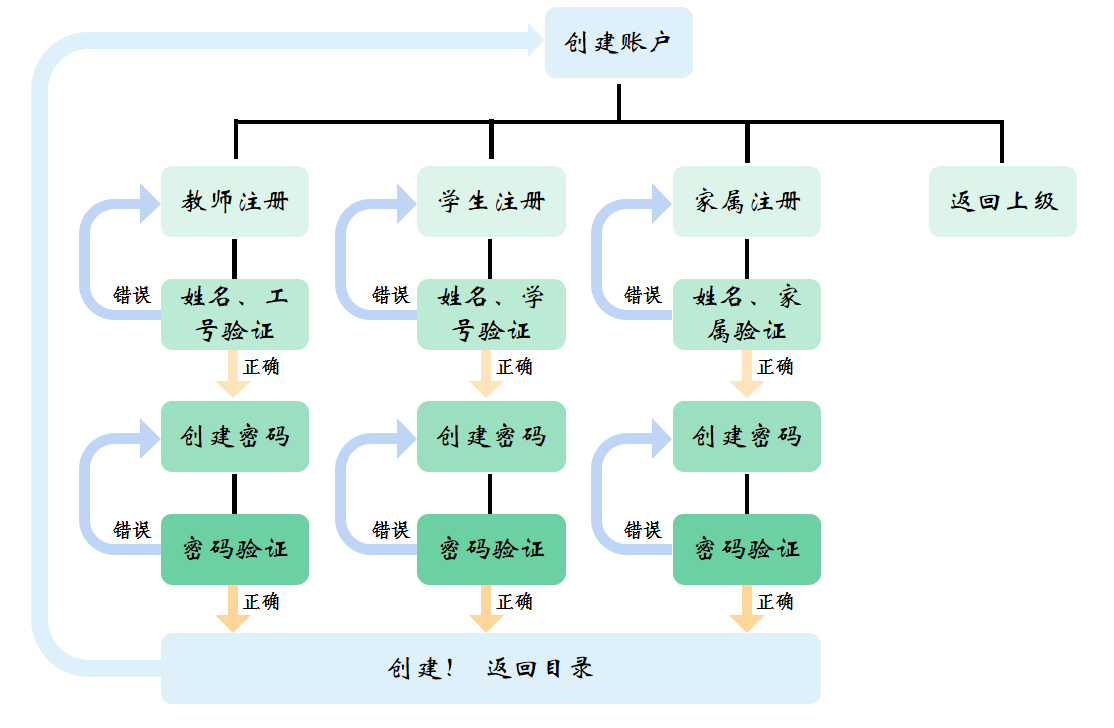
系统主要采用“树形”的层次结构进行功能分支。整体的层次结构如图一所示(**可放大查看**)。

****

**图3.1 系统功能结构示意图**

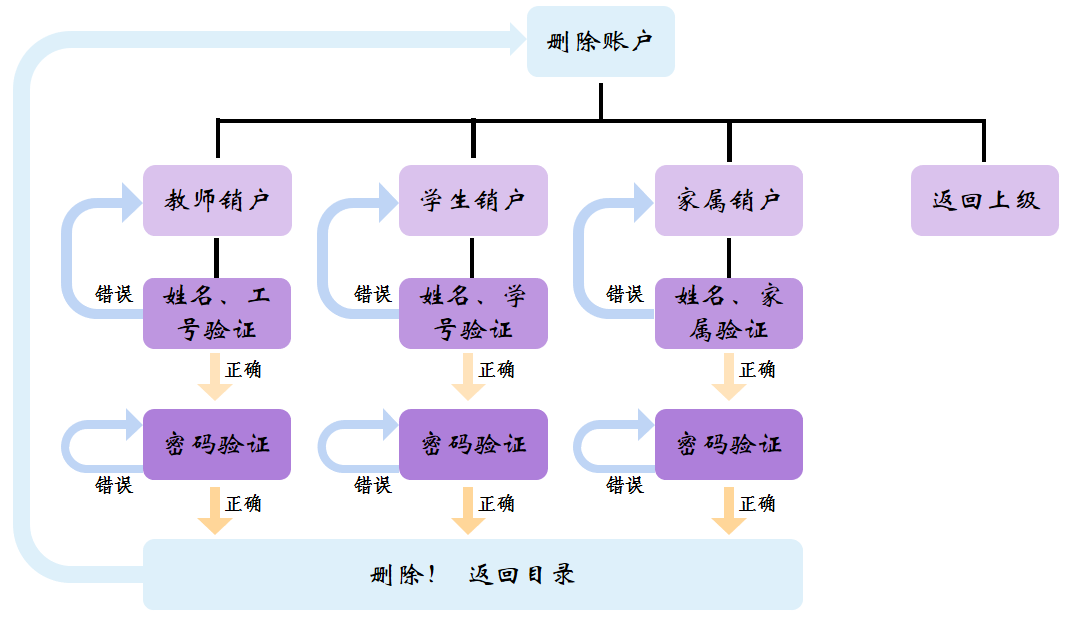
图一所示的系统分为4个大模块：创建账户、注销账户、乘车或充值、退出系统。**创建账户**涉及的功能包括：教师创建账户、学生创建账户、家属创建账户、返回上级；**注销账户**教师注销账户、学生注销账户、家属注销账户、返回上级；**乘车**涉及的功能包括：教师乘车、学生乘车、家属乘车、返回上级；**充值**涉及的功能包括：学生充值、家属乘车、返回上级。

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的文件(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。



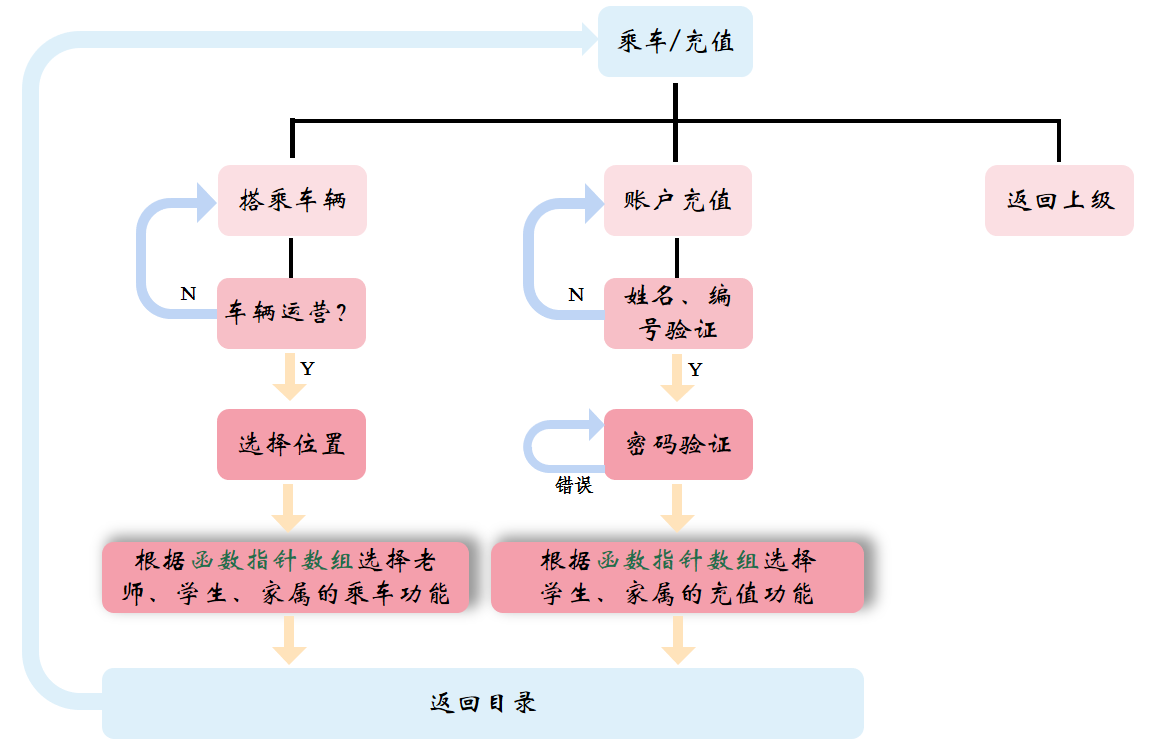
**图3.2 账户创建逻辑示意图**

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**对不同的用户进行充值。

****

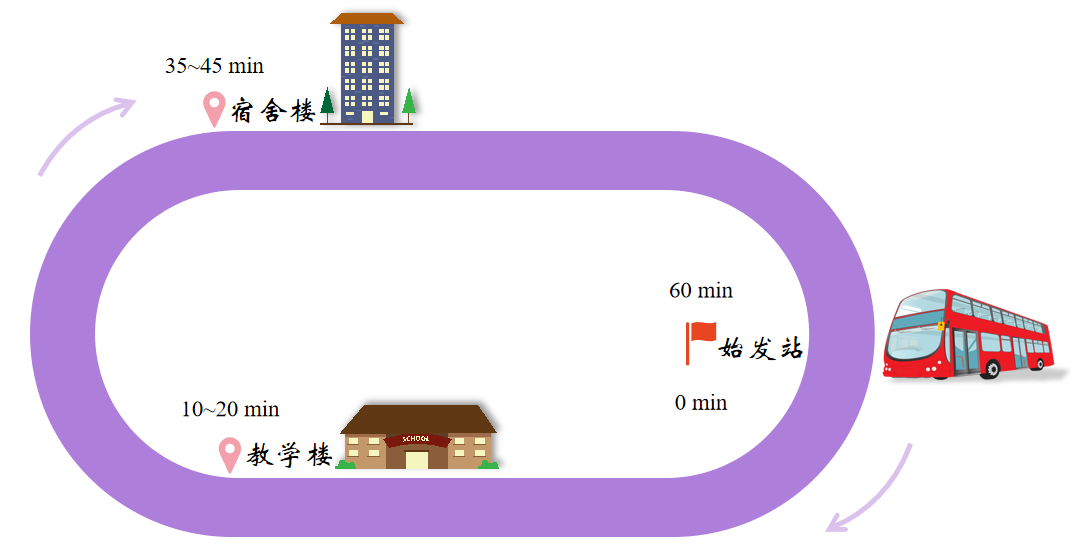
**图3.3 用户乘车或充值功能示意图**

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 |

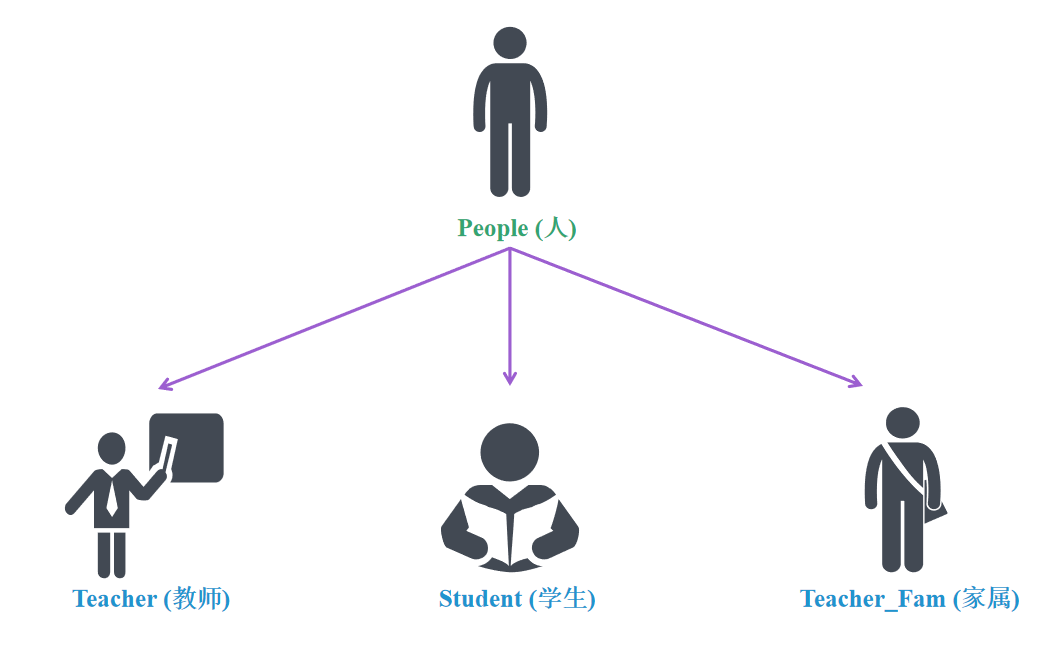
系统所涉及的算法中，Bus的构造函数较为复杂，其算法伪代码如Algorithm1中所示。由于在Bus\_Message.txt中已经存在有50辆车的基本信息。在模拟行为的时候，由于不能初始化2辆同样的车，这里需要在初始化第二辆车的构造函数参数列表中输入不合适的车辆编号(指第一个Bus对象的车辆编号)。为保证一致性，在调用的时候，第一辆车的参数取51即可。同时还需要根据系统时间得到车辆信息。

|  |
| --- |
| **Algorithm 1: Constructor of Bus** |
| **Input Parameter**: 非法车辆编号un\_permit //第1辆车取51，第2辆车区第一辆车的编号 |
| **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)： |
| **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**

**3.4 Bug改进及用户体验改善**

1. **实验结果**
2. **实验总结**
3. **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；

使用IDE：DEV C++。

1. **实验内容**

系统分为四个模块：创建账户、注销账户、乘车/充值、退出系统。

1. Sfd
2. **实验步骤**

写出实验思路、关键数据结构定义和类的接口定义、主要算法流程以及遇到的问题和解决方案。

1. **实验结果**

针对不同输入，该程序都存在哪几类可能出现的情况，你的测试数据要完全覆盖了你所想到的这些情况，并给出测试结果。

1. **实验总结**

阐述通过实验你对对本次实验的认识和理解（200字左右）。

附写作要求：

1. 独立完成，如有抄袭，以0分记；
2. 请按照实验的实际内容填写；
3. 正文格式（微软雅黑、行距固定值20磅、五号）；
4. 不要复制代码，可以用算法或者程序流程图描述；
5. 代码清单作为附件单独附上；
6. 以附件方式电子版提交至学在西电平台；
7. 写作要求在提交前请删除！