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	20009200713	**姓名**	**曾凡浩**
班级	**2003052**	**任课教师**	**张淑平**
实验名称	一卡通乘车系统		
实验学期	**2021 – 2022 学年第 2 学期**		
实验日期	2022 年 6 月 13 日	**实验地点**	
报告成绩			

西安电子科技大学计算机科学与技术学院

1、 实验目的

本次实验，练习使用 C++ 例程库中提供的典型例程和通过命令行参数向程序 main()传递来自程序外部的数据，以便学生了解并掌握例程库中的最基本要素和命令行参数的基本用法。

2、 实验环境

操作系统：[Windows11]

开发工具：[Visual studio 2022]

3、 实验内容

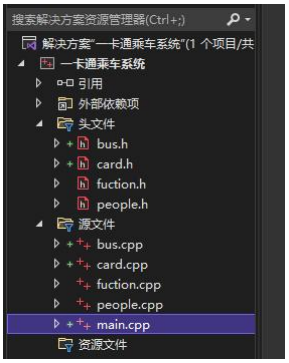
（1）完成一个完整的简化系统：校园一卡通乘车模拟系统。
（2）在实现该系统的功能时，不要求实现图形界面（有余力的同学可以考虑），控制台输出所要求的信息即可。
（3）必须采用面向对象程序设计范型、C++语言实现。
校园一卡通乘车模拟系统主要用于管理校园班车和一卡通信息，维护乘车过程中一卡通信息的动态变化。
主要功能要求如下：
1. 校园中的人员分为三类：教工、学生、家属。按照人员身份，将一卡通分为教师卡（教工持有）、学生卡（学生持有）和家属卡（家属持有）。 每人只
能持有一张一卡通。系统中一卡通的数量关系为：学生卡>教工卡>家属卡。
2. 使用一卡通之前需要向系统申请开卡。开卡时需提供持卡人的姓名、性别、身份（教工、学生、家属）、所属学院或部门（家属不需要提供此项信息）、工资号/学号（教工提供工资号，学生提供学号，家属不需要提供此项信息）。教
工卡由工资号进行区分，学生卡由学号进行区分；家属卡在开卡时由系统生成一个系统唯一的卡号。
3．一卡通可以充值。
4. 一卡通不再使用时，需向系统申请注销。注销之后的一卡通不能乘坐班车。
5. 有两种型号的班车，大客车载乘人数为 50 人（不含司机），小客车载乘人数为 30 人（不含司机）。班车的基本信息包括：车牌号、型号、载乘人数、驾驶员姓名。
6. 使用一卡通乘车时，遵循以下规则：使用教工卡，免费乘车；使用学生卡， 每次乘车扣费 2 元；使用家属卡乘车，每月的前 20 次免费（免费次数当月有效，不累计），超出之后每次扣费 2 元。
7. 需要模拟的行为有：
- 班车发车时间间隔为 1 个小时，每次发 2 辆车，坐满即开；站内停留时间 10 分钟。早上 8 点为第一班车，晚上 18 点为最后一班车。为简单

起见，班车只考虑单向运行，到点班车发出即可。

- 随机产生每个小时的班车种类和乘车人数。
- 显示当前要发车的班车基本信息。
- 刷卡上车时，统计已经上车的人数。如果达到该班车的载乘人数，则禁止继续上车。
- 刷卡乘车时，应显示出每张卡片的持有人姓名及卡内余额。对于家属卡，还应显示出已经乘坐的次数；当达到 20 次时，提示这是本月最后一次免 费乘车；第 21 次时，提示本次开始收费。
- 如果刷卡后卡内余额小于等于 5 元，应提示持卡人“余额过低，及时充值”；如果卡内余额少于 2 元，提示“余额不足”，不准乘车。
- 能够显示出每张一卡通每月的累计乘车次数。

4、 数据结构与算法设计

4.1 模块划分与文件组织



模块 1：主控模块，仅包括文件 main.cpp ，定义了 main()函数。

模块 2：用户模块，包括以下两个文件：

people.h　　定义了用户类和用户信息管理类，以及操作接口的声明；

people.cpp　　实现了三种卡的新建和删除等操作。

模块 3：三种卡操作模块，包括以下两个文件：

card.h　　定义了 Card 类，对应的教师卡、学生卡、限制卡三个子类和对应的三个信息管理类，以及操作接口的声明；

card.cpp　　实现了三种卡的充值、乘车和使用情况查询等操作。

模块 4：班车操作模块，包括以下两个文件：

bus.h　　定义了班车类和班车信息管理类，以及操作接口的声明；

bus.cpp　　实现了新建班车、乘车、信息汇总等操作。

模块 5：function 模块，包括以下两个文件：

function.h　　声明了系统操作的一些函数；

function.cpp　　实现了界面展示、刷卡、充值、新建和注销卡的操作。

4.2 模块的设计

4.2.1 关键数据结构和类型的设计

主要设计了两种类：对象类和对象的信息管理类

像客户的 people 类和 people_data 类

卡的部分设计了 Card 类，teacher、student、limited 三个子类；针对读文件写文件又设计了对应的 data 类

班车也是设计了 bus 类记录班车的乘车情况、发车情况等；针对读写文件设计了对应的 data 类

像新建、充值、刷卡、注销等功能，单独地放在了 function.h 和 function.cpp 里来用函数实现

4.2.2 关键算法（函数）的设计

算法 1.1　int main()

作　用：主控函数，也实现对题目所需其他内容的测试。

参　数：无参数。

返回值：总是返回 0。

计算过程：

用了一个 switch case 来实现功能的选择

```
while (choice!=0) {
        Show_Menu();
        cout << "请输入您的选择：" << endl;
        cin >> choice ;
        switch (choice)
        {
        case 0://退出系统
            ExitSystem();
            break;
        case 1:
        { build();//新增卡
        break; }
        case 2:
        { Delete();//删除卡
        break; }
        case 3:
        { Rechange();//充值卡
        break; }
        case 4:
        { use();//刷卡
        break; }
        case 5:
        { bus.buildbus();//新增班车
        break; }
        case 6:
        { bus.show();//班车信息
        break; }
        case 7:
        { people p;
        p.show();//人信息
        break; }
        default:
            cout << "输入错误！请重新输入" << endl << endl << endl;
```

算法 1.2 班车的实现

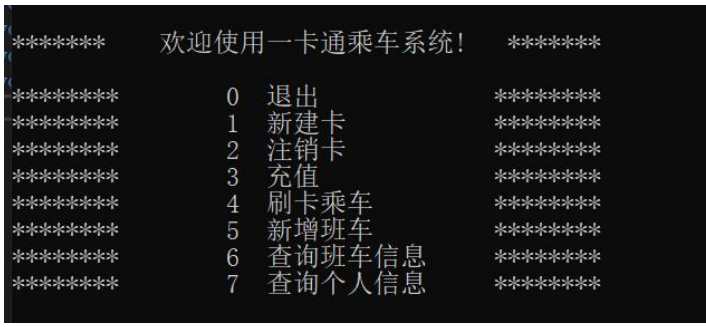
```
class Bus {
private:
    int bus_number;//车牌号
    char model[20];//车型号
    int limiters;//核载乘人数
    char driver_name[20];//司机姓名
    float startime[N][N + 1];//发车时间
    float endtime[N][N + 1];//收车时间
    char time_section[20];//运行时间段
    float r_startime[N][N + 1];//实际发车时间
    float r_endtime[N][N + 1];//实际收车时间
    int passengers[N];//实际载乘人数
    int now_passengers[N];//当前人数
    int station[N];//当前站
public:
    Bus() {}
    //set,get函数
    int* get_station() { return station; }
    int get_num() { return bus_number; }
    char* get_model() { return model; }
    int get_limiters() { return limiters; }
    char* get_name() { return driver_name; }
    char* get_section() { return time_section; }
    int* get_passengers() { return passengers; }
    int* get_now_passengers() { return now_passengers; }
    void* set_station(int a[]) {
        for (int j = 0; j < N; j++)station[j] = a[j];
        return station;
    }
    void set_num(int a) { bus_number = a; }
    void set_model(char a[]) { strcpy_s(model, a); }
    void set_limiters(int a) { limiters = a; }
    void set_name(char a[]) { strcpy_s(driver_name, a); }
    void set_startime(float a[][N + 1])
    {
        for (int j = 0; j < N; j++)
            for (int k = 0; k < N + 1; k++)
                startime[j][k] = a[j][k];
    }
    void set_endtime(float a[][N + 1]) { for (int j = 0; j < N; j++) for (int k = 0; k <
N + 1; k++) endtime[j][k] = a[j][k]; }
    void set_section(char a[]) { strcpy_s(time_section, a); }
```

```
    void set_passengers(int a[]) { for (int j = 0; j < N; j++) passengers[j] = a[j]; }
    void set_now_passengers(int a[]) { for (int j = 0; j < N; j++) now_passengers[j] = a[j]; }
    void set_r_startime(float a[][N + 1]) { for (int j = 0; j < N; j++) for (int k = 0; k
< N + 1; k++) r_startime[j][k] = a[j][k]; }
    void set_r_endtime(float a[][N + 1]) { for (int j = 0; j < N; j++) for (int k = 0; k
< N + 1; k++) r_endtime[j][k] = a[j][k]; }
    void buildbus();//新增班车
    void show();//显示班车信息
    void ontime_rate();//计算准时率
    ~Bus() {}
};//班车类

class bus_data {
private:
    Bus bus[M];//班车类对象数组
    int i;//数组下标（对象个数）
public:
    bus_data();//构造函数，读取文件信息，存入数组
    Bus* get_bus() { return bus; }//返回对象数组地址
    int search_bus();//寻找班车，返回对象数组下标
    int select_bus(int j, int k, int s);//上车，传入卡类对象数组下标，车类对象数组下标，
以及下车人数
    friend void Bus::buildbus();
    friend void Bus::show();
    ~bus_data();//析构函数，作用结束时重新写入文件
};//班车信息管理类
```

5、 测试用例与测试结果

- 初始界面下选择所需要的功能：



- 测试新建卡的功能：

  - 新建教师卡

```
新建一卡通，请录入信息：
请输入卡号：111
请输入姓名：张三
请输入工资号/学号：123456
请输入单位：XDU
请输入有效期：2023
请输入性别：男
请输入职务：教授
创建成功！
```

  - 新建学生卡

```
新建一卡通，请录入信息：
请输入卡号：222
请输入姓名：曾凡浩
请输入工资号/学号：9200713
请输入单位：xdu
请输入有效期：2023
请输入充值钱数/元：100
创建成功！
```

  - 新建限制卡

```
新建一卡通，请录入信息：
请输入卡号：333
请输入姓名：李四
请输入工资号/学号：12345
请输入单位：xdu
请输入有效期：2023
请输入充值钱数/元：50
请输入性别：男
请输入职务：家属
创建成功！
```

- 注销卡功能测试

  （教师卡，学生卡，限制卡分别注销）

```
请输入卡号：222
注销成功！
```

- 充值功能测试

```
请输入卡号：222

请输入充值钱数/元：100
充值成功！
```

- 新增班车功能测试

```
请输入您的选择：
5
新增车辆，请录入信息：
请输入车牌号：1
请输入车型号：a
请输入司机姓名：wzc
请输入载乘人数：2
请输入运行时间段：12：00-13：00
请输入第1班次预计发车时间：1200
请输入第1班次预计到达时间：1220
请输入第2班次预计发车时间：1230
请输入第2班次预计到达时间：1250
新增班车成功！
```

- 查询班车功能测试

```
请输入车牌号：1
班车信息：
此班车的预定运行时刻表为：

————————————————————————————
                班次1
————————————————————————————
运行时间段      |    7：00——11:00
————————————————————————————
发车时间        |    7
————————————————————————————
停车点1到达时间 |    7.7
————————————————————————————
停车点1离开时间 |    7.7
————————————————————————————
停车点2到达时间 |    8.3
————————————————————————————
停车点2离开时间 |    8.3
————————————————————————————
收车时间        |    9
————————————————————————————
全程运行时间    |    2
————————————————————————————
```

```
————————————————————————————
                班次2
————————————————————————————
运行时间段      |    7：00——11:00
————————————————————————————
发车时间        |    9
————————————————————————————
停车点1到达时间 |    9.7
————————————————————————————
停车点1离开时间 |    9.7
————————————————————————————
停车点2到达时间 |    10.3
————————————————————————————
停车点2离开时间 |    10.3
————————————————————————————
收车时间        |    11
————————————————————————————
全程运行时间    |    2
————————————————————————————
车牌号：  1
车型号：  car
司机  ：  siji
```

- 乘车模拟功能测试

- 个人信息查询功能测试

6、 实验总结

本次实验第一次尝试用 c++语言完成一个项目工程，对于一个工程来说所需要考虑的函数安排和变量的设定是之前那些小实验不可比拟的，不像之前的实验一样依托单一的 cpp，或者仅仅划分类来完成。之前的实验中我们只需考虑变量和函数满足当前片段即可，但是这次的一卡通实验需要我们对于不同类的函数，变量进行相互配合。在实验中也遇到了很多问题有的时候对于整形数据的使用常常习惯于依赖 int，但这恰恰有时候就会出现问题，比如当我们在进行开卡的时候我的学号是 20009200713，这样的整形数字肯定是不能用 int 来表示的，一旦我们输入这样的大数字就会溢出导致整个程序的崩溃。所以这也让我懂得了不是什么时候一提及到整形变量就想当然的使用 int，这是本次实验的一个小收获。通过本次多个类的相互协作我也有了一些新的思考，以前的代码逻辑都是想着怎么通过一个类来描述对象的特征，这明显是不可能的，如果说一个对象的所有特征都用一个类来表示的话，当我们仅仅只修改一小部分数据成员就需要重新定义一个新的类，这无疑会造成代码的重复性极高，并且当类的数据成员过多的时候，就难以进行代码的维护，所以我们可以在适当的地方引入继承，当我们在进行多个类的相互协作的时候可以考虑他们是否有一些共同点，然后将他们的共同点抽象成一个基类，大家继承这个基类，这样就会大大缩减代码量，并且也很方便维

护。一个工程开始的时候是最困难的，最开始拿到这个实验的时候不知道从何处下手，怎么切入这个实验，这在当时困惑了我很久，这其实就是我最开始的代码思路问题，就想的是怎么能一步到位，但这种较大工程量的代码我们需要做的事情是对题目进行切割，将其划分成不同的类，然后在不同的类下面去具体实现即可。通过本次实验我觉得我最大的收获是初步学会了如何去切分一个大工程，然后怎么让他们相互配合！