



|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | **19030500024** | **姓名** | **马子豪** |
| **班级** | **1903058** | **任课老师** | **褚华** |
| **实验名称** | **校园一卡通乘车模拟系统** | | |
| **实验学期** | **2020-2021学年第二学期** | | |
| **实验日期** | **2021.5.29** | **实验地点** | **G楼314** |
| **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

# 一、实验目的

采用面向对象程序设计范型、C++语言实现一个完整的简化系统——校园一卡通乘车模拟系统。通过编写该系统熟悉掌握面向对象的编程思想。

# 二、实验环境

* **硬件环境：**

Lenovo Legion Y7000P

处理器：Intel(R) Core(TM) i5-9300H CPU @2.40GHz

Windows 10 64位操作系统

* **集成开发环境**：

Visual Studio 2019

* **数据存储：**

MySQL Ver 8.0.24

* **代码版本管理：**

[Github](https://github.com/ZiHao256/Bus_Card_System)

# 三、实验内容

1. **系统增加对三种一卡通类型的管理**，每种一卡通所有的属性不尽相同。在一卡通持有者刷卡乘车时，根据所持有的一卡通种类的不同，对每张一卡通的属性值（余额、乘车次数等）进行相对应的更新。

2. **系统增加对班车信息的管理**，每辆班车都有各自的基本信息。根据班车的基本信息编制运行时刻表。乘客刷卡上车时，根据班车的载乘人数进行相应的操作。

3. **系统增加对一卡通持卡人信息的管理**。增加对每个人乘车的记录，和对一卡通的业务操作：申请一卡通、注销一卡通、挂失一卡通等。

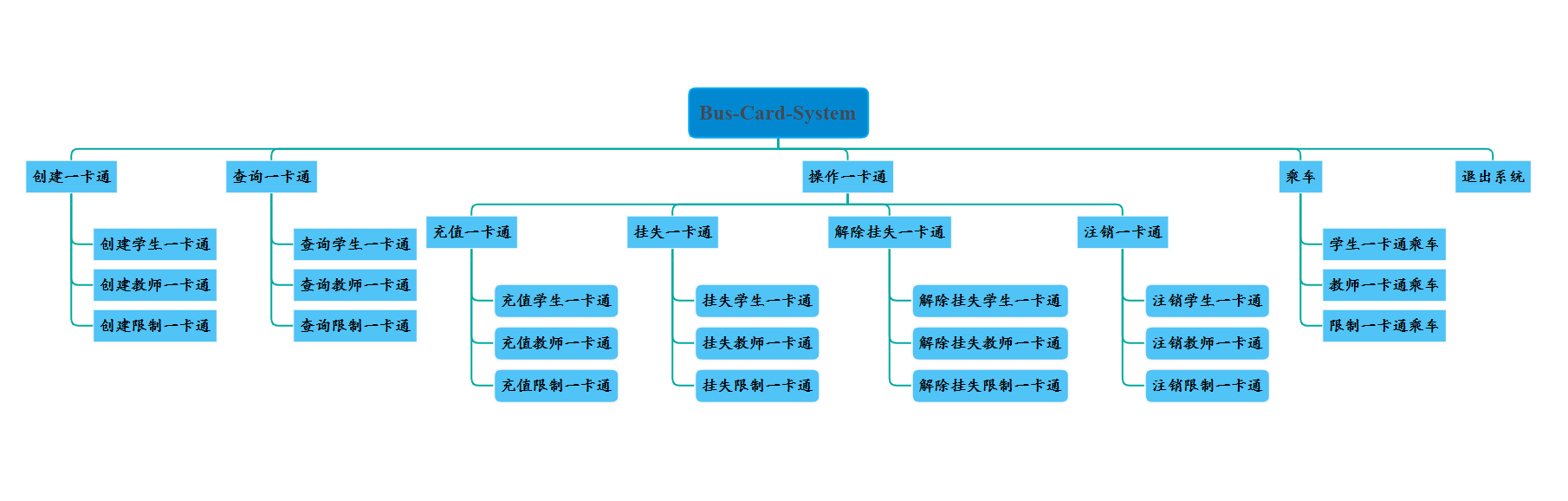
4. **从相应文件中获取信息并完成对班车的仿真**。并设置中途停车点。

# 四、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。

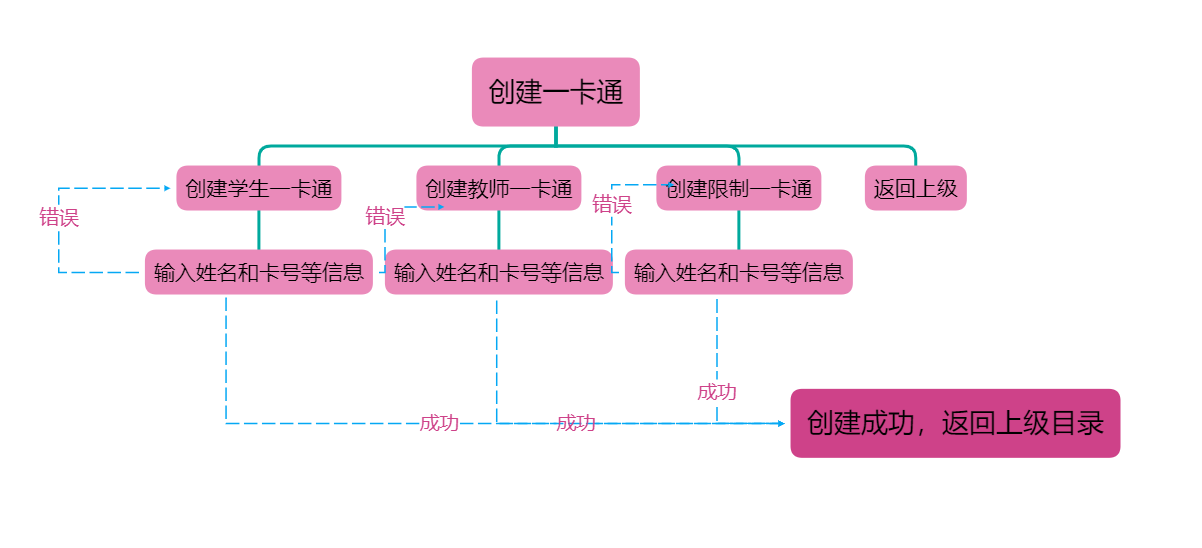
## 4.1 实验思路

### 4.1.1 系统整体结构设计

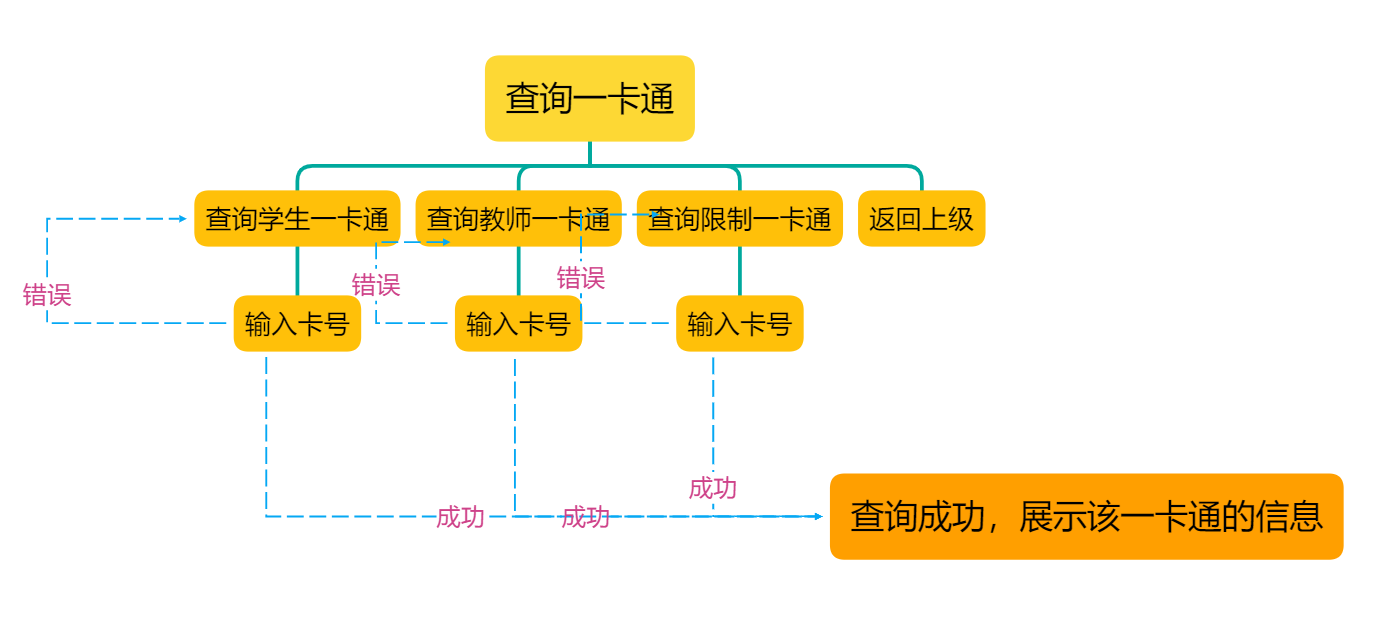


**图4.1 系统整体结构设计图**

### 4.1.2 创建一卡通功能逻辑设计

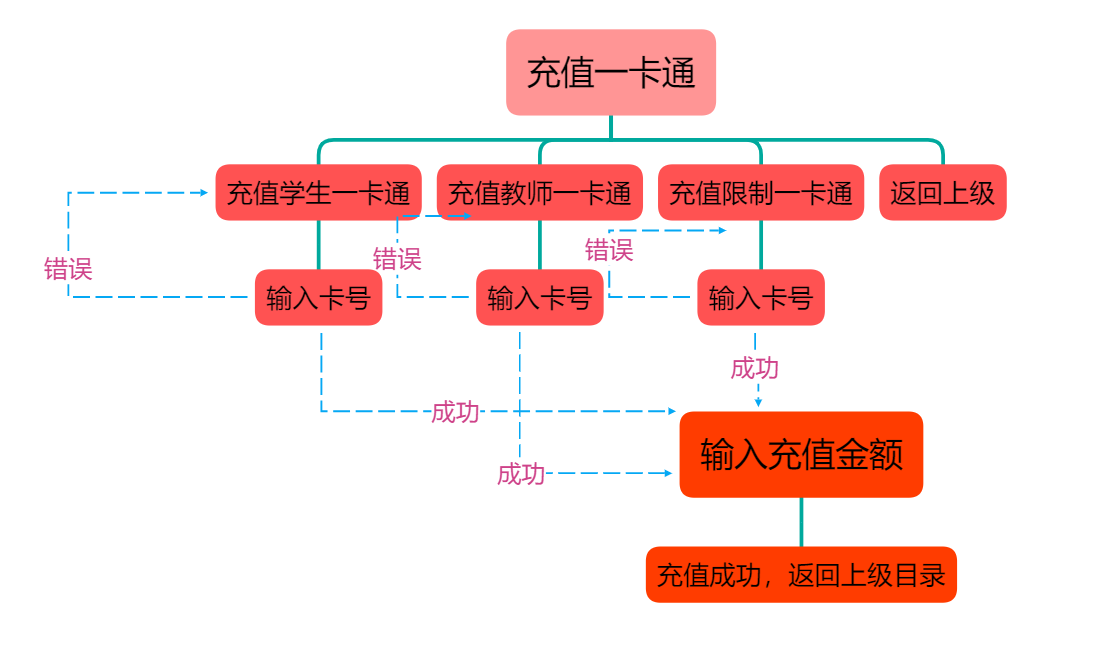


### 4.1.3 查询一卡通功能逻辑设计

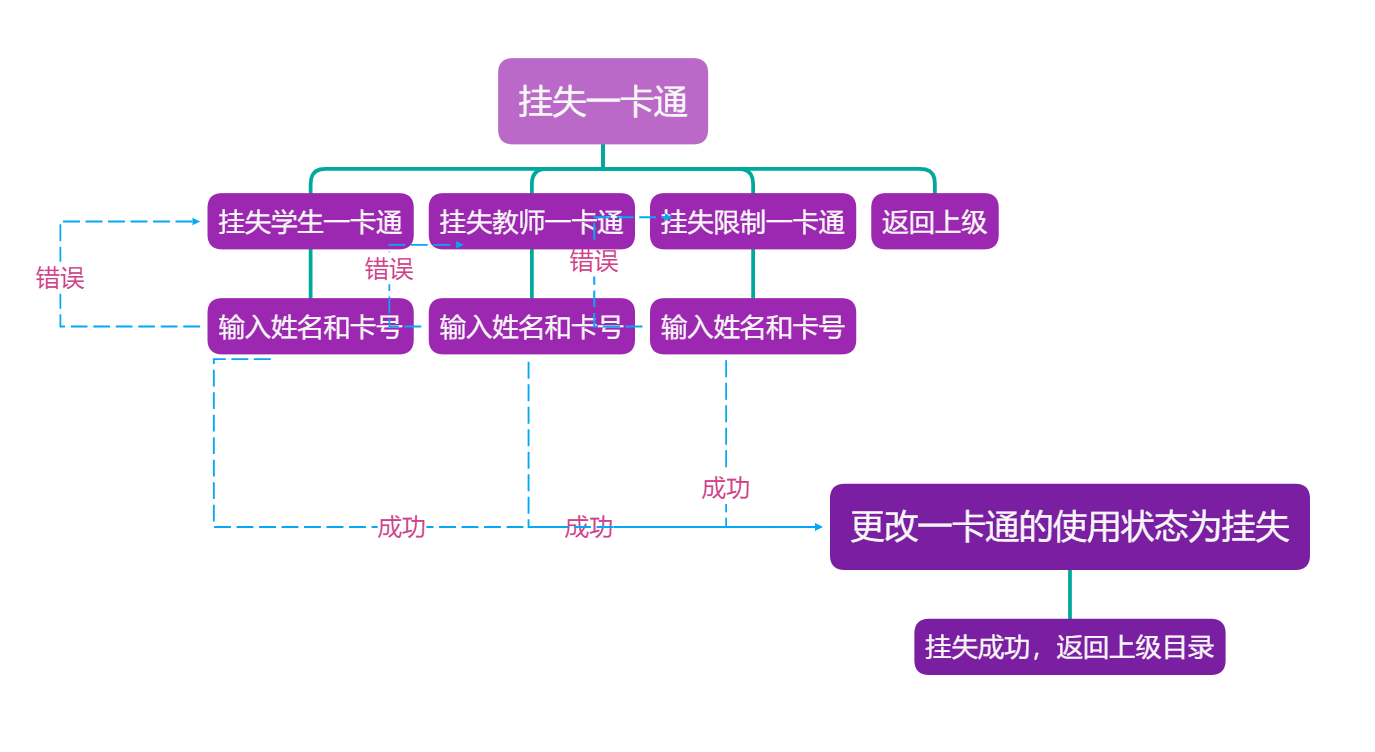


### 4.1.4 操作一卡通功能逻辑设计

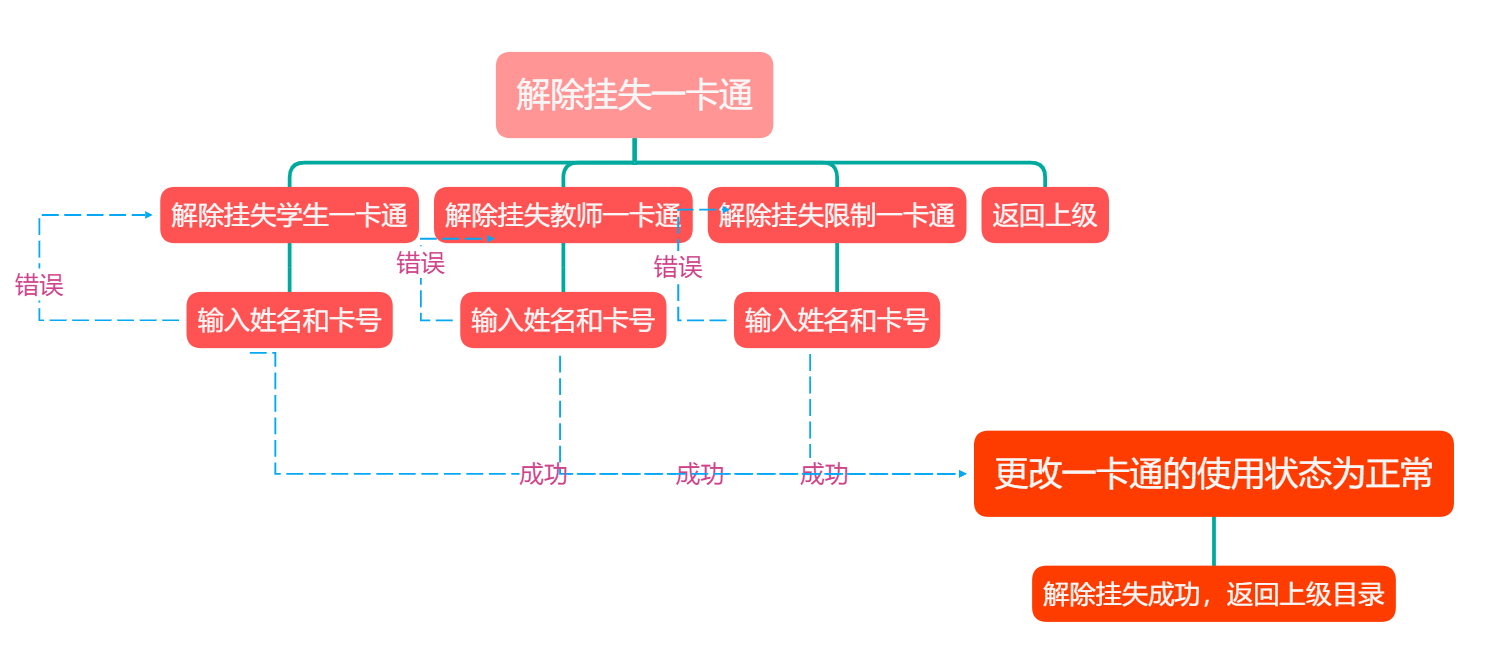
#### 4.1.4.1 充值一卡通功能逻辑设计



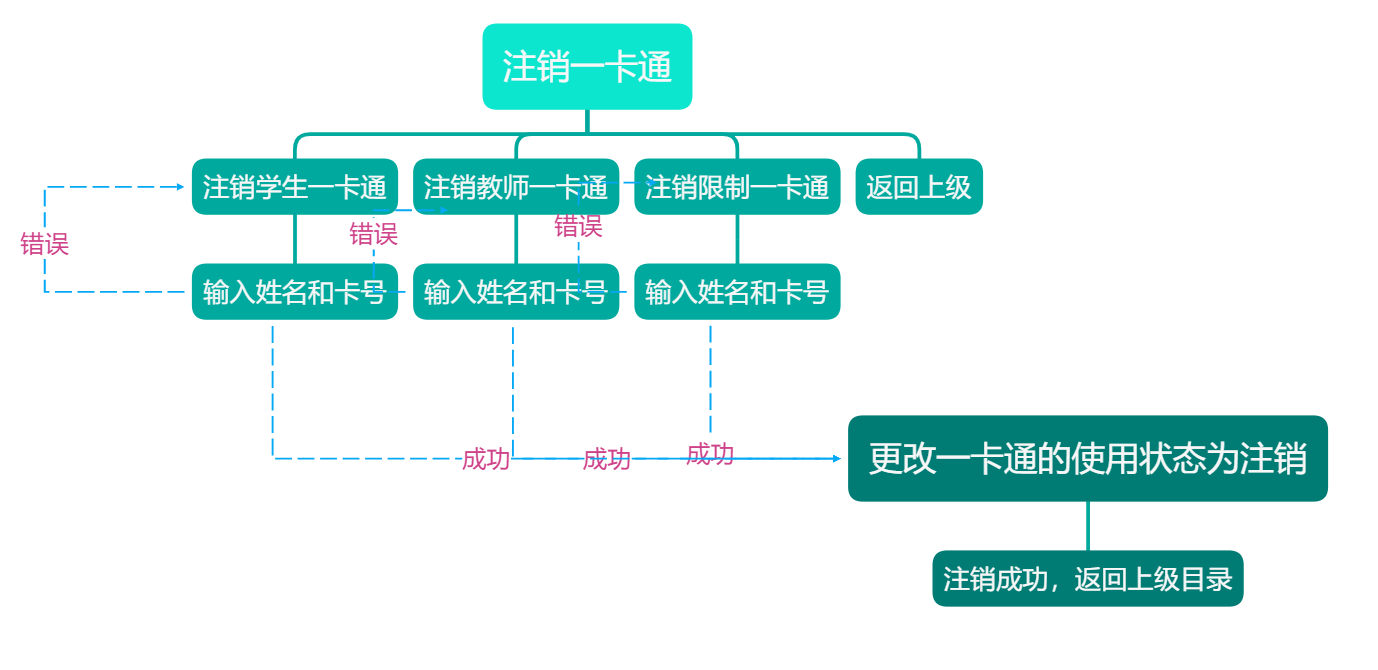
#### 4.1.4.2 挂失一卡通功能逻辑设计



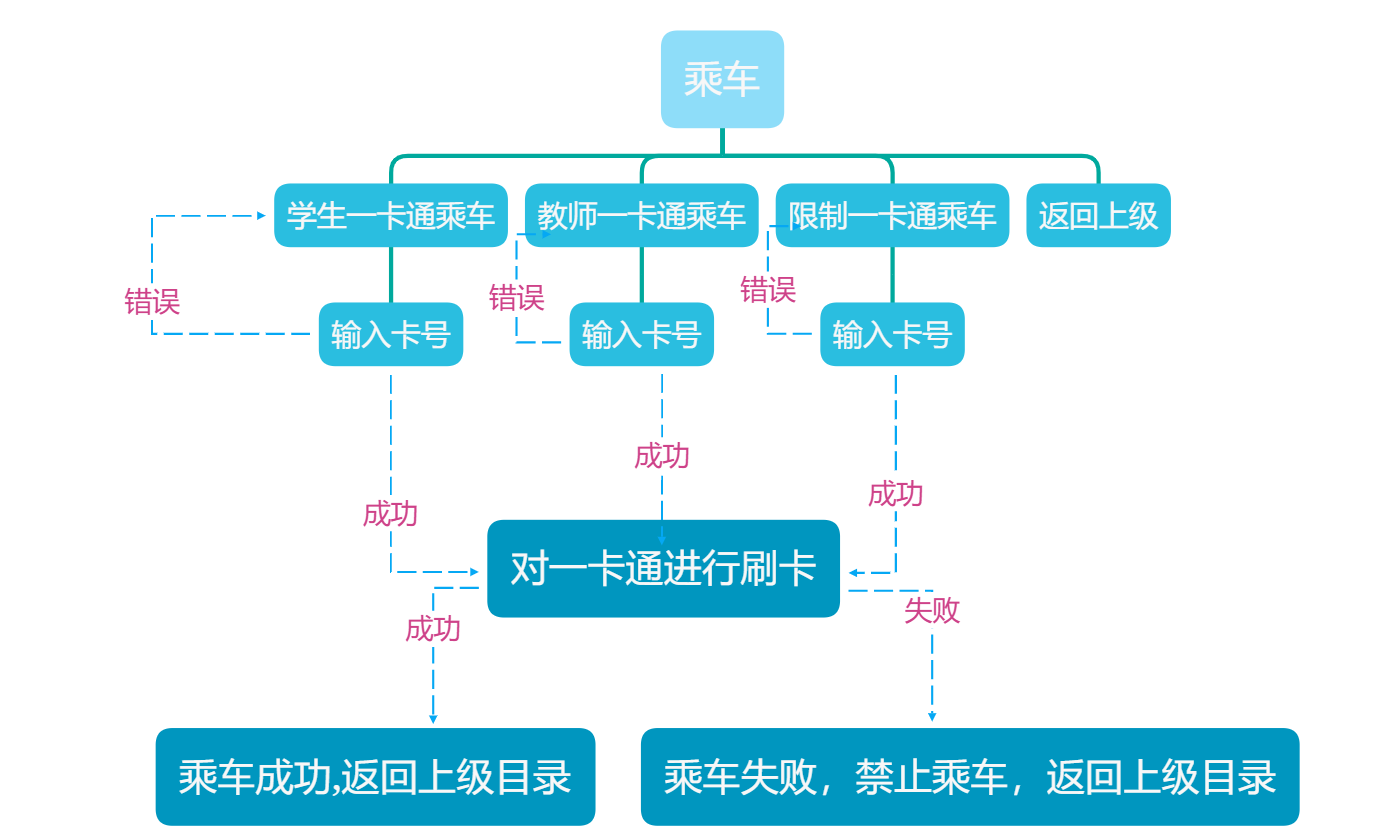
#### 4.1.4.3 解除挂失一卡通功能逻辑设计



#### 4.1.4.4 注销一卡通功能逻辑设计



### 4.1.5 乘车功能逻辑设计



## 4.2 关键数据结构的定义

该校园一卡通乘车模拟系统编写了五个类来实现。

### 4.2.1 class Student 学生一卡通

Student类有8个数据成员和8个成员函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 名称 | 类型 | 含义 |
| private  数据成员 | status | int | 一卡通状态 |
| name | string | 持卡人姓名 |
| number | string | 卡号 |
| gender | string | 持卡人性别 |
| times | int | 乘车次数 |
| remain | int | 一卡通余额 |
| subordinate | string | 持卡人所属单位 |
| date | string | 一卡通有效期 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 名称 | 参数列表 | 返回值 | 功能 |
| public  成员函数 | Student | (int,string,string,string,  int,int,string,string) | — | 构造函数 |
| ~Student | (void) | — | 析构函数 |
| show\_info | (void) | void | 展示一卡通信息 |
| get\_on | (char[ ]) | void | 使用一卡通乘车 |
| get\_off | (char[ ]) | void | 下车 |
| log\_out | (void) | void | 一卡通注销 |
| report\_loss | (void) | void | 一卡通挂失 |
| unlock\_report\_loss | (void) | void | 一卡通解除挂失 |

### 4.2.2 class Teacher 教师一卡通

Teacher 类有7个数据成员和8个成员函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 名称 | 类型 | 含义 |
| private  数据成员 | status | int | 一卡通状态 |
| name | string | 持卡人姓名 |
| number | string | 卡号 |
| gender | string | 持卡人性别 |
| times | int | 乘车次数 |
| subordinate | string | 持卡人所属单位 |
| date | string | 一卡通有效期 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 名称 | 参数列表 | 返回值 | 功能 |
| public  成员函数 | Teacher | (int,string,string,string,  int,string,string) | — | 构造函数 |
| ~Teacher | (void) | — | 析构函数 |
| show\_info | (void) | void | 展示一卡通信息 |
| get\_on | (char[ ]) | void | 使用一卡通乘车 |
| get\_off | (char[ ]) | void | 下车 |
| log\_out | (void) | void | 一卡通注销 |
| report\_loss | (void) | void | 一卡通挂失 |
| unlock\_report\_loss | (void) | void | 一卡通解除挂失 |

### 4.2.3 class Limit 限制一卡通

Limit 类有9个数据成员和8个成员函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 名称 | 类型 | 含义 |
| private  数据成员 | status | int | 一卡通状态 |
| name | string | 持卡人姓名 |
| number | string | 卡号 |
| gender | string | 持卡人性别 |
| times | int | 乘车次数 |
| remain | int | 余额 |
| free\_times | int | 剩余免费次数 |
| subordinate | string | 持卡人所属单位 |
| date | string | 一卡通有效期 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 名称 | 参数列表 | 返回值 | 功能 |
| public  成员函数 | Limit | (int,string,string,string,  int,int,int,string,string) | — | 构造函数 |
| ~Limit | (void) | — | 析构函数 |
| show\_info | (void) | void | 展示一卡通信息 |
| get\_on | (char[ ]) | void | 使用一卡通乘车 |
| get\_off | (char[ ]) | void | 下车 |
| log\_out | (void) | void | 一卡通注销 |
| report\_loss | (void) | void | 一卡通挂失 |
| unlock\_report\_loss | (void) | void | 一卡通解除挂失 |

### 4.2.4 class Bus 班车

Bus 类有6个成员函数和9个成员函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 名称 | 类型 | 含义 |
| private  数据成员 | number | string | 车牌号 |
| type | string | 车类型 |
| driver | string | 司机名字 |
| max\_num | int | 最大乘客数 |
| cur\_num | int | 现有乘客数 |
| start\_time | int | 发车时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 名称 | 参数列表 | 返回值 | 功能 |
| public  成员函数 | Bus | (string,string,string,int,int,int) | — | 构造函数 |
| ~Bus | (void) | — | 析构函数 |
| show\_bus | (void) | void | 展示该车的信息 |
| get\_cur\_num | (void) | int | 获得当前该车上乘客数 |
| check\_bus | (void) | void | 乘客乘车 |
| add\_people | (string,string) | void | 增加乘客 |
| delete\_people | (string,string) | void | 删除乘客 |
| bus\_detail | (void) | void | 展示该车上当前乘客具体信息 |
| clear\_bus | (void) | void | 清空该车 |

### 4.2.5 class System 系统

System 类较为特殊，无数据成员，只有成员函数。由于该系统中并未对System类进行实例化，因此仅有静态成员函数。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 名称 | 参数列表 | 返回值 | 功能 |
| public  成员函数 | start | (void) | static void | 进入系统主页 |
| moni | (void) | static void | 模拟班车运行 |
| get\_student | (char[]) | static Student | 获得一个Student对象 |
| get\_teacher | (char[]) | static Teacher | 获得一个Teacher对象 |
| get\_limit | (char[]) | static Limit | 获得一个Limit对象 |
| **create\_card** | (void) | static void | 进入创建一卡通页面 |
| create\_student | (void) | static void | 创建学生一卡通 |
| create\_teacher | (void) | static void | 创建教师一卡通 |
| create\_limit | (void) | static void | 创建限制一卡通 |
| **show\_card** | (void) | static void | 进入展示一卡通页面 |
| show\_student\_card | (void) | static void | 展示学生一卡通 |
| show\_teacher\_card | (void) | static void | 展示教师一卡通 |
| show\_limit\_card | (void) | static void | 展示限制一卡通 |
| **get\_on** | (void) | static void | 进入乘车页面 |
| get\_on\_student | (char[],char[]) | static void | 学生一卡通乘车 |
| get\_on\_teacher | (char[],char[]) | static void | 教师一卡通乘车 |
| get\_on\_limit | (char[],char[]) | static void | 限制一卡通乘车 |
| **operate\_card** | (void) | static void | 进入操作一卡通页面 |
| **log\_out** | (void) | static void | 进入注销一卡通页面 |
| log\_out\_student | (void) | static void | 学生一卡通注销 |
| log\_out\_teacher | (void) | static void | 教师一卡通注销 |
| log\_out\_limit | (void) | static void | 限制一卡通注销 |
| **report\_loss** | (void) | static void | 进入挂失一卡通页面 |
| report\_loss\_student | (void) | static void | 学生一卡通挂失 |
| report\_loss\_teacher | (void) | static void | 教师一卡通挂失 |
| report\_loss\_limit | (void) | static void | 限制一卡通挂失 |
| **unlock\_report\_loss** | (void) | static void | 进入解除挂失一卡通页 |
| unlock\_student | (void) | static void | 学生一卡通解除挂失 |
| unlock\_teacher | (void) | static void | 教师一卡通解除挂失 |
| unlock\_limit | (void) | static void | 限制一卡通解除挂失 |
| get\_bus | (char[]) | static Bus | 获得一个Bus对象 |
| add\_bus | (void) | static void | 添加一个班车 |

## 4.3 类的接口定义

## 4.4 关键算法流程

### 4.4.1 class Student

类Student中，涉及的关键算法是对实例的析构函数。

* **~Student 算法：**



* **get\_on 算法：**

****

* **top\_up 算法**

### 4.4.2 class Teacher

* ~Teacher

和Student类的析构函数类似。

* get\_on

和Student类的get\_on函数相似

### 4.4.3 class Limit

* ~Limit:

和Student类的析构函数相似

* get\_on

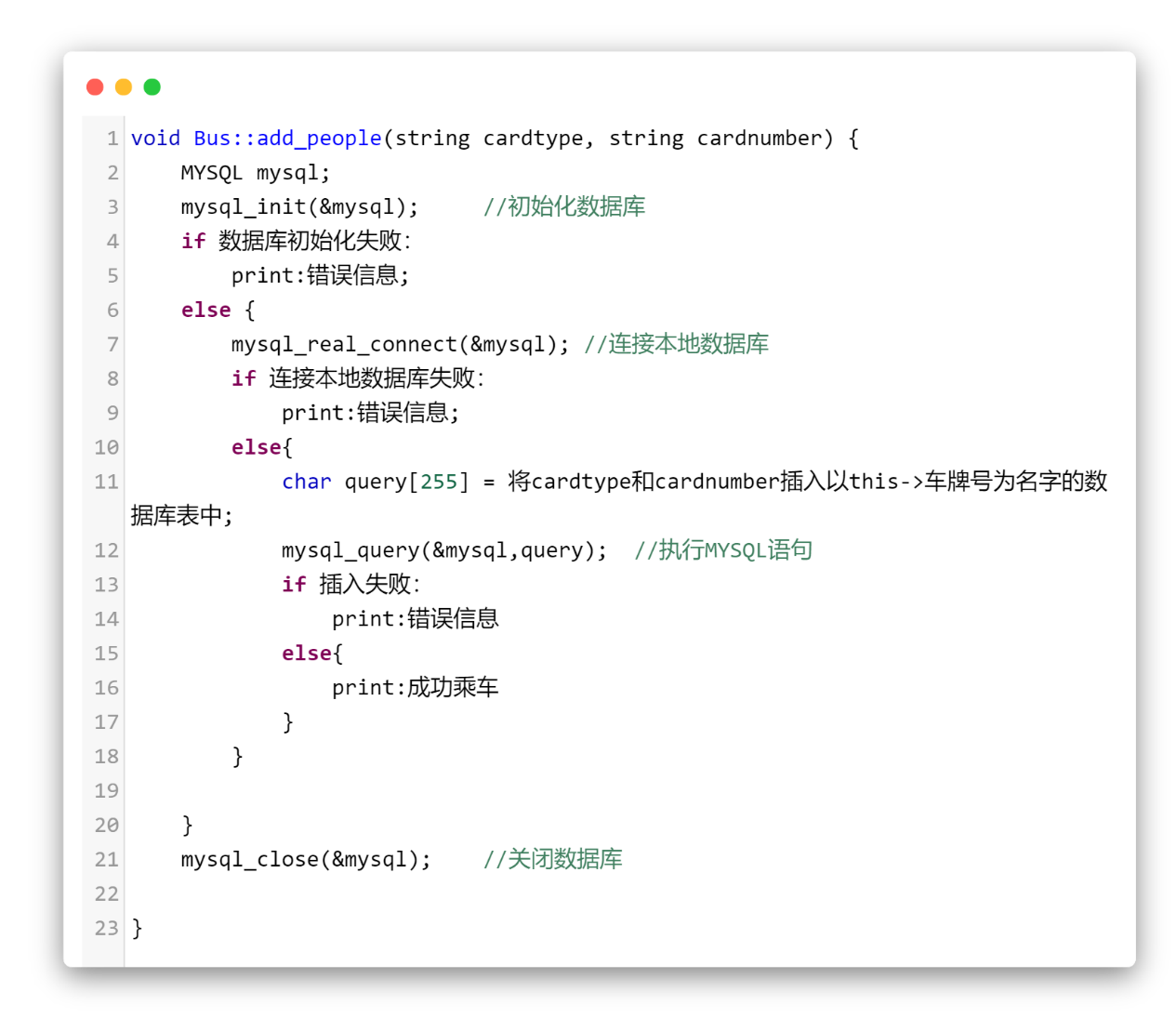
和Student类的get\_on函数相似

### 4.4.4 class Bus

* get\_cur\_num



* add\_people



* bus\_detail



### 4.4.5 class System

* get\_setudent



* get\_teacher

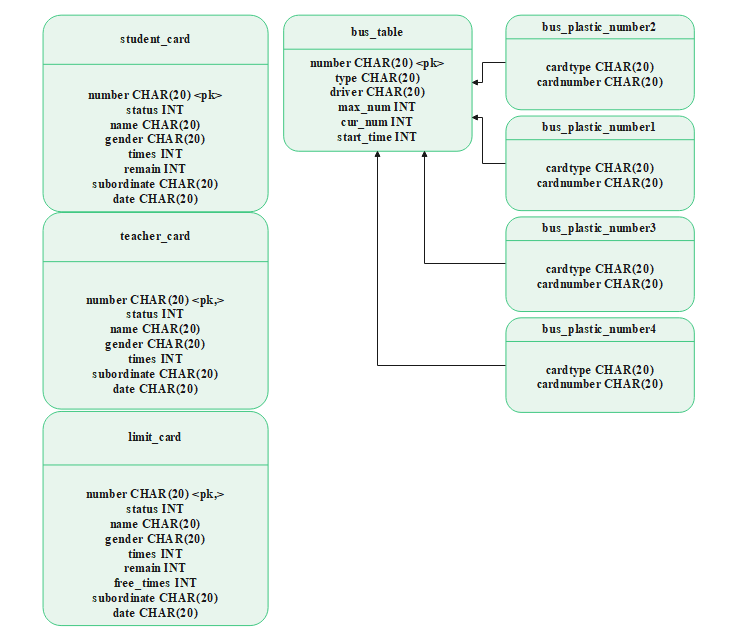
和get\_student类似

* get\_limit

和get\_student类似

## 4.5 数据存储

使用Visual Studio 2019连接本地MySQL数据库，来存储学生一卡通、教师一卡通、限制一卡通、班车信息以及每辆班车当前所载的乘客信息。



## 4.5 遇到的问题

# 五、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。

# 六、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. 正文格式（微软雅黑、行距固定值20磅、五号）；
3. 不要复制代码，可以用算法或者程序流程图描述；
4. 写作要求在提交前请删除！