

**课程设计报告**

**车辆管理系统设计**

|  |  |
| --- | --- |
| 学 院 | 计算机与通信工程学院 |
| 专 业 | 电子信息类 |
| 班级序号 | 210401 |
| 学 号 | **202112504** |
| 姓 名 | 杨靖普 |
| 指导教师 | 程龙 |
| 验收日期 | 2021年12月17日 |

核心知识点清单，由学生确认

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数组 | 链表 | 指针 | 文件读写 | 默认参数 | 函数模板 | 多文件 | 类 | 派生 | 虚函数 | 友元函数 | 重载 | 多继承 |
| √ | √ | √ | √ | √ |  |  | √ | √ |  | √ | √ | √ |

以下为教师评分表，学生不可填写

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 程序质量  （60%） | 课程设计报告（20%） | 答辩效果  （20%） | 总分 | 等级 |
|  |  |  |  |  |

评分标准与说明：

1. 程序质量（60%）包含程序正确性与所用知识点数量（40%），代码可读性（10%）与界面友好性（10%）。
2. 课程设计报告（20%）要求排版规范，模块设计有文字说明，图、表、代码清单要有序号和名字。
3. 现场答辩（20%）要求根据学生制作的PPT、讲述清晰、回答问题等情况综合评分。
4. 收齐所有纸质报告的同时，要求学委收集所有学生的代码工程、报告电子版和答辩PPT以备存档。

**设计要求：**

设计一车辆管理的应用程序，使其能完成简单的车辆管理功能。

管理功能包括添加车辆、查询车辆、显示车辆信息库、编辑车辆、删除车辆、删除车辆、车辆存盘、读取车辆，车辆分为大客车、小轿车、卡车等车辆，车辆信息包括车辆编号、车牌号、制造商、购买时间、型号、特性（卡车：载重量、小轿车：车厢数、大客车：载客量）、总共里、养路费、维护费、油耗以及总费用的管理，程序界面自己设计，从而达到对车库中车辆的简单管理的目的。

目录

1 基本功能描述 4

2 设计思路 5

3 软件设计 12

3.1 设计步骤 12

3.2 界面设计 12

3.3 关键功能的实现 18

4 结论与心得体会 51

5 参考文献 52

6 附录 52

6.1 调试报告 52

6.2 测试结果 55

6.3 关键源代码 70

车辆管理系统设计

# 1 基本功能描述

基于Windows的应用程序的车辆管理系统主要负责各种车辆的常规信息管理工作。系统中的车辆主要有大客车、小轿车和卡车。每种车辆有车辆编号、车牌号、车辆制造公司、车辆购买时间、车辆型号（大客车、小轿车和卡车）、总公里数、耗油量/公里、基本维护费用、养路费、累计总费用等信息。大客车还有载客量（最大载客数）信息，小轿车还有箱数（两厢或三厢）信息，卡车还有载重量等信息。

每台车辆当月总费用=油价\*耗油量/公里+基本维护费用。

基本维护费用：客车：2000元/月，小轿车：1000元/月，卡车：1500元/月

基本要求：

（1）添加车辆：主要完成车辆信息的添加，要求编号唯一。当添加了重复的编号时，则提示数据添加重复并取消添加；当车辆信息库已满，则提示不能再添加新的数据。

（2）查询车辆：

可按照三种方式来查询物品，分别为：

按车辆制造公司查询：输入车辆制造公司，输出所查询的信息，若不存在该记录，则提示“该车辆制造公司不存在！”；

按编号查询：输入编号，输出所查询的信息，若不存在该记录，则提示“该编号不存在！”；

按类别查询：输入类别，输出所查询的信息，若不存在记录，则提示“该类别没有车辆！”；

（3）显示车辆信息库：输出当前车辆信息库中所有车辆信息，每条记录占据一行。

（4）编辑功能：可根据查询结果对相应的记录进行修改，修改时注意编号的唯一性。

（5）删除车辆：主要完成车辆信息的删除。如果当前车辆信息库为空，则提示“车辆信息库为空！”，并返回操作；否则，输入要删除的编号，根据编号删除该车辆的记录，如果该编号不在车辆信息库库中，则提示“该编号不存在”。

（6）统计信息

输出当前车辆信息库中总物品数，以及按车辆类别，统计出当前车辆信息库中各类别的物品数并显示。

（7）车辆信息存盘：将当前程序中的车辆信息存入文件中。

（8）读出车辆信息：从文件中将车辆信息读入程序。

# 2 设计思路

如图是整个车辆管理系统的流程图，打开管理系统程序，输入用户需要实现的功能，调用输入数据子程序。子程序开始时top（车辆）为零，字符型的i为空。输入相应数字后，判断i的类型及数值，对其进行相应处理，然后再返回至主界面或进行其他功能的实现。

通过一个界面函数与其他功能函数进行互联，各个类函数函数之间互相调用，直至退出程序。

车辆管理系统包括添加车辆、查询车辆、显示车辆信息库、编辑车辆、删除车辆、统计车辆、存盘车辆和读取车辆的功能。通过这些功能实现对车辆编号、车牌号、车辆制造公司、车辆购买时间、车辆型号（大客车、小轿车和卡车）、总公里数、耗油量/公里、基本维护费用、养路费以及结合现实油费计算累计总费用，大客车的载客量（最大载客数），小轿车的箱数（两厢或三厢），卡车的载重量等信息的管理。

通过不同类函数之间的互联，实现功能连续不中断，让界面和管理系统显得更简洁高效。

如图是整个系统的流程图，打开图书管理系统，显示操作菜单

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*车辆管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1.添加车辆

2.查询车辆

3.显示车辆

4.编辑车辆

5.删除车辆

0.退出系统

请输入你需要操作的选项：

输入想要完成的操作的序号，0-5，开始执行。

输入数据子函数



图1 输入数据至子函数流程图



图2 车库管理系统的添加函数



图3 车库管理系统的查询函数



图4 车辆管理系统显示函数



图5 车辆管理系统删除函数



图6 车辆管理系统编辑函数

# 3 软件设计

## 3.1 设计步骤

打开DEV-C++,在起始页中点击新建项目中点击New中的Source File，建立新的源文件，点击确定并选择存储该文件的地址，如图8所示。

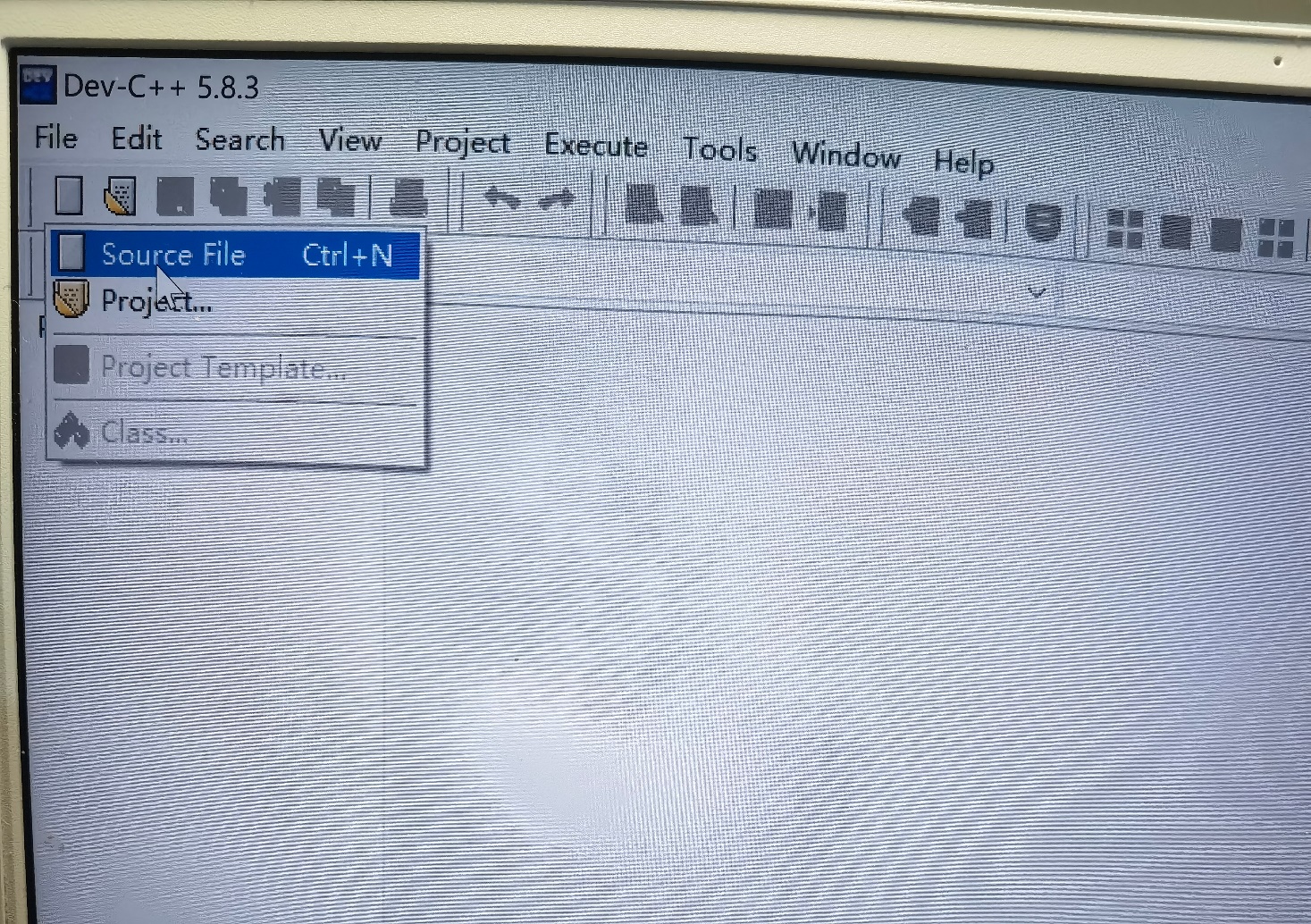


图7 新建文件的方法

## 

## 3.2 界面设计

1、主界面的设计

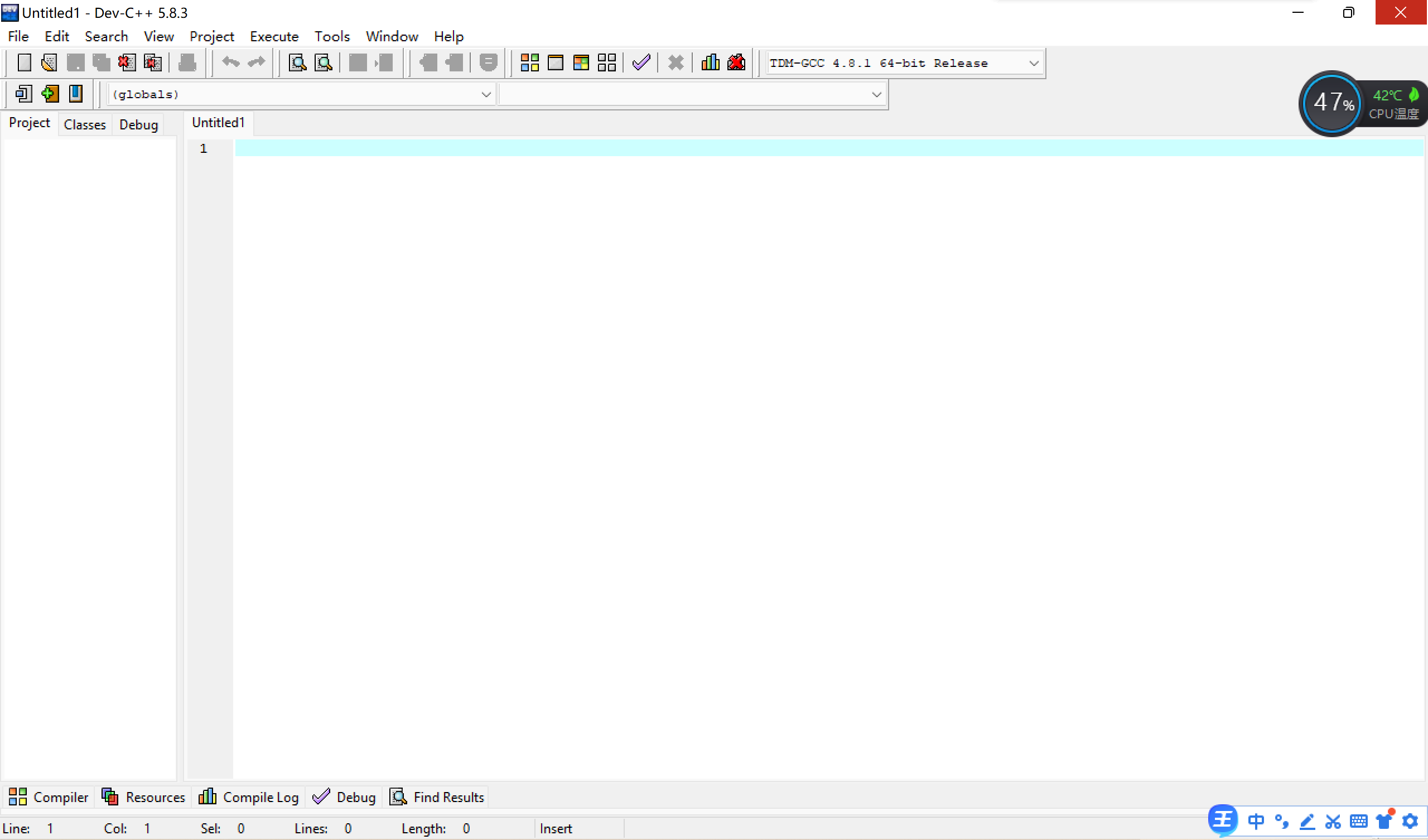


图8 源文件页面

写入界面函数，通过主函数调用界面友元类Manager中的jiemian函数，完成主界面的实现，从而实现主界面在运行窗口的显示，通过对输入数字的判断，以此来进行功能的实现。

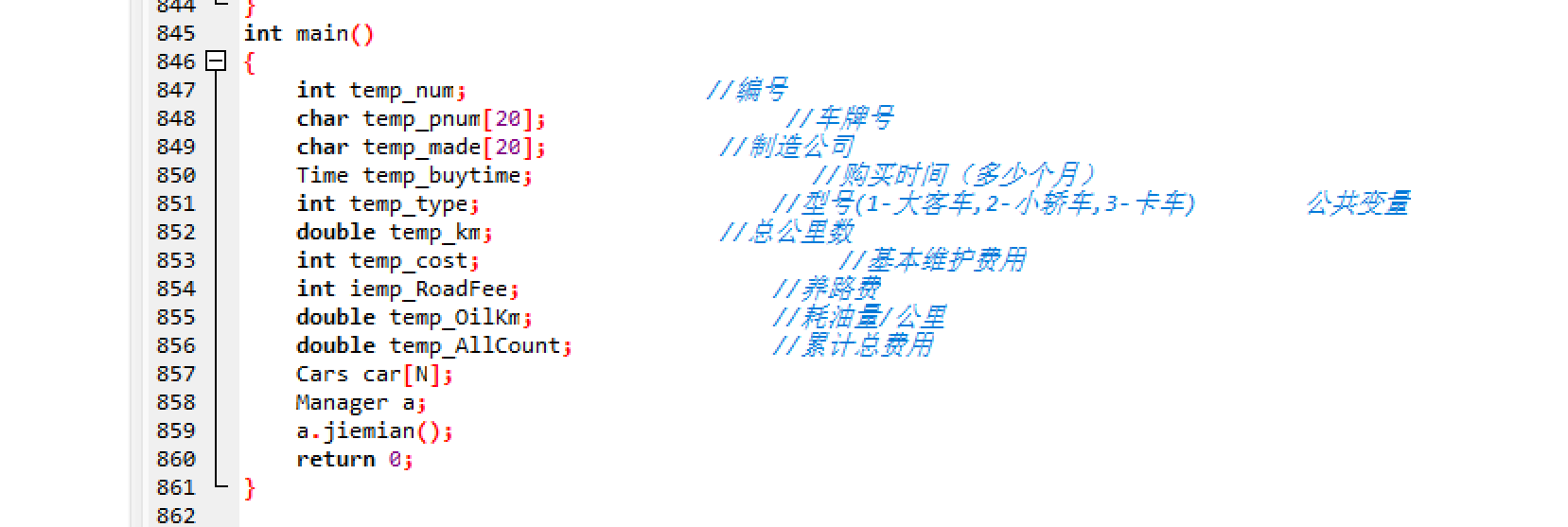


图9 主函数的代码显示

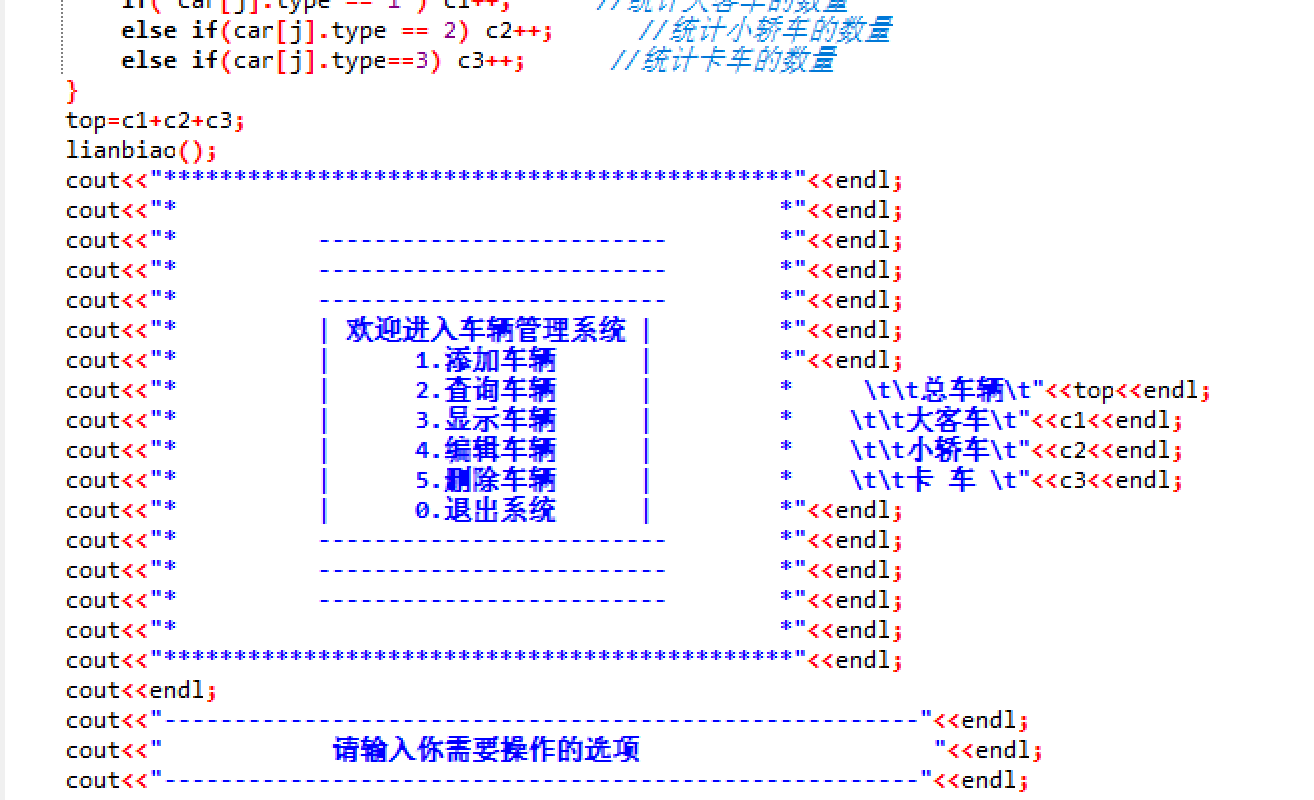


图10 jiemian函数的部分代码

在主界面函数的现实中，不仅显示了相关的功能所需输入对应数字，同时还统计不同车辆类型的数量以及车库中的车辆总数。从而更好地实现对车辆的管理以及数据的掌握和管控。



图11 主界面的显示

2、结束界面的设计

在结束界面中调用了menu函数，首先对是否退出程序进行确认，如果取消，则按任意键返回至主界面。如果确认退出，则使屏幕延迟0.5秒，并清空当前屏幕，同时展示“欢迎下次使用”的字样。

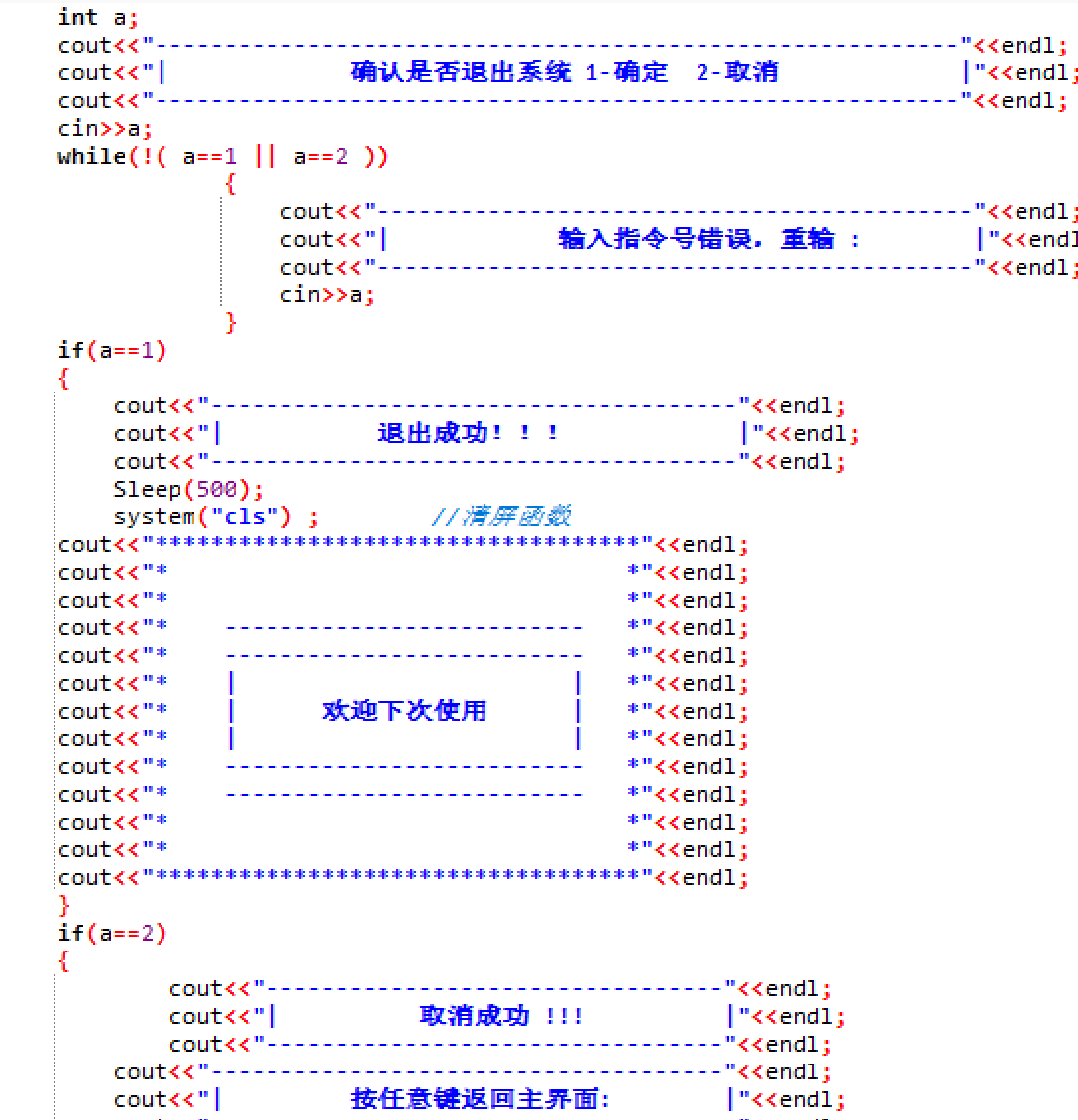


图12 menu函数的部分代码



图13 退出界面的功能实现（1）

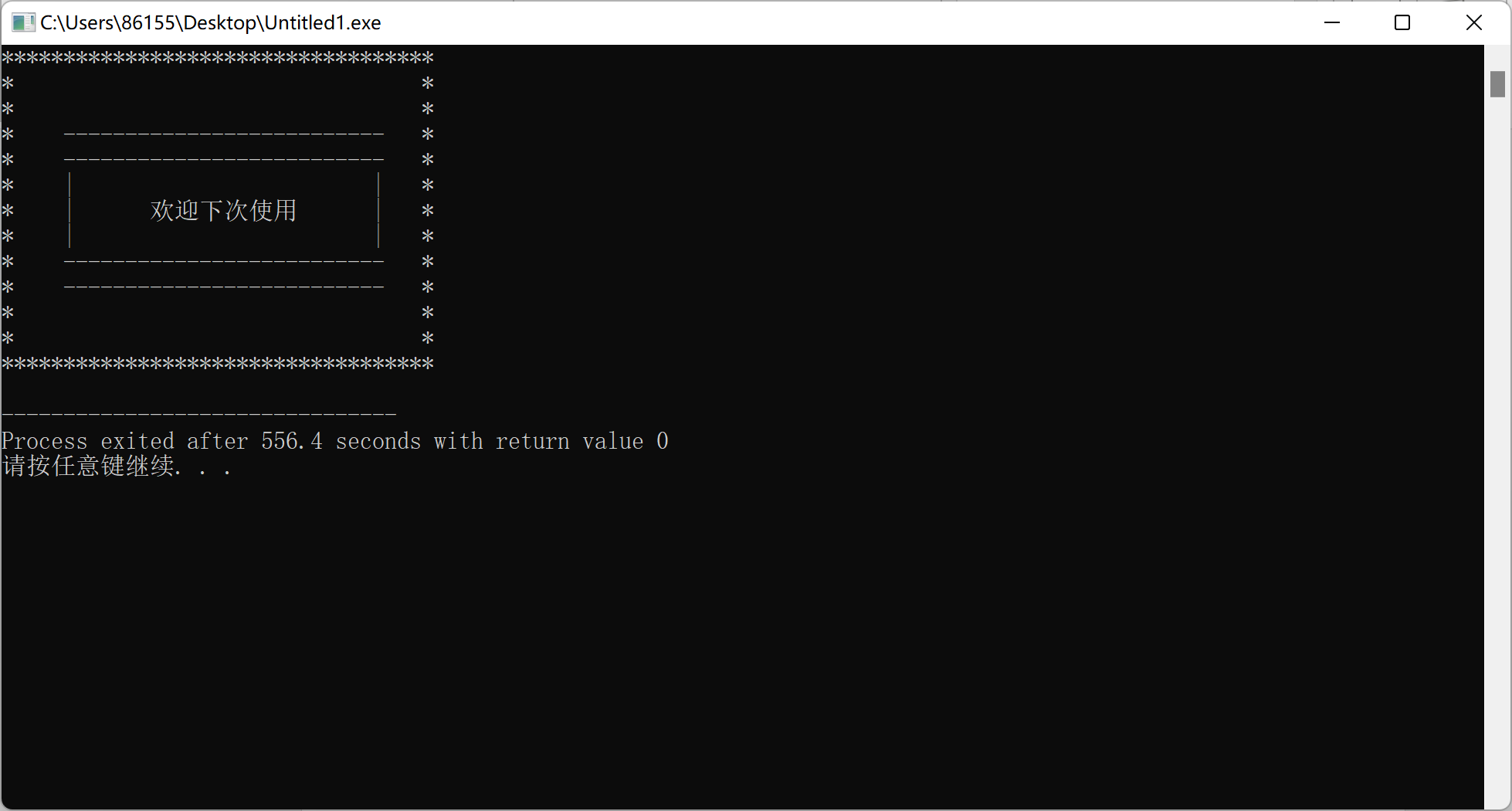


图14 退出界面的功能实现（2）



图15 退出界面的功能实现（3）

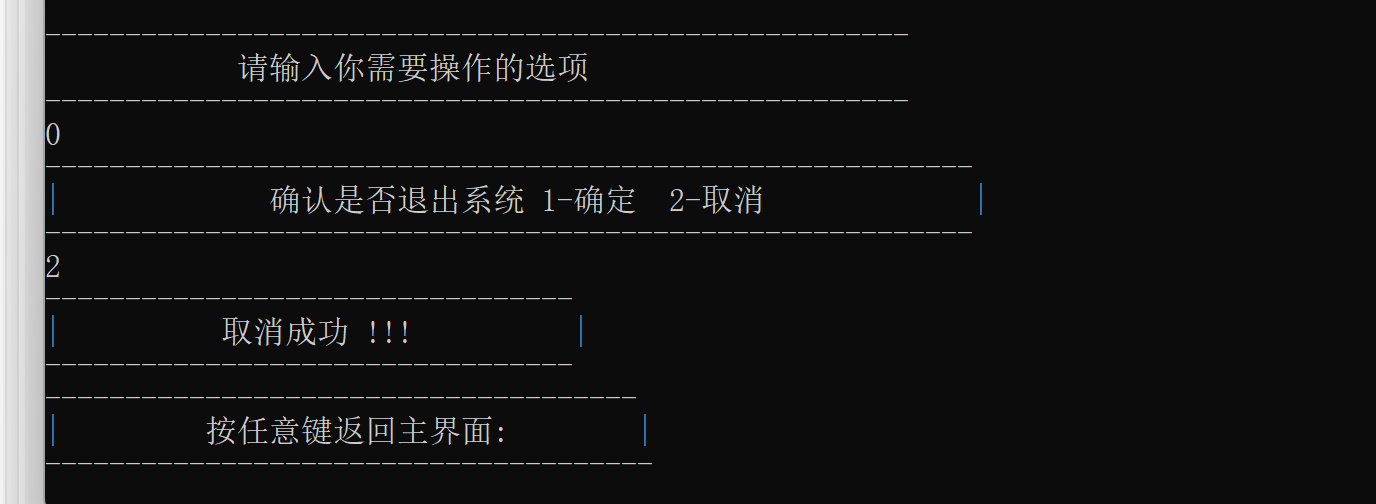


图16 退出界面的功能实现（4）

## 3.3 关键功能的实现

1）类的声明

声明Time类、Temp类、Car类、Manager类，其中Car类公有继承了Temp类和Time类中的数据，而Manager类设为Car类的友元函数，因此在Manager中声明函数可以调用Car类、Temp类、Time类中的private型的数据。具体参见（6）

Time类元素及其函数的声明

class Time

{

public:

Time()

{

year=0;

month=0;

day=0;

}

int year;

int month;

int day;

void set\_time();

void show\_time();

friend ostream& operator<<(ostream&output,Time&t);

};

Temp类的声明

class temp

{

public:

int temp\_num; //临时编号

string temp\_pnum; //临时车牌号

string temp\_made; //临时制造公司

Time temp\_buytime; //临时购买时间（多少个月）

string temp\_type[3]={"大客车","小轿车","卡车"}; //临时型号(1-大客车,2-小轿车,3-卡车) 公共变量

double temp\_km; //临时总公里数

int temp\_cost; //临时基本维护费用

int temp\_RoadFee; //临时养路费

double temp\_OilKm; //临时耗油量/公里

int people; //临时最大载客量（大客车）

int coach; //临时箱数（小轿车）

int weight;

};

Car类元素及函数的声明

class Cars:public Time,public temp //汽车类

{

public:

int num; //编号

string pnum; //车牌号

string made; //制造公司

Time buytime; //购买时间（多少个月）

int type; //型号(1-大客车,2-小轿车,3-卡车) 公共变量

double km; //总公里数

int cost; //基本维护费用

int RoadFee; //养路费

double OilKm; //耗油量/公里

int AllCount; //累计总费用

friend Manager; //友元函数 Manager

Cars \*next;

};

Manager类元素及其函数的声明

class Manager //总的类

{

private:

// Cars car[N]; //汽车数

int top; //记录当前车辆数量

int people[N]; //最大载客量（大客车）

int coach[N]; //箱数（小轿车）

int weight[N]; //载重量（卡车）

public:

Manager(){top = 0;}//车辆初始值为0 汽车总数

void add(); //添加车辆

void search(); //查询车辆

void show(); //显示车辆信息库

void edit(); //编辑功能

void delet(); //删除车辆

void sum(); //统计信息

void read(); //读出车辆信息

void write(); //车辆信息存盘

void jiemian(); //总界面

void lianbiao(); //链表初始化

void menu();

int count(int b,int c,int d,int a,int e=8,int f=7);

Cars car[N]; //汽车数

friend Cars;

Manager \*next;

};

2）添加车辆功能的实现

起初，车辆库中的top值设置为0，通过添加车辆功能使得，车库中的车辆增加，而添加时又通过车辆类型数量分别计算，使得不同类型的车辆数量得以分别展示。

在添加功能中，用户需要输入车辆编号、车牌号、车辆购买时间、养路费、车辆型号、以及车辆型号所对应的特性（大客车：载客数，小轿车：车厢数，卡车：载重量）、总公里数，系统现将输入的数据存放至temp类里面，在所有数据输入完毕后，系统将temp类中所有数据进行输出，让客户进行信息确认，如果用户确认无误后确定添加，则系统先根据用户输入的车辆类型进行判别，确定车辆的基本维护费用和每公里油耗，同时计算出车辆所需的总费用，放入Car类型的车库中，使得车库总数和其同类型的车辆数量加一，并调用链表函数，使得新数据加入至原链表中，成立新的链表关系。与此同时，调入存盘函数，将新的数据写入存储至指定位置的文本，使得信息得以保存。如果用户取消添加车辆，则车库数量不变，其之前储存的temp临时类型车辆也将不再赋值与车辆库类型中，同时也不再调用链表函数，保持原链表关系不变。

无论是否添加成功，系统都会询问是否继续添加车辆，以防用户填写错误后还需要删除车辆数据或重新调用添加函数，使得用户操作起来更方便。在用户取消继续添加车辆后，系统延迟1秒，清空当前屏幕，回到主界面中，继续其他功能。具体结果参见（6）

添加功能的主要相关代码

void Manager::add() //添加车辆 调用总的类之中的add函数

{

int a = 0, x, y,sm; // x代表车辆编号

while(1)

{

lianbiao();

if(top < N) //判断top当前已存车辆数小于汽车最大存储数N

{

cout<<"请输入需添加的车辆信息"<<endl;

cout<<"车辆编号：";

cin>>x;

for(int j=0; j<top;j++) //判断编号是否重复

{

if (car[j].num==x)

{

cout<<"---------------------------------------------------------"<<endl;

cout<<"| 编号重复,添加失败! 即将跳转至功能主界面 |"<<endl;

cout<<"---------------------------------------------------------"<<endl;

a++; //如果重复++，此时a不为0

break;

}

}

if(a==0) //此时编号不重复

{

temp\_plus.temp\_num=x;

car[top].num = x;

cout<<"车牌号：";

cin>>car[top].pnum;

temp\_plus.temp\_pnum=car[top].pnum;

cout<<"车辆制造公司：";

cin>>car[top].made;

temp\_plus.temp\_made=car[top].made;

cout<<"购买时间：年/月/日"<<endl;

car[top].buytime.set\_time();

temp\_plus.temp\_buytime.year=car[top].buytime.year;

temp\_plus.temp\_buytime.month=car[top].buytime.month;

temp\_plus.temp\_buytime.day=car[top].buytime.day;

cout<<"养路费：";

cin>>car[top].RoadFee;

temp\_plus.temp\_RoadFee=car[top].RoadFee;

cout<<"车辆型号(1-大客车,2-小轿车,3-卡车)：";

cin>>car[top].type;

while(!(car[top].type==1||car[top].type==2||car[top].type==3)) //判断输入指令是否为1-3

{

cout<<"输入指令号错误,重输：";

cin>>car[top].type;

}

cout<<"总公里数：";

cin>>car[top].km;

temp\_plus.temp\_km=car[top].km;

if( car[top].type==1) //此时是1-大客车，

{

cout<<"最大载客量：";

cin>>people[top];

temp\_plus.people=people[top];

car[top].cost = 2000; //基本维护费用

car[top].OilKm = 0.1; //耗油量/公里

}

else if( car[top].type == 2 ) //此时是2-小轿车

{

cout<<"车厢数：";

cin>>coach[top];

temp\_plus.coach=coach[top];

car[top].cost = 1000;

car[top].OilKm = 0.05;

}

else //此时是3-卡车

{

cout<<"载重量：";

cin>>weight[top];

temp\_plus.weight=weight[top];

car[top].cost = 1500;

car[top].OilKm = 0.15;

}

cout<<endl;

car[top].AllCount=count(car[top].km,car[top].buytime.year,car[top].buytime.month,car[top].cost,7,8);

//car[top].AllCount =(OilPrice \* car[top].OilKm \* car[top].km) +(((car[top].buytime.year\*12)+car[top].buytime.month)\*car[top].cost); //计算总费用

cout<<"-----------------------------------------"<<endl;

cout<<"| 信息确认 !!! |"<<endl;

cout<<"-----------------------------------------"<<endl;

cout<<"车辆编号："<<temp\_plus.temp\_num<<" ";

cout<<"车牌号: "<<temp\_plus.temp\_pnum<<" "<<endl;

cout<<"车辆制造公司："<<temp\_plus.temp\_made<<" ";

cout<<"购买时间: "<<temp\_plus.temp\_buytime<<endl;

cout<<"养路费： "<<temp\_plus.temp\_RoadFee<<" "<<"总公里数："<<temp\_plus.temp\_km<<endl;

cout<<"车型：";

if( car[top].type==1)

{

cout<<temp\_plus.temp\_type[0]<<" "<<"基本维护费用："<<car[top].cost;

cout<<" "<<"耗油量（升/公里)："<<car[top].OilKm<<" ";

cout<<"最大载客量："<<temp\_plus.people<<endl;

}

if( car[top].type==2)

{

cout<<temp\_plus.temp\_type[1]<<" "<<"基本维护费用："<<car[top].cost;

cout<<" "<<"耗油量（升/公里)："<<car[top].OilKm<<" ";

cout<<"车厢数："<<temp\_plus.coach<<endl;

}

if( car[top].type==3)

{

cout<<temp\_plus.temp\_type[2]<<" "<<"基本维护费用："<<car[top].cost;

cout<<" "<<"耗油量（升/公里)："<<car[top].OilKm<<" ";

cout<<"载重量："<<temp\_plus.weight<<endl;

}

cout<<"1.确认 2.取消 "<<endl;

cin>>sm;

while(!( sm==1 || sm==2 ))

{

cout<<"-------------------------------------------"<<endl;

cout<<"| 输入指令号错误，重输 : |"<<endl;

cout<<"-------------------------------------------"<<endl;

cin>>sm;

}

if(sm==1)

{

write(); //调用车辆信息存盘函数,在下面定义

cout<<"-------------------------------"<<endl;

cout<<"| 存储成功 !!! |"<<endl;

cout<<"-------------------------------"<<endl;

lianbiao();

top++; //车辆总数加一1

}

else if(sm==2)

{

cout<<"--------------------------------"<<endl;

cout<<"| 取消成功 ！！ |"<<endl;

cout<<"--------------------------------"<<endl;

car[top].type =0;

}

cout<<"是否继续添加车辆？(1-是,2-否)：";

cin>>y;

while(!( y==1 || y==2 ))

{

cout<<"-------------------------------------------"<<endl;

cout<<"| 输入指令号错误，重输 : |"<<endl;

cout<<"-------------------------------------------"<<endl;

cin>>y;

}

if(y == 2)

{

cout<<"---------------------------------------------"<<endl;

cout<<"| 即将跳转至功能主界面.... |"<<endl;

cout<<"---------------------------------------------"<<endl;

break;

}

}

else //此时编号重复

break;

}

else //top>N

{

cout<<"-----------------------------------------"<<endl;

cout<<"| 车辆信息库已满,不能再添加车辆 ! |"<<endl;

cout<<"| 即将跳转至功能主界面.... |"<<endl;

cout<<"-----------------------------------------"<<endl;

break;

}

}

Sleep(1000); //延迟1秒

system("cls") ;//清屏函数

jiemian(); //调用总界面函数

}

3）查询车辆功能的实现

查询车辆功能的实现是通过链表实现的，首先调用链表函数，建立链表关系，输入查询方式及车辆的相关信息，创建指向Car类的指针\*head、\*p、\*next，将Cars[0]的地址给\*head，然后依次将cars[i]的指针指向cars[i+1]的地址，直到i等于车库的数量减一，再将最后一个数据的地址设为NULL，这样，将head的地址给P，P从cars[0]开始寻找输入的相关信息，直到\*P的地址指向NULL，如果找到相关信息，跳出循环，显示车辆相关信息。并询问用户是否修改该信息，如果选择修改，则调用修改车辆的函数，否则，返回主界面继续其他功能的选择。如果未找到相关信息，则返回车辆相关信息不存在，按任意键返回主界面。具体结果参见（6）

建立链表函数相关代码

void Manager::lianbiao()

{

for(int i=0;i<top-1;i++)

{

car[i].next=&car[i+1];

}

car[top-1].next=NULL;

}

添加功能的主要相关代码

void Manager::search() //查询车辆

{

int choose,m,n,a=0;

char i,t[20];

Cars \*p,\*head;

head=&car[0];

lianbiao();

if(top == 0) //车库为空无法查询

{

cout<<"-----------------------------------------------------------"<<endl;

cout<<"| 车辆信息库为空！系统将返回功能界面.... |"<<endl;

cout<<"-----------------------------------------------------------"<<endl;

Sleep(600);

system("cls");

jiemian();

}

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* | 1.按车辆制造公司查询 | \*"<<endl;

cout<<"\* | 2.按编号查询 | \*"<<endl;

cout<<"\* | 3.按类别查询 | \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 请输入你的选项：";

while(1)

{

cin>>i;

cout<<endl;

if(i=='1'||i=='2'||i=='3')

break;

cout<<"输入错误,重输：";

}

if(i=='1')

{

cout<<"请输入需查询车辆的制造公司：";

cin>>t;

}

else if(i =='2')

{

cout<<"请输入需查询的车辆编号：";

cin>>n;

}

else if(i == '3')

{

cout<<"请输入需查询的车辆型号(1-大客车,2-小轿车,3-卡车)：";

cin>>m;

}

for(p=head;p!=NULL;p=p->next)

{

if(p->made==t || p->num==n || p->type==m) //string字符串比较.

{ cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"编号 车牌号 制造商 购买时间 型号 特性 "<<endl;

a++; //使a不为0

break;

}

}

for(p=head;p!=NULL;p=p->next)

{

int j=0;

if(p->made==t || p->num==n || p->type==m)

{

cout<<setw(2)<<p->num<<setw(14)<<p->pnum<<setw(15)<<p->made<<setw(14);

cout<<p->buytime;

if(p->type==1 )

cout<<setw(14)<<"大客车"<<setw(14)<<"载客量:"<<people[j]<<endl; //输出特性

else if(car[j].type == 2 )

cout<<setw(14)<<"小轿车"<<setw(14)<<"车厢数: "<<coach[j]<<endl;

else

cout<<setw(14)<<"卡 车"<<setw(14)<<"载重量:"<<weight[j]<<endl;

cout<<"----------------------------------------------------------------------------"<<endl;

cout<<"--------------------------------------------------------------"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"总公里 养路费 维护费 油耗 总费用"<<endl;

cout<<setw(2)<<p->km<<setw(15)<<p->RoadFee<<setw(16)<<p->cost;

cout<<setw(14)<<p->OilKm<<setw(16)<<p->AllCount<<endl;

}

j++;

}

if(a!=0)

{

cout<<"---------------------------------------------------------"<<endl;

cout<<"| 是否进行修改?(1-是,2-否)： |"<<endl;

cout<<"---------------------------------------------------------"<<endl;

cin>>choose;

if(choose == 1)

edit(); //调用编辑功能函数

}

else //上述a++的函数未成功，最上面1，2，3的查询结果没有

{

if(i=='1')

{

cout<<"--------------------------------------------------"<<endl;

cout<<"| 车辆制造公司不存在！ |"<<endl;

cout<<"--------------------------------------------------"<<endl;

}

if(i=='2')

{

cout<<"---------------------------------------------------"<<endl;

cout<<"| 车辆编号不存在！ |"<<endl;

cout<<"---------------------------------------------------"<<endl;

}

else if (i=='3')

{

cout<<"----------------------------------------------------"<<endl;

cout<<"| 车辆型号不存在！ |"<<endl;

cout<<"----------------------------------------------------"<<endl;

}

}

cout<<"-----------------------------------------------------------"<<endl;

cout<<"| 按任意键返回功能主界面 |"<<endl;

cout<<"----------------------------------------------------------"<<endl;

char zz=getch(); //无回显的取一个字符

Sleep(500);

system("cls") ;//清屏函数

jiemian(); //调用界面函数，返回主界面

}

4）车辆库显示功能的实现

显示车辆库信息功能的实现是通过链表实现的，首先调用链表函数，建立链表关系，，创建指向Car类的指针\*head、\*p、\*next，将Cars[0]的地址给\*head，然后依次将cars[i]的指针指向cars[i+1]的地址，直到i等于车库的数量减一，再将最后一个数据的地址设为NULL，这样，将head的地址给P，P从cars[0]的信息开始依次向后显示，直至P指向NULL，显示完毕，按任意键返回主界面。具体结果参见（6）。

车辆库信息显示功能的相关代码

void Manager::show()

{

Cars \*p,\*head;

head=&car[0];

p=head;

int j=0;

lianbiao();

if(top != 0) //top记录当前车辆数目

{

do

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"编号 车牌号 制造商 购买时间 型号 特性 "<<endl;

cout<<setw(3)<<p->num<<setw(13)<<p->pnum<<setw(15)<<p->made<<setw(16);

cout<<p->buytime;

if(p->type == 1 )

cout<<setw(15)<<"大客车"<<setw(15)<<"载客量: "<<setw(2)<<people[j]<<endl;

else if(p->type == 2 )

cout<<setw(15)<<"小轿车"<<setw(15)<<"车厢数: "<<setw(2)<<coach[j]<<endl;

else

cout<<setw(15)<<"卡 车 "<<setw(15)<<"载重量: "<<setw(2)<<weight[j]<<endl; cout<<"------------------------------------------------------------"<<endl;

cout<<"总公里 养路费 维护费 油耗 总费用"<<endl;

cout<<setw(3)<<p->km<<setw(13)<<p->RoadFee<<setw(13)<<p->cost<<setw(13)<<p->OilKm<<setw(13)<<p->AllCount<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<endl;

p=p->next;

j++;

}

while(p!=NULL);

}

else

{

cout<<"----------------------------------"<<endl;

cout<<"| 车辆库为空！！！ |"<<endl;

cout<<"----------------------------------"<<endl;

cout<<endl;

}

cout<<"-------------------------------------"<<endl;

cout<<"| 按任意键返回主界面: |"<<endl;

cout<<"--------------------------------------"<<endl;

char z=getch(); //从输出获得一个字符之后调用界面函数

system("cls") ; //清屏函数

jiemian();

}

5）车辆编辑功能的实现

调用车辆编辑函数，显示车辆库中所有车辆的车辆编号，以便用户对车库中编号一目了然，更快的进行车辆信息修改，使得编辑操作更方便，选择需要修改的车辆编号进行修改，修改完毕后系统将显示所有修改信息，进行信息确认，确认后调用存盘函数，将修改后的信息写入存储至指定位置的文本，调用界面函数，返回至主界面。如果取消确认，则按任意键返回至主界面，具体结果参见（6）。

车辆编辑功能的主要代码

void Manager::edit() //编辑车辆，既修改车辆

{

int i, j, a = 0,sm,q;

lianbiao();

Cars \*p,\*head;

head=&car[0];

if(top == 0) //车库为空无法编辑

{

cout<<"---------------------------------------------------------------"<<endl;

cout<<"| 车辆信息库为空，按任意键返回功能主界面 |"<<endl;

cout<<"---------------------------------------------------------------"<<endl;

char zz=getch(); //无回显的取一个字符

Sleep(500);

system("cls") ;//清屏函数

jiemian(); //调用界面函数

}

cout<<"------------------------------------------------------------"<<endl;

cout<<"| 车辆信息库中有以下车辆编号 |"<<endl;

cout<<"------------------------------------------------------------"<<endl;

for(int i=0;i<top;i++)

{

cout<<setw(10)<<car[i].num;

}

cout<<endl;

cout<<"-------------------------------------------------"<<endl;

cout<<"| 请依次输入需要修改的车辆信息： |"<<endl;

cout<<"-------------------------------------------------"<<endl;

while(1)

{

cout<<"车辆编号：";

cin>>i;

for(j = 0; j < top; j++)

{

if(car[j].num == i) //车库有此编号的车

{

cout<<"车牌号：";

cin>>temp\_plus.temp\_pnum;

cout<<"制造公司：";

cin>>temp\_plus.temp\_made;

cout<<"购买时间: 年/月/日"<<endl;

car[j].set\_time();

cout<<"型号(1-大客车，2-小轿车，3-卡车)：";

cin>>temp\_plus.temp\_types;

while(!(temp\_plus.temp\_types==1||temp\_plus.temp\_types==2||temp\_plus.temp\_types==3)) //判断是否有误

{

cout<<"---------------------------------------------------"<<endl;

cout<<" 输入信息有误，请输入1-3： "<<endl;

cout<<"---------------------------------------------------"<<endl;

cin>>temp\_plus.temp\_types;

}

cout<<"总公里数：";

cin>>temp\_plus.temp\_km;

cout<<"养路费：";

cin>>temp\_plus.temp\_RoadFee;

if( temp\_plus.temp\_types == 1 ) //此时是1-大客车

{

cout<<"最大载客量：";

cin>>temp\_plus.people;

temp\_plus.temp\_cost = 2000;

temp\_plus.temp\_OilKm = 0.1;

}

else if(temp\_plus.temp\_types == 2 ) //此时是2-小轿车

{

cout<<"车厢数：";

cin>>temp\_plus.coach;

temp\_plus.temp\_cost = 1000;

temp\_plus.temp\_OilKm = 0.05;

}

else //此时是3-卡车

{

cout<<"载重量：";

cin>>temp\_plus.weight;

temp\_plus.temp\_cost = 1500;

temp\_plus.temp\_OilKm = 0.15;

}

cout<<endl;

cout<<"---------------------------------------------"<<endl;

cout<<"| 信息确认 !!! |"<<endl;

cout<<"---------------------------------------------"<<endl;

cout<<"车辆编号："<<temp\_plus.temp\_num<<" ";

cout<<"车牌号: "<<temp\_plus.temp\_pnum<<" "<<endl;

cout<<"车辆制造公司："<<temp\_plus.temp\_made<<" ";

cout<<"购买时间: "<<temp\_plus.temp\_buytime<<endl;

cout<<"养路费： "<<temp\_plus.temp\_RoadFee<<" "<<"总公里数："<<temp\_plus.temp\_km<<endl;

cout<<"车型：";

if( temp\_plus.temp\_types==1)

{

cout<<temp\_plus.temp\_type[0]<<" ";

cout<<"最大载客量："<<temp\_plus.people<<endl;

}

if( temp\_plus.temp\_types==2)

{

cout<<temp\_plus.temp\_type[1]<<" ";

cout<<"车厢数："<<temp\_plus.coach<<endl;

}

if( temp\_plus.temp\_types==3)

{

cout<<temp\_plus.temp\_type[2]<<" ";

cout<<"载重量："<<temp\_plus.weight<<endl;

}

cout<<"--------------------------------------------------"<<endl;

cout<<"| 1.确认 2.取消 |"<<endl;

cout<<"--------------------------------------------------"<<endl;

cin>>sm;

while(!( sm==1 || sm==2 ))

{

cout<<"-------------------------------------------"<<endl;

cout<<"| 输入指令号错误，重输 : |"<<endl;

cout<<"-------------------------------------------"<<endl;

cin>>sm;

}

if(sm==1)

{

car[j].pnum=temp\_plus.temp\_pnum;

car[j].made=temp\_plus.temp\_made;

car[j].buytime.year=temp\_plus.temp\_buytime.year;

car[j].buytime.month=temp\_plus.temp\_buytime.month;

car[j].buytime.day=temp\_plus.temp\_buytime.day;

car[j].RoadFee=temp\_plus.temp\_RoadFee;

car[j].km=temp\_plus.temp\_km;

car[j].type=temp\_plus.temp\_types;

people[j]=temp\_plus.people;

coach[j]=temp\_plus.coach;

weight[j]=temp\_plus.weight;

car[j].cost=temp\_plus.temp\_cost;

car[j].OilKm=temp\_plus.temp\_OilKm;

write(); //调用存盘函数

a++; //成功后加一

}

else if(sm==2)

{

cout<<"------------------------------------------------------"<<endl;

cout<<"| 取消修改 ！即将返回至页面..... |"<<endl;

cout<<"------------------------------------------------------"<<endl;

Sleep(700);

system("cls") ; //清屏函数

jiemian();

}

}

}

if(a == 0) //未存成功，说明有错误

{

cout<<"-------------------------------------------------"<<endl;

cout<<"| 此编号不存在，不能进行修改，请重新输入! |"<<endl;

cout<<"-------------------------------------------4------"<<endl;

}

else

{

cout<<"--------------------------------------------------------"<<endl;

cout<<"| 修改成功！系统将返回主界面.... |"<<endl;

cout<<"--------------------------------------------------------"<<endl;

Sleep(700);

system("cls") ; //清屏函数

break; //跳出while循环

}

}

Sleep(700); //延迟0.7秒

jiemian();

}

6） 车辆删除功能的实现

调用车辆删除函数，首先判断车辆库中车辆数量是否为零，如果为零，则无法删除车辆，返回“车辆信息库为空”，无法删除车辆，如果数量不为零，则显示车辆库中所有车辆的编号，以便用户了解库中所有编号信息，快速进行删除，方便用户操作，提高车辆管理效率，删除后将所有删除车辆后面的车辆信息依次提前，进行补位，然后再调用链表函数，建立新的链表关系，最后调用存盘函数，写入删除后的所有车辆信息，具体结果参见（6）

车辆删除功能主要编码

void Manager::delet() //删除车辆

{

int i, a = 0; //a用来判断循环是否成功

Cars \*p,\*head;

head=&car[0];

while(1)

{

if(top == 0) //车库为空无法删除

{

cout<<"-----------------------------------------------------------"<<endl;

cout<<"| 车辆信息库为空！系统将返回功能界面.... |"<<endl;

cout<<"-----------------------------------------------------------"<<endl;

break; //跳出while循环

}

else //不为空

{

cout<<"------------------------------------------------------------"<<endl;

cout<<"| 车辆信息库中有以下车辆编号 |"<<endl;

cout<<"------------------------------------------------------------"<<endl;

for(int i=0;i<top;i++)

{

cout<<setw(10)<<car[i].num;

}

cout<<endl;

cout<<"请输入需要删除的车辆编号：";

cin>>i;

for(int j = 0; j < top; j++) //for循环用来删除所有存盘的信息

{

if(car[j].num == i) //车辆编号存在

{

if(top==1)

car[0].type=0; //就一辆车，使其变为0

else //top不等于1

{

if( car[top - 1].type == 1 ) //判断出此时car数组里储存在最后一组的是1-大客车

{

people[j] = people[j+1]; //people载客量

car[j].cost = 2000;

car[j].OilKm = 0.1;

}

else if(car[top - 1].type == 2 ) //判断出此时car数组里储存在最后一组的是2-小轿车

{

coach[j] = coach[top - 1];

car[j].cost = 1000;

car[j].OilKm = 0.05;

}

else if(car[top - 1].type == 3 ) //判断出此时car数组里储存在最后一组的是3-卡车

{

weight[j] = weight[j+1];

car[j].cost = 1500;

car[j].OilKm = 0.15;

}

car[j].num = car[top-1].num;

car[j].pnum, car[top-1].pnum;

car[j].made, car[top-1].made;

car[j].buytime = car[top-1].buytime;

car[j].type = car[top-1].type;

car[j].km = car[top-1].km;

car[j].RoadFee = car[top-1].RoadFee;

car[j].AllCount = car[top-1].AllCount;

car[j+1].type=0;

}

a++;

lianbiao();

write(); //存盘函数

}

}

if(a == 0)

{

cout<<"----------------------------------------------------------"<<endl;

cout<<"| 此编号不存在，不能进行删除，请重新输入！ |"<<endl;

cout<<"----------------------------------------------------------"<<endl;

}

else

{

cout<<"-------------------------------------------------------"<<endl;

cout<<"| 删除成功！系统将返回功能界面.... |"<<endl;

cout<<"--------------------------------------------------------"<<endl;

Sleep(700); //延迟0.7秒。

system("cls") ; //清屏函数

break;

}

}

}

Sleep(300); //延迟0.3秒

system("cls") ; //清屏函数

jiemian();

}

7）文件写入、读取功能的实现

通过对文件的操作，创建txt文件，调用write函数，将相关车库信息写入，然后再通过read函数，将车辆信息从txt文件中读取出来，进行文件的输入和输出，从而进行面向对象的操作。

文件写入功能的相关代码

void Manager::write() //车辆信息存盘

{

ofstream write;

write.open("d:\\b.txt",ios::out); //以输出方式打开文件

if(write)

{

for(int i = 0; i <=top; i++)

{

write<<endl;

write<<car[i].num<<" "<<car[i].pnum<<" "<<car[i].made<<" ";

write<<car[i].buytime.year<<" "<<car[i].buytime.month<<" "<<car[i].buytime.day;

write<<" "<<car[i].type<<" ";

if(car[i].type == 1)

write<<people[i];//存入 载客量

else if(car[i].type == 2)

write<<coach[i];//存入 车厢数

else

write<<weight[i];//存入 载重量

write<<" "<<car[i].km<<" "<<car[i].RoadFee<<" "<<car[i].cost<<" "<<car[i].OilKm<<" "<<car[i].AllCount;

}

}

}

文件读取功能的相关代码

void Manager::read() ///读出车辆信息

{

while(1)

{

ifstream read("d:\\b.txt",ios::in);

if(read)

{

while(!read.eof()) //文件不结束，还有内容

{

read>>car[top].num>>car[top].pnum>>car[top].made;

read>>car[top].buytime.year>>car[top].buytime.month>>car[top].buytime.day>>car[top].type;

if(car[top].type == 1)

read>>people[top];//读取 载客量

else if(car[top].type == 2)

read>>coach[top];//读取 车厢数

else

read>>weight[top];//读取 载重量

read>>car[top].km>>car[top].RoadFee>>car[top].cost>>car[top].OilKm>>car[top].AllCount;

if(car[top].type==1||car[top].type==2||car[top].type==3)

top++;

}

read.close(); //关闭文件

break;

}

else

{

ofstream outfile("d:\\a.txt");

outfile.close();

}

}

}

# 4 结论与心得体会

经过多番测试，得出结论为设计的车辆管理系统能够完成功能描述内容中的全部功能。

车辆管理系统这样一个小小的程序中有许许多多的功能，其实这些功能都是由一个个函数得以实现，主函数必不可少，但也只是起到了一个调用的作用，真正有用的地方是函数的部分，而我也是这么做的。在主函数中只有初始化数据和函数调用这两个作用。功能的实现都存放在了一个个函数中。

一个好的c++程序，算法应该好，界面应该简洁明了，应该有充足的注释让别人去看懂明白，我应该尽力实现这些目标。

在写报告的过程中，我体会到了无论是做什么事都应该严谨，符合规范要求，不天马行空地随意添加，不画蛇添足，也不能缺少东西，严格按照要求去做，这对于无论是写代码，还是做报告都是十分重要的。

经过2个星期的C++课程[设计](http://action.foho.cc/click/click.php?r=http%3A//www.baidu.com/s%3Flm%3D0%26si%3D%26rn%3D10%26ie%3Dgb2312%26ct%3D0%26wd%3D%25BF%25CE%25B3%25CC%25C9%25E8%25BC%25C6+%25D0%25C4%25B5%25C3%25CC%25E5%25BB%25E1%26pn%3D10%26ver%3D0%26cn%3Dbaidu16%26cl%3D3%26uim%3D1%26usm%3D0&ads_id=22&site_id=163&click=1&url=http%3A//www.totole.com.cn/activity/opus.aspx&v=0&k=%u8BBE%u8BA1&s=http%3A//www.ttshopping.net/Article/726/show/308349382572.html&rn=763357" \t "_blank" \o "plash笑话之霸王别鸡篇 @foho.cc)，我在老师和同学的共同帮助下，顺利完成该课程设计，收获很大。本学期我们开设了《C++程序设计》的课程设计教学内容，正所谓“纸上谈兵终觉浅，觉知此事要躬行。”所以在本学期课程设计是很及时、很必要的。这样不仅能加深我们对C++程序设计的掌握，而且还及时、真正的做到了学以致用。主要是以下几点。

基本功要过硬。不能马虎经常犯犯小错误。提高自己找错纠错能力，通常错误总是那几种，多练习就熟练了。

要善于查找资料提高自己的资料查询能力，开始着手程序设计时，有些功能感觉无从下手，都不知道要干什么。这些问题只要找到相关的资料就会得到提示从而理解所要解决问题的大致方向。有了明确的目的，就可以发挥自己的头脑来完成它了。

要善于思考。程序中有些实现难点和最初完成程序时免不了的许多不足就需要自己根据代码认真思考来解决。只有这样才能纠正程序的不足并使其更加人性化。

最后还有，算法永远都有更好的，我们要做的只不过是不断地改进自己的算法、自己的思路，力求做到最好，程序的功能更加完善、更加简单。希望自己能够不断进步，做到更好、更优秀。

这次课设带给我的不只是技能的提升，更是锻炼了我的心境，培养了我的性格，因此我很感激这次课设，我也真心地希望我能得到更多的锻炼机会！

# 5 参考文献

[1] 谭浩强.C++程序设计（第3版）.清华大学出版社，2004

[2] 【美】史蒂芬.普拉达. C++ primer Plus（第6版）人民邮电出版社.

# 6 附录

## 6.1 调试报告

1、编译时提示错误：

C:\Users\86155\Desktop\Untitled1.cpp [Error] stray '\243' in program

原因为分号是在中文输入法状态下输入，将其改为英文输入状态下了分号即可。

C:\Users\86155\Desktop\Untitled1.cpp [Error] expected ';' before 'cout'

C:\Users\86155\Desktop\Untitled1.cpp [Error] expected ';' at end of member declaration

C:\Users\86155\Desktop\Untitled1.cpp [Error] expected ';' after class definition

原因是没有在语句和类的末尾添加分号。再末尾添加分号即可。

C:\Users\86155\Desktop\Untitled1.cpp [Error] no match for 'operator>>' (operand types are 'std::ostream {aka std::basic\_ostream<char>}' and 'const char [34]')

C:\Users\86155\Desktop\Untitled1.cpp [Error] no match for 'operator<<' (operand types are 'std::istream {aka std::basic\_istream<char>}' and 'int')

原因是输入输出流的方向写反，调换过来即可。

C:\Users\86155\Desktop\collect2.exe [Error] ld returned 1 exit status

原因是已经打开了一个程序，无法再进行另一个程序，把原程序关掉即可。

C:\Users\86155\Desktop\Untitled1.cpp [Error] no match for 'operator=' (operand types are 'Cars' and 'Cars\*')

C:\Users\86155\Desktop\Untitled1.cpp [Error] cannot convert 'Cars' to 'Cars\*' in assignment

C:\Users\86155\Desktop\Untitled1.cpp [Error] cannot convert 'Cars\*\*' to 'Cars\*' in assignment

C:\Users\86155\Desktop\Untitled1.cpp [Error] lvalue required as left operand of assignment

原因是指针使用的方法不当，使用正确的方法对指针进行操作和赋值即可。

C:\Users\86155\Desktop\Untitled1.cpp [Error] 'Manager::Manager()' is private

原因是类中相关元素是私有的无法直接访问，将元素设置为公有或运用类内函数调用私有数据即可。

C:\Users\86155\Desktop\Untitled1.cpp [Error] default argument missing for parameter 2 of 'int Manager::count(int, int, int, int, int, int)'

C:\Users\86155\Desktop\Untitled1.cpp [Error] default argument missing for parameter 3 of 'int Manager::count(int, int, int, int, int, int)'

原因是默认参数的赋值时含默认参数的形参要放到右侧，将不含默认参数的形参放在左侧即可。

C:\Users\86155\Desktop\Untitled1.cpp [Error] declaration of 'void Manager::lianbiao()' outside of class is not definition [-fpermissive]

原因是在函数声明后面多加了分号，去掉分号即可。

2、当给int型的参数赋以小数的时候，其只会显示整数部分，此时要将int型改成double型，再进行其他功能的实现和函数计算。

3.对类中的数据进行初始化，否则会出现数据乱码的情况。

4．在建立链表时需要弄清相互之间的关系，防止关系不明，导致指针指向错误，产生关系建立错误

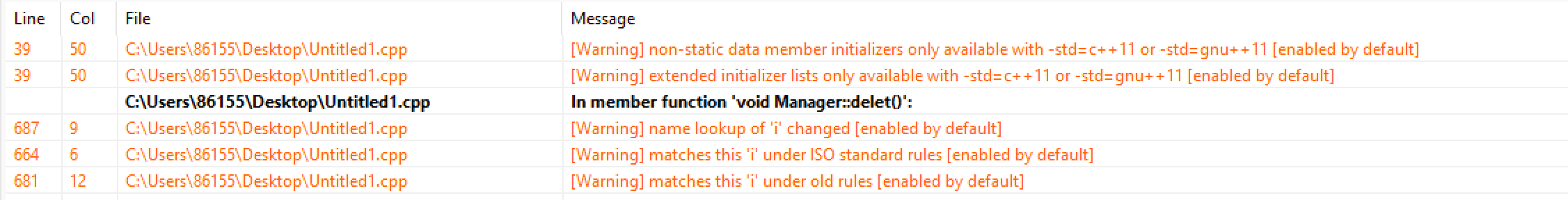


图17 最终调试结果

## 6.2 测试结果

1.添加车辆功能的实现

****

图18 添加车辆功能实现（1）

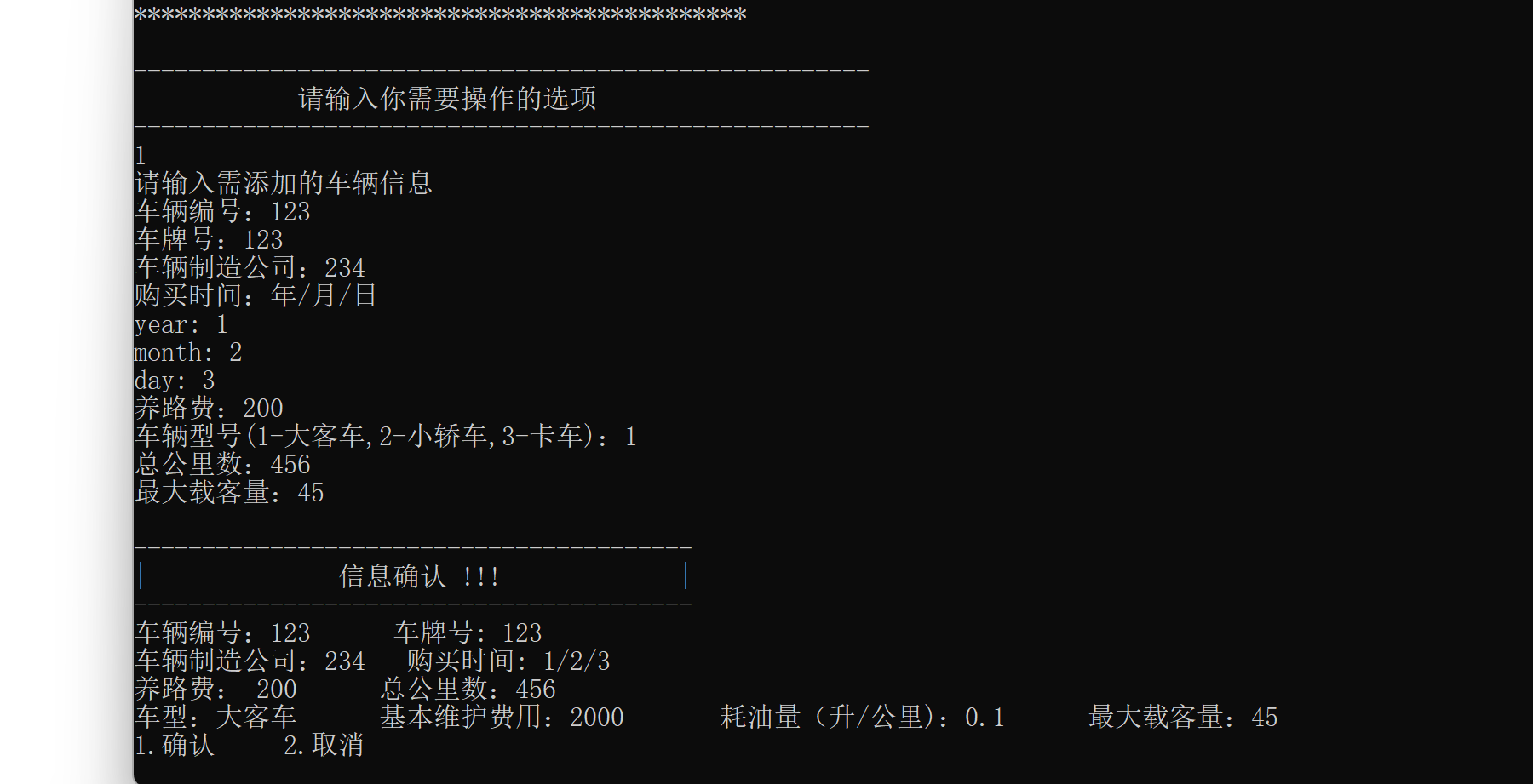


图19 添加车辆功能实现（2）

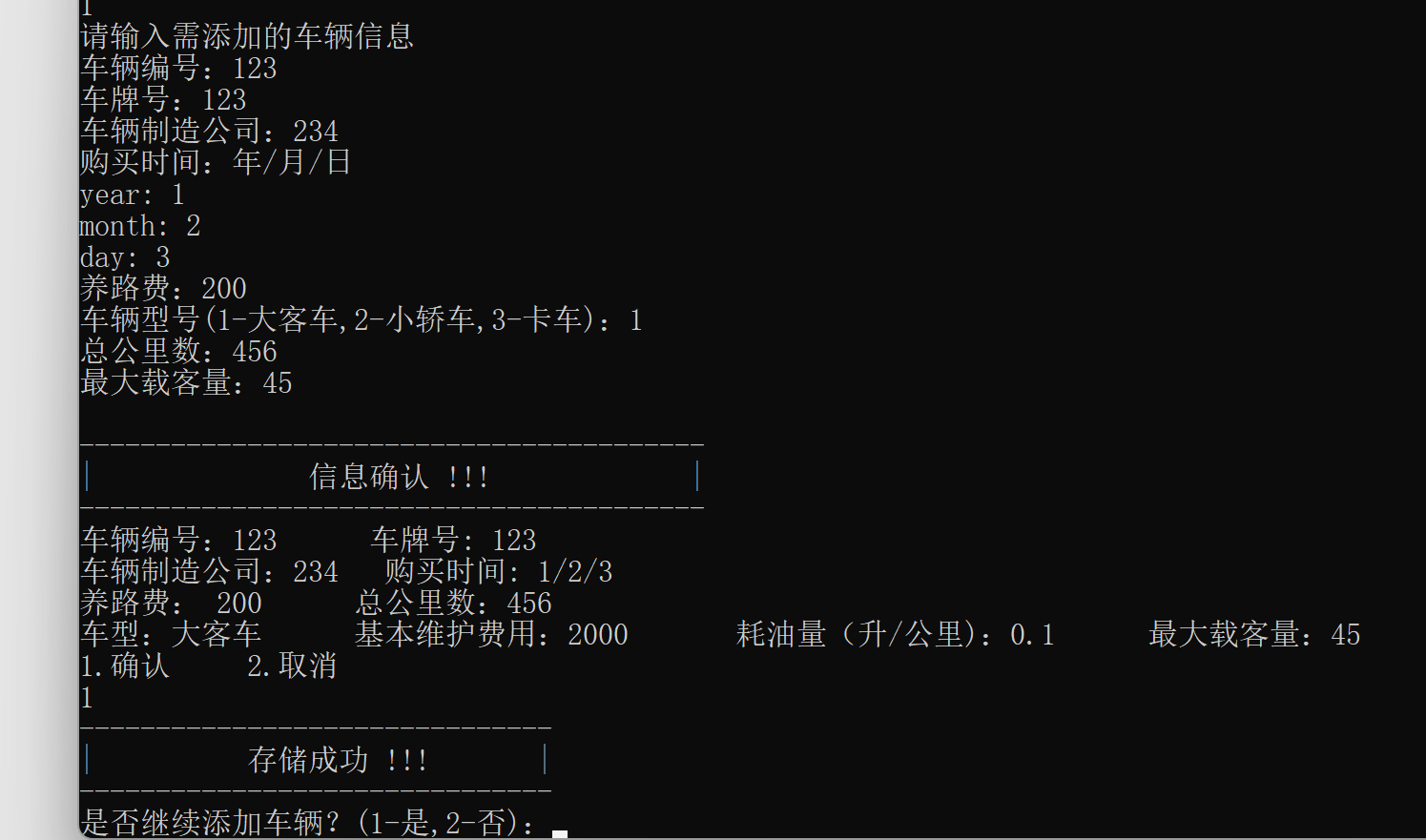


图20 添加车辆功能实现-确认存储

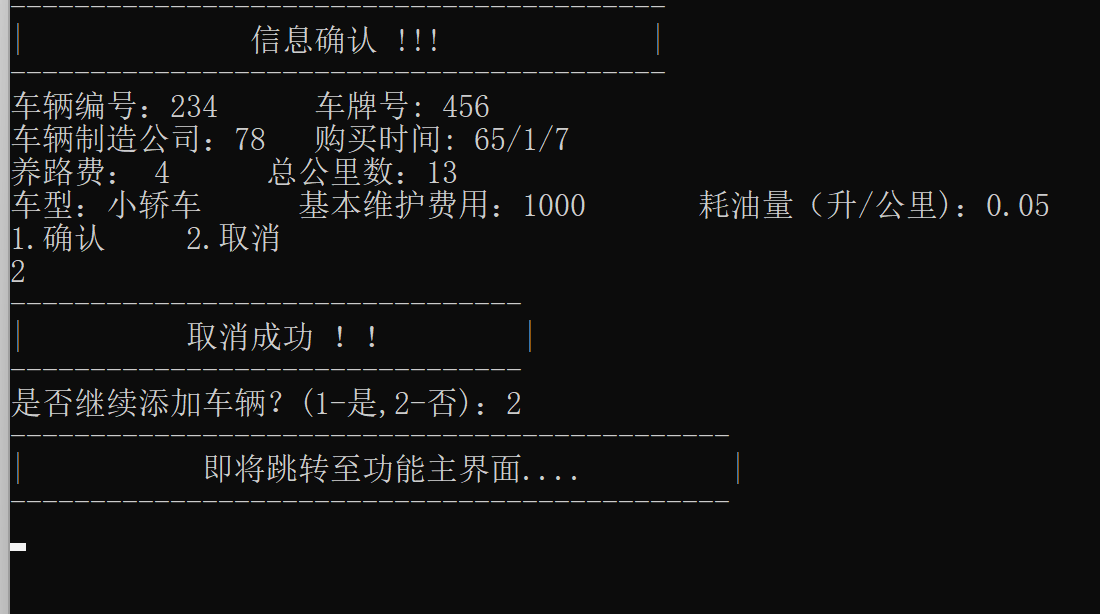


图21 添加车辆功能实现-取消存储

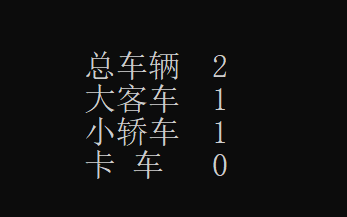


图22 添加车辆后车库中数量和类型的显示

结果与要求匹配，证明可以实现车辆的连续添加和车库中数量及种类的统计功能，且功能各方面正常。

2.查询车辆功能的实现



图23 查询功能的实现（1）

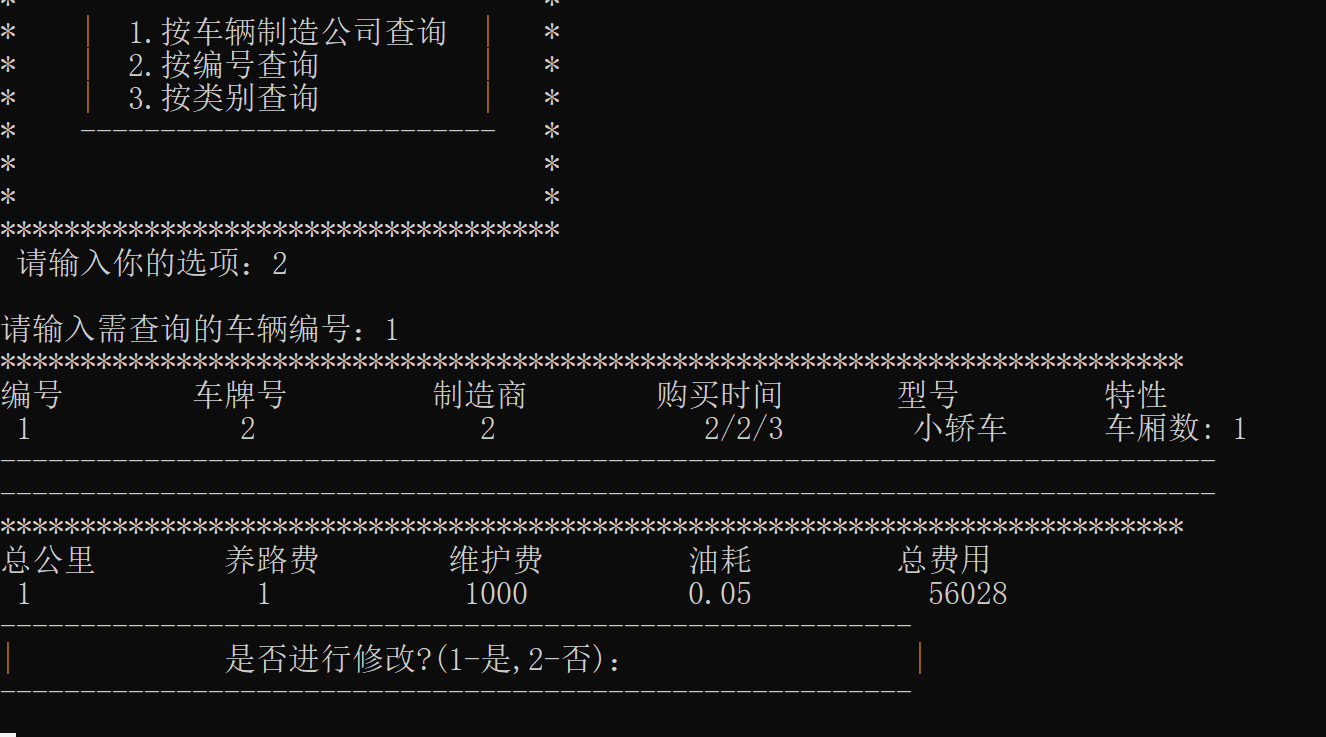


图24 查询功能的实现（2）



图25 查询功能的实现（3）



图26 未查询到相关车辆的情况（1）



图27 未查询到相关车辆的情况（2）

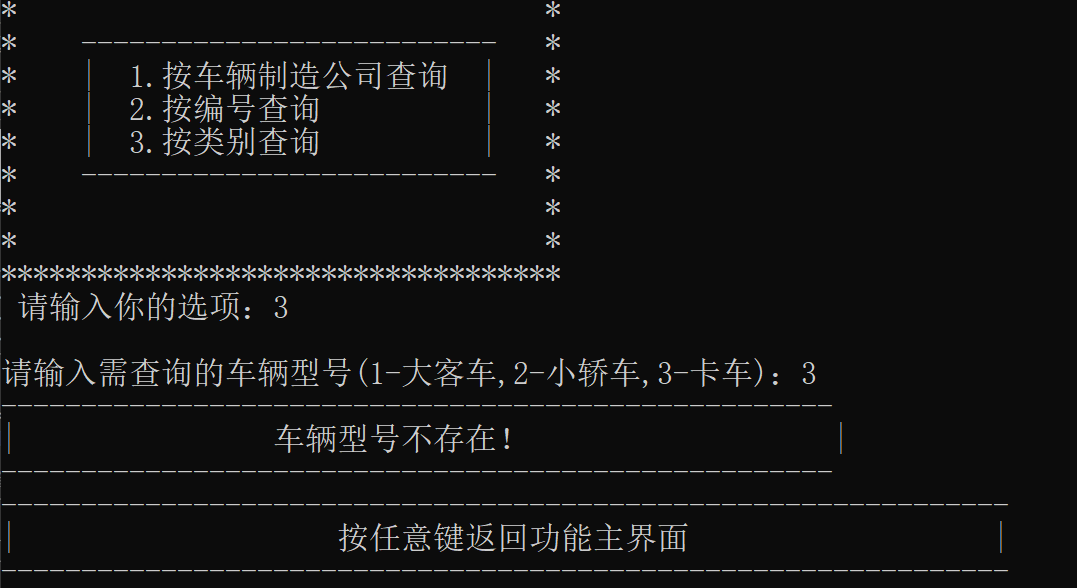


图28 未查询到相关车辆的情况（3）



图29 车辆库为空时调用查询车辆函数

结果与要求匹配，证明可以实现车辆的分类别查询和车库中数量及种类的统计功能，且功能各方面正常。

3.显示车辆库信息功能的实现

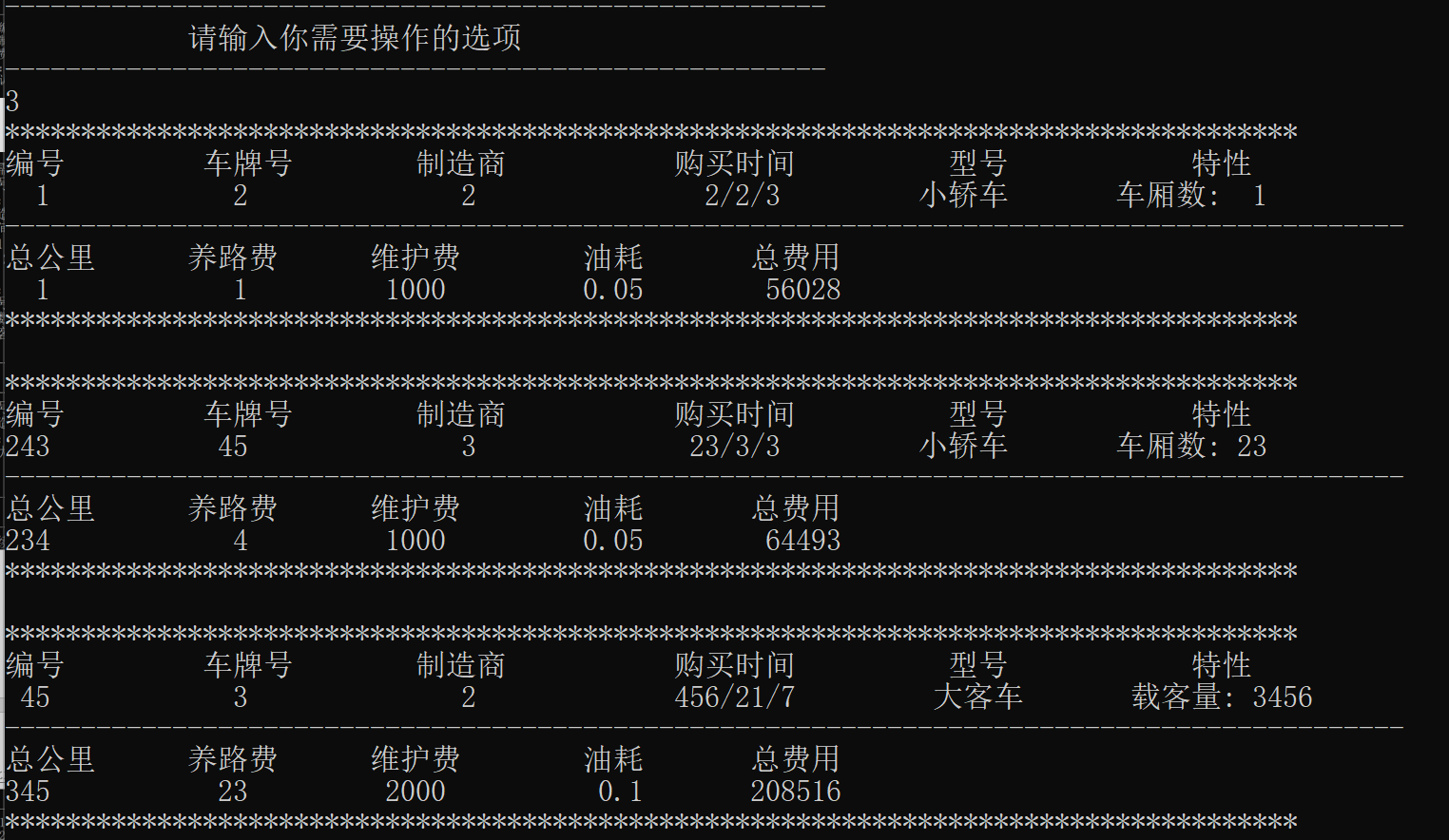


图30 车辆库信息显示的实现（1）



图31 车辆库为空时调用显示车辆函数

结果与要求匹配，可以正确的实现显示车辆库信息的功能。

4.编辑车辆信息功能的实现



图32 车辆库信息为空时调用编辑车辆函数

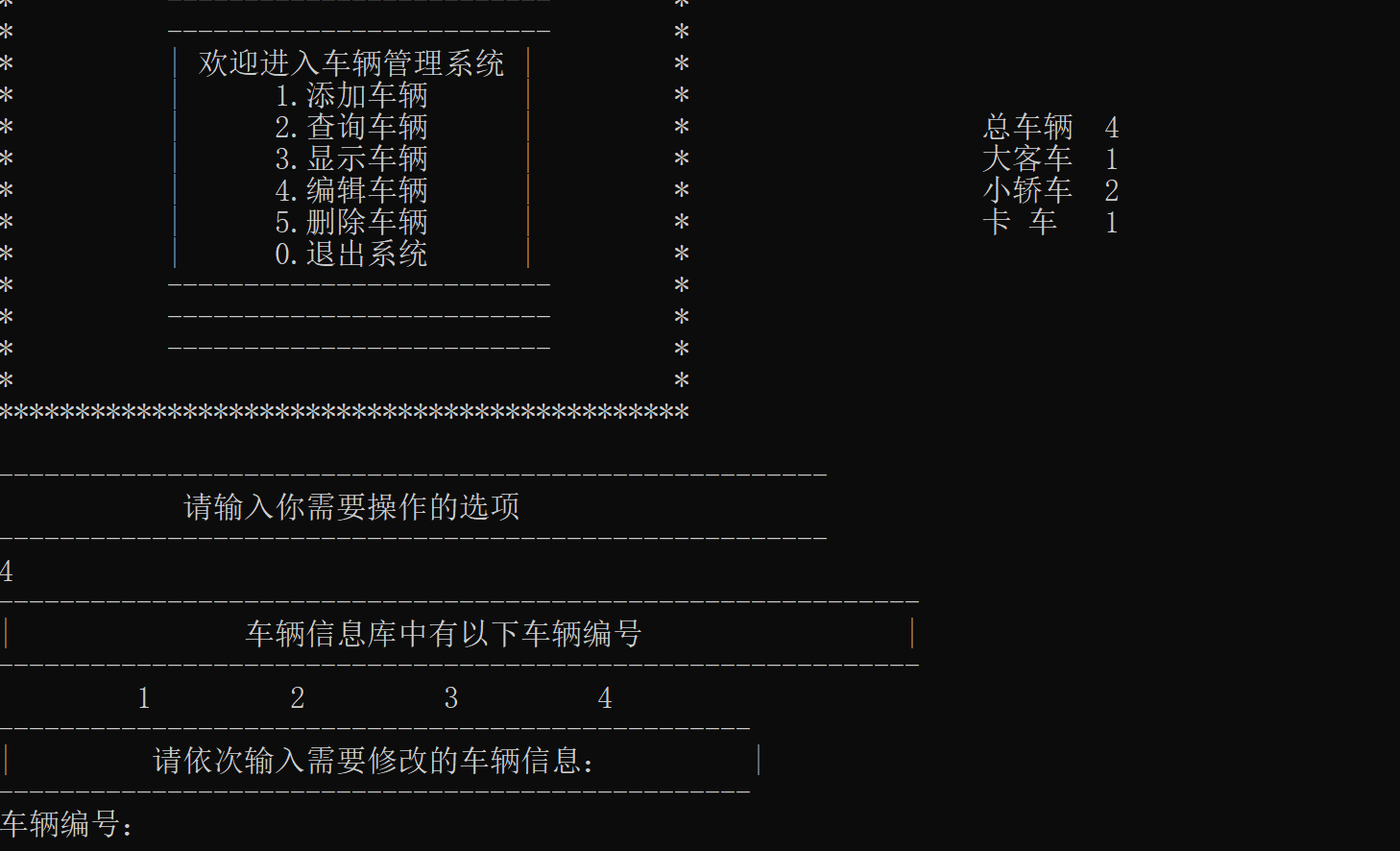


图33 编辑车辆信息功能的实现（1）

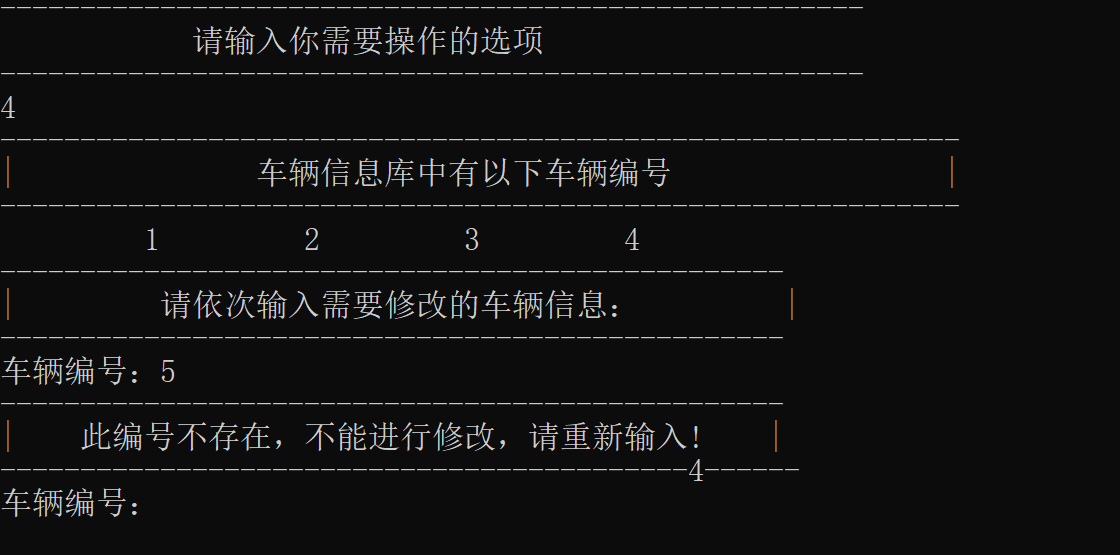


图34 编辑车辆信息功能的实现（2）

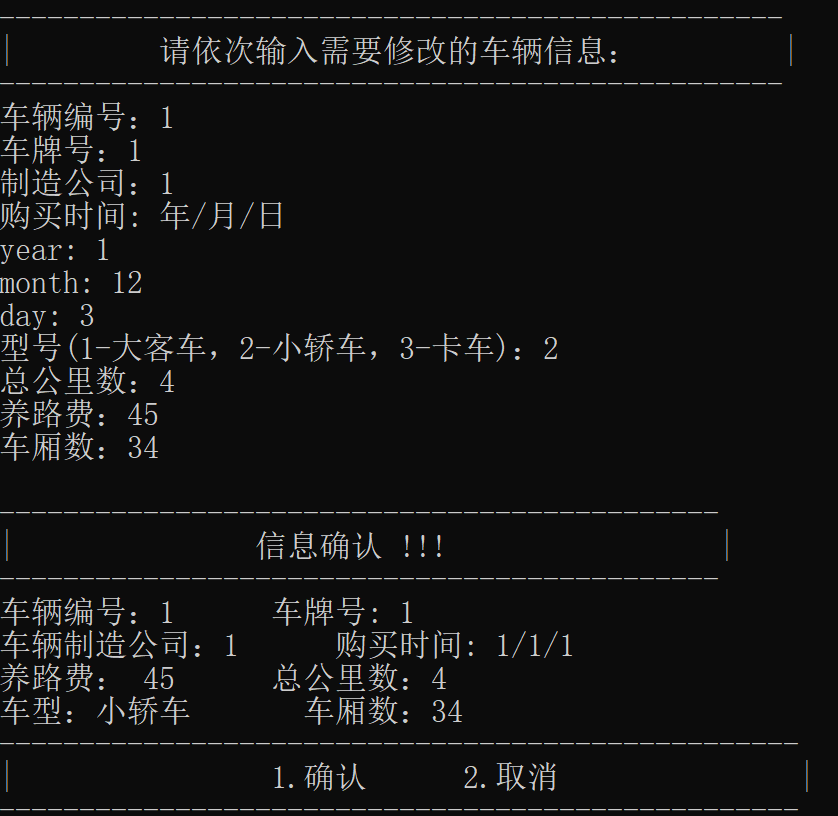


图35 编辑车辆信息功能的实现（3）

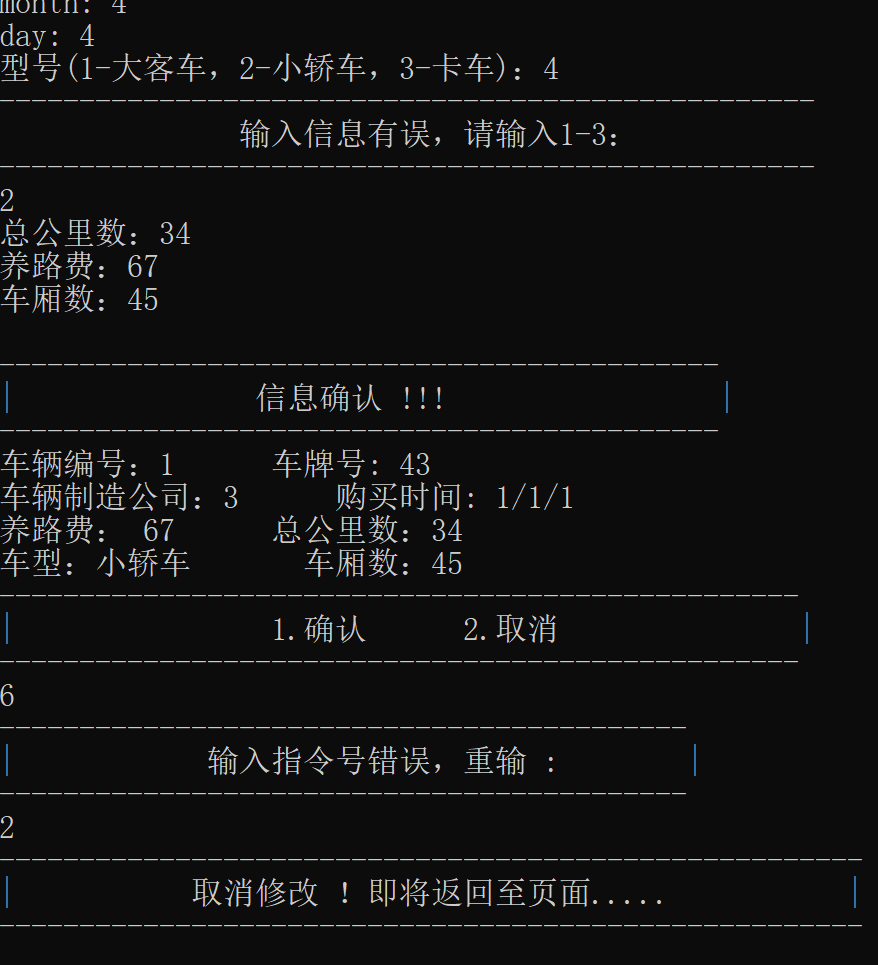


图36 编辑车辆信息功能实现-取消修改

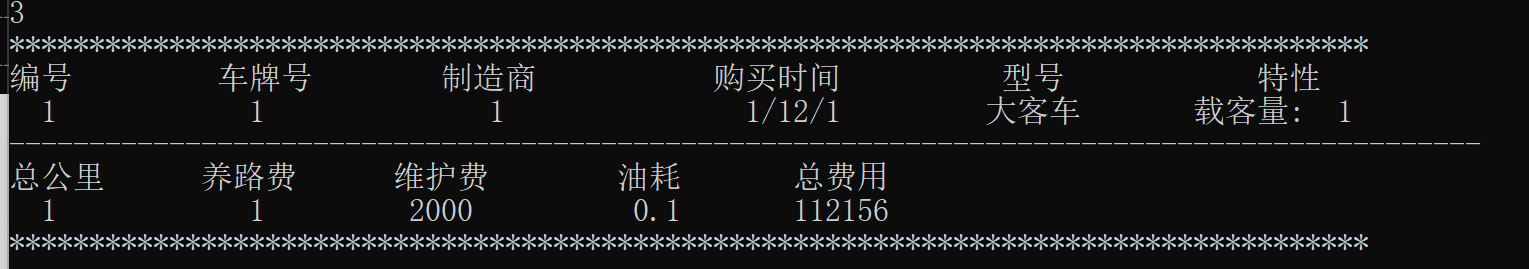


图37 取消后车辆信息并未改变

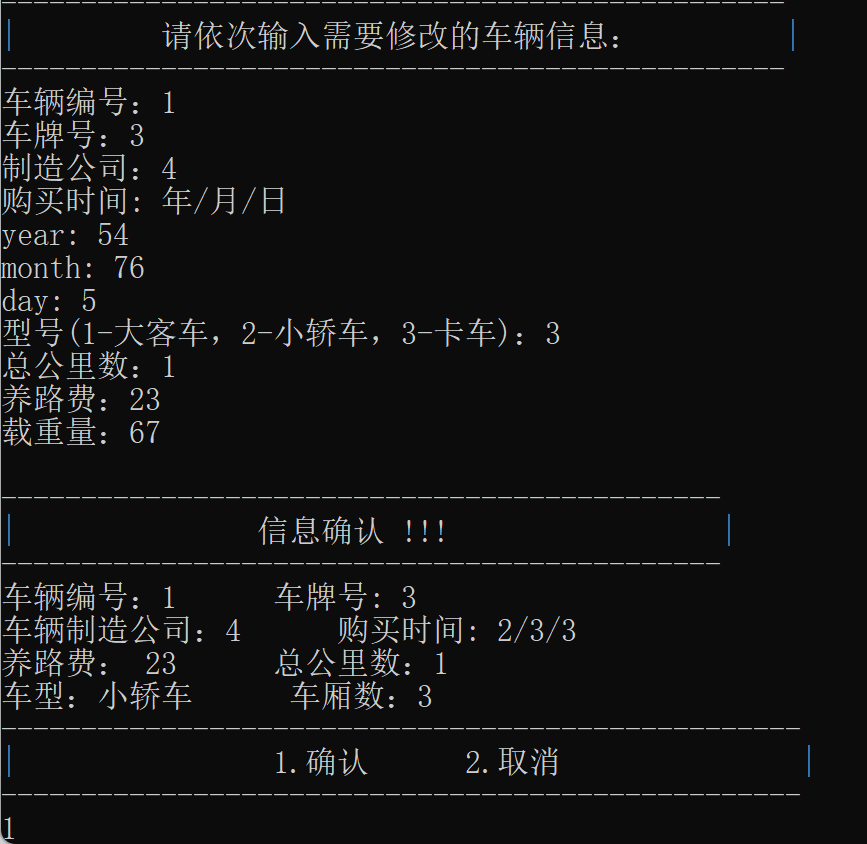


图38 确认修改车辆信息



图39 确认后车辆信息改变

5．删除车辆信息功能实现



图40 车辆库为空时无法删除车辆

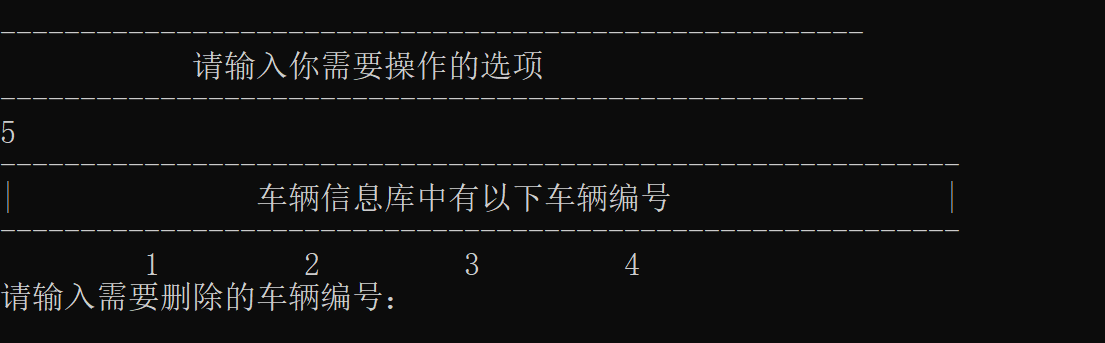


图41 在车库中显示所有车辆的编号



图42 车库中无此车辆编号时

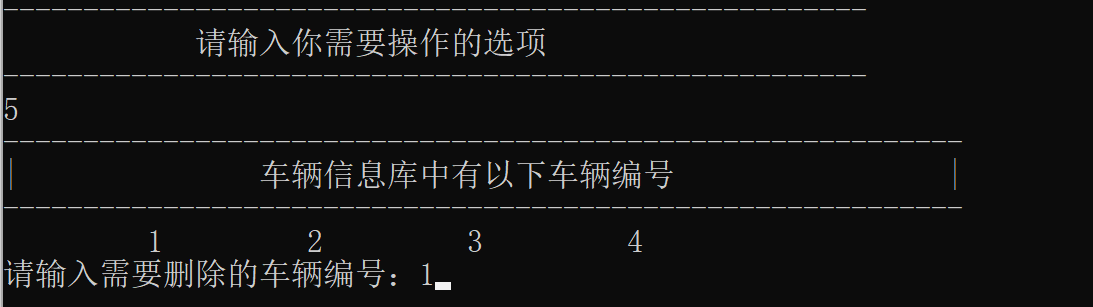


图43 删除1号车辆

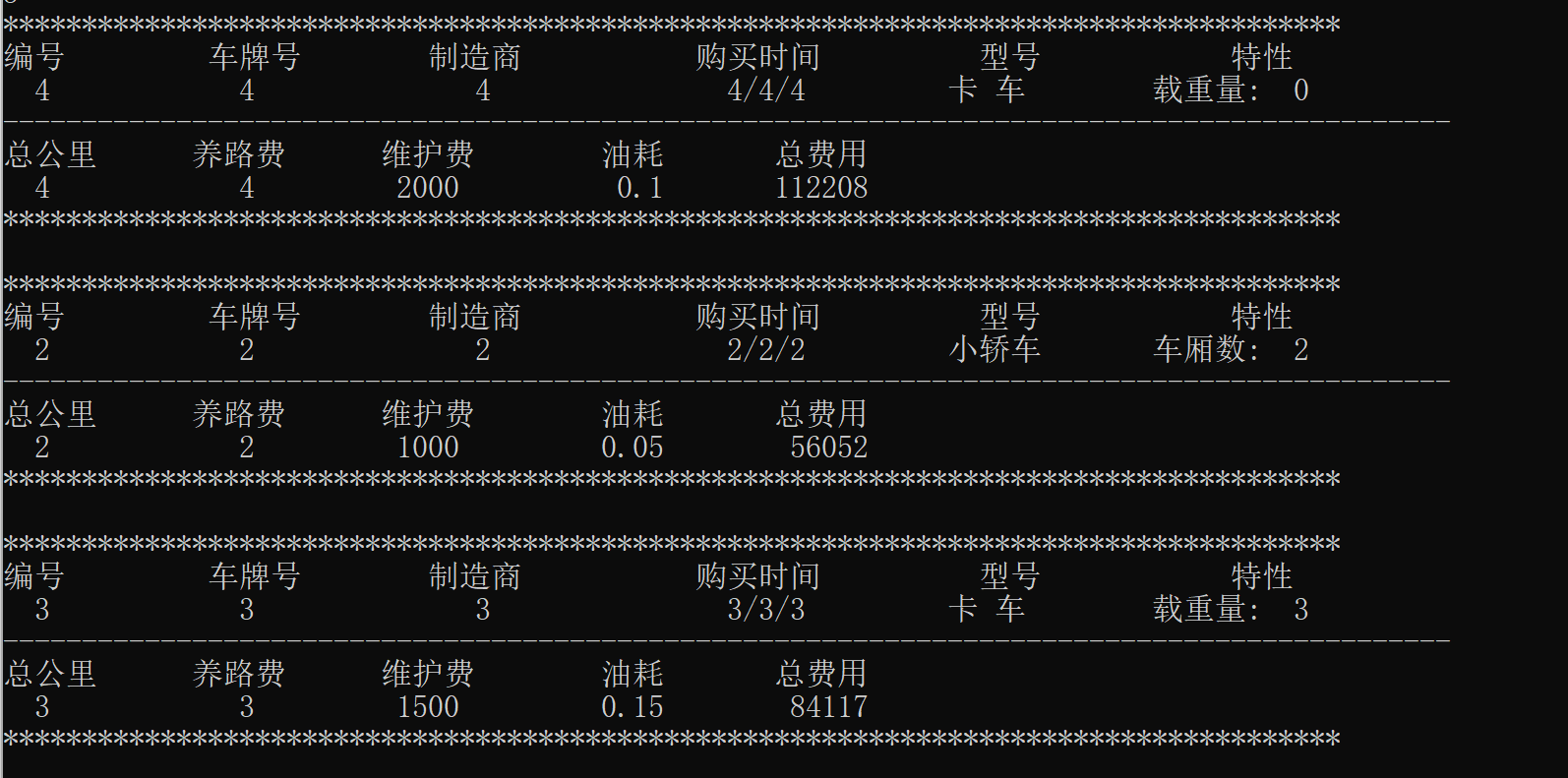


图44 删除车辆后车库的相关信息

实验结果与要求匹配，符合要求的所有功能，并在此基础上添加了显示所有车辆编号的功能，以此使得用户管理操作更方便，提高了管理效率。

## 6.3 关键源代码

#include<windows.h>

#include<iostream>

#include<fstream>

#include<stdlib.h>

#include<iomanip>

#include<conio.h>

#include<windows.h>

using namespace std;

#define N 200 //车辆最大存储量200

#define OilPrice 8.0 //油价8元每升

class Time

{

public:

Time()

{

year=0;

month=0;

day=0;

}

int year;

int month;

int day;

void set\_time();

void show\_time();

friend ostream& operator<<(ostream&output,Time&t);

};

ostream &operator<<(ostream& output,Time&t)

{

output<<t.year<<"/" <<t.month<<"/"<<t.day;

return output;

}

class temp

{

public:

int temp\_num; //临时编号

string temp\_pnum; //临时车牌号

string temp\_made; //临时制造公司

Time temp\_buytime; //临时购买时间（多少个月）

string temp\_type[3]={"大客车","小轿车","卡车"}; //临时型号(1-大客车,2-小轿车,3-卡车)

int temp\_types;

double temp\_km; //临时总公里数

int temp\_cost; //临时基本维护费用

int temp\_RoadFee; //临时养路费

double temp\_OilKm; //临时耗油量/公里

int people; //临时最大载客量（大客车）

int coach; //临时箱数（小轿车）

int weight;

};

void Time::show\_time()

{

cout<<year<<"/"<<month<<"/"<<day;

}

void Time::set\_time()

{

cout<<"year: "; //输入时间

cin>>year;

cout<<"month: ";

cin>>month;

cout<<"day: ";

cin>>day;

}

class Manager;

temp temp\_plus;

class Cars:public Time,public temp //汽车类

{

public:

int num; //编号

string pnum; //车牌号

string made; //制造公司

Time buytime; //购买时间（多少个月）

int type; //型号(1-大客车,2-小轿车,3-卡车) 公共变量

double km; //总公里数

int cost; //基本维护费用

int RoadFee; //养路费

double OilKm; //耗油量/公里

int AllCount; //累计总费用

friend Manager; //友元函数 Manager

Cars \*next;

};

class Manager //总的类

{

private:

int top; //记录当前车辆数量

int people[N]; //最大载客量（大客车）

int coach[N]; //箱数（小轿车）

int weight[N]; //载重量（卡车）

public:

Manager(){top = 0;}//车辆初始值为0 汽车总数

void add(); //添加车辆

void search(); //查询车辆

void show(); //显示车辆信息库

void edit(); //编辑功能

void delet(); //删除车辆

void sum(); //统计信息

void read(); //读出车辆信息

void write(); //车辆信息存盘

void jiemian(); //总界面

void lianbiao(); //链表初始化

void menu();

int count(int b,int c,int d,int a,int e=8,int f=7);

Cars car[N]; //汽车数

friend Cars;

Manager \*next;

};

// 开始 Manager的函数

void Manager::lianbiao()

{

for(int i=0;i<top-1;i++)

{

car[i].next=&car[i+1];

}

car[top-1].next=NULL;

}

int Manager::count(int b,int c,int d,int a,int e,int f)

{

return (e\*f\*a)+(b\*12+c)\*d; //计算总费用

}

void Manager::menu()

{

int a;

cout<<"----------------------------------------------------------"<<endl;

cout<<"| 确认是否退出系统 1-确定 2-取消 |"<<endl;

cout<<"----------------------------------------------------------"<<endl;

cin>>a;

while(!( a==1 || a==2 ))

{

cout<<"-------------------------------------------"<<endl;

cout<<"| 输入指令号错误，重输 : |"<<endl;

cout<<"-------------------------------------------"<<endl;

cin>>a;

}

if(a==1)

{

cout<<"--------------------------------------"<<endl;

cout<<"| 退出成功！！！ |"<<endl;

cout<<"--------------------------------------"<<endl;

Sleep(500);

system("cls") ; //清屏函数

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* | | \*"<<endl;

cout<<"\* | 欢迎下次使用 | \*"<<endl;

cout<<"\* | | \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

if(a==2)

{

cout<<"---------------------------------"<<endl;

cout<<"| 取消成功 !!! |"<<endl;

cout<<"---------------------------------"<<endl;

cout<<"-------------------------------------"<<endl;

cout<<"| 按任意键返回主界面: |"<<endl;

cout<<"--------------------------------------"<<endl;

char z=getch(); //从输出获得一个字符之后调用界面函数

system("cls") ; //清屏函数

jiemian();

}

}

void Manager::show()

{

Cars \*p,\*head;

head=&car[0];

p=head;

int j=0;

lianbiao();

if(top != 0) //top记录当前车辆数目

{

do

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"编号 车牌号 制造商 购买时间 型号 特性 "<<endl;

cout<<setw(3)<<p->num<<setw(13)<<p->pnum<<setw(15)<<p->made<<setw(16);

cout<<p->buytime;

if(p->type == 1 )

cout<<setw(15)<<"大客车"<<setw(15)<<"载客量: "<<setw(2)<<people[j]<<endl;

else if(p->type == 2 )

cout<<setw(15)<<"小轿车"<<setw(15)<<"车厢数: "<<setw(2)<<coach[j]<<endl;

else

cout<<setw(15)<<"卡 车 "<<setw(15)<<"载重量: "<<setw(2)<<weight[j]<<endl;

cout<<"--------------------------------------------------------------------------------------------"<<endl;

cout<<"总公里 养路费 维护费 油耗 总费用"<<endl;

cout<<setw(3)<<p->km<<setw(13)<<p->RoadFee<<setw(13)<<p->cost<<setw(13)<<p->OilKm<<setw(13)<<p->AllCount<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<endl;

p=p->next;

j++;

}

while(p!=NULL);

}

else

{

cout<<"----------------------------------"<<endl;

cout<<"| 车辆库为空！！！ |"<<endl;

cout<<"----------------------------------"<<endl;

cout<<endl;

}

cout<<"--------------------------------------"<<endl;

cout<<"| 按任意键返回主界面: |"<<endl;

cout<<"--------------------------------------"<<endl;

char z=getch(); //从输出获得一个字符之后调用界面函数

system("cls") ; //清屏函数

jiemian();

}

void Manager::add() //添加车辆 调用总的类之中的add函数

{

int a = 0, x, y,sm; // x代表车辆编号

while(1)

{

lianbiao();

if(top < N) //判断top当前已存车辆数小于汽车最大存储数N

{

cout<<"请输入需添加的车辆信息"<<endl;

cout<<"车辆编号：";

cin>>x;

for(int j=0; j<top;j++) //判断编号是否重复

{

if (car[j].num==x)

{

cout<<"---------------------------------------------------------"<<endl;

cout<<"| 编号重复,添加失败! 即将跳转至功能主界面 |"<<endl;

cout<<"---------------------------------------------------------"<<endl;

a++; //如果重复++，此时a不为0

break;

}

}

if(a==0) //此时编号不重复

{

temp\_plus.temp\_num=x;

car[top].num = x;

cout<<"车牌号：";

cin>>car[top].pnum;

temp\_plus.temp\_pnum=car[top].pnum;

cout<<"车辆制造公司：";

cin>>car[top].made;

temp\_plus.temp\_made=car[top].made;

cout<<"购买时间：年/月/日"<<endl;

car[top].buytime.set\_time();

temp\_plus.temp\_buytime.year=car[top].buytime.year;

temp\_plus.temp\_buytime.month=car[top].buytime.month;

temp\_plus.temp\_buytime.day=car[top].buytime.day;

cout<<"养路费：";

cin>>car[top].RoadFee;

temp\_plus.temp\_RoadFee=car[top].RoadFee;

cout<<"车辆型号(1-大客车,2-小轿车,3-卡车)：";

cin>>car[top].type;

while(!(car[top].type==1||car[top].type==2||car[top].type==3)) //判断输入指令是否为1-3

{

cout<<"输入指令号错误,重输：";

cin>>car[top].type;

}

cout<<"总公里数：";

cin>>car[top].km;

temp\_plus.temp\_km=car[top].km;

if( car[top].type==1) //此时是1-大客车，

{

cout<<"最大载客量：";

cin>>people[top];

temp\_plus.people=people[top];

car[top].cost = 2000; //基本维护费用

car[top].OilKm = 0.1; //耗油量/公里

}

else if( car[top].type == 2 ) //此时是2-小轿车

{

cout<<"车厢数：";

cin>>coach[top];

temp\_plus.coach=coach[top];

car[top].cost = 1000;

car[top].OilKm = 0.05;

}

else //此时是3-卡车

{

cout<<"载重量：";

cin>>weight[top];

temp\_plus.weight=weight[top];

car[top].cost = 1500;

car[top].OilKm = 0.15;

}

cout<<endl;

car[top].AllCount=count(car[top].km,car[top].buytime.year,car[top].buytime.month,car[top].cost,7,8);

//car[top].AllCount =(OilPrice \* car[top].OilKm \* car[top].km) +(((car[top].buytime.year\*12)+car[top].buytime.month)\*car[top].cost); //计算总费用

cout<<"-----------------------------------------"<<endl;

cout<<"| 信息确认 !!! |"<<endl;

cout<<"-----------------------------------------"<<endl;

cout<<"车辆编号："<<temp\_plus.temp\_num<<" ";

cout<<"车牌号: "<<temp\_plus.temp\_pnum<<" "<<endl;

cout<<"车辆制造公司："<<temp\_plus.temp\_made<<" ";

cout<<"购买时间: "<<temp\_plus.temp\_buytime<<endl;

cout<<"养路费： "<<temp\_plus.temp\_RoadFee<<endl;

cout<<"总公里数："<<temp\_plus.temp\_km;

cout<<"车型：";

if( car[top].type==1)

{

cout<<temp\_plus.temp\_type[0]<<" "<<"基本维护费用："<<car[top].cost<<endl;

cout<<"耗油量（升/公里)："<<car[top].OilKm<<" ";

cout<<"最大载客量："<<temp\_plus.people<<endl;

}

if( car[top].type==2)

{

cout<<temp\_plus.temp\_type[1]<<" "<<"基本维护费用："<<car[top].cost<<endl;

cout<<"耗油量（升/公里)："<<car[top].OilKm<<" ";

cout<<"车厢数："<<temp\_plus.coach<<endl;

}

if( car[top].type==3)

{

cout<<temp\_plus.temp\_type[2]<<" "<<"基本维护费用："<<car[top].cost<<endl;

cout<<"耗油量（升/公里)："<<car[top].OilKm<<" ";

cout<<"载重量："<<temp\_plus.weight<<endl;

}

cout<<"1.确认 2.取消 "<<endl;

cin>>sm;

while(!( sm==1 || sm==2 ))

{

cout<<"-------------------------------------------"<<endl;

cout<<"| 输入指令号错误，重输 : |"<<endl;

cout<<"-------------------------------------------"<<endl;

cin>>sm;

}

if(sm==1)

{

write(); //调用车辆信息存盘函数,在下面定义

cout<<"-------------------------------"<<endl;

cout<<"| 存储成功 !!! |"<<endl;

cout<<"-------------------------------"<<endl;

lianbiao();

top++; //车辆总数加一1

}

else if(sm==2)

{

cout<<"--------------------------------"<<endl;

cout<<"| 取消成功 ！！ |"<<endl;

cout<<"--------------------------------"<<endl;

car[top].type =0;

}

cout<<"是否继续添加车辆？(1-是,2-否)：";

cin>>y;

while(!( y==1 || y==2 ))

{

cout<<"-------------------------------------------"<<endl;

cout<<"| 输入指令号错误，重输 : |"<<endl;

cout<<"-------------------------------------------"<<endl;

cin>>y;

}

if(y == 2)

{

cout<<"---------------------------------------------"<<endl;

cout<<"| 即将跳转至功能主界面.... |"<<endl;

cout<<"---------------------------------------------"<<endl;

break;

}

}

else //此时编号重复

break;

}

else //top>N

{

cout<<"-----------------------------------------"<<endl;

cout<<"| 车辆信息库已满,不能再添加车辆 ! |"<<endl;

cout<<"| 即将跳转至功能主界面.... |"<<endl;

cout<<"-----------------------------------------"<<endl;

break;

}

}

Sleep(1000); //延迟1秒

system("cls") ;//清屏函数

jiemian(); //调用总界面函数

}

void Manager::search() //查询车辆

{

int choose,m,n,a=0;

char i,t[20];

Cars \*p,\*head;

head=&car[0];

lianbiao();

if(top == 0) //车库为空无法查询

{

cout<<"-----------------------------------------------------------"<<endl;

cout<<"| 车辆信息库为空！系统将返回功能界面.... |"<<endl;

cout<<"-----------------------------------------------------------"<<endl;

Sleep(600);

system("cls");

jiemian();

}

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* | 1.按车辆制造公司查询 | \*"<<endl;

cout<<"\* | 2.按编号查询 | \*"<<endl;

cout<<"\* | 3.按类别查询 | \*"<<endl;

cout<<"\* -------------------------- \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" 请输入你的选项：";

while(1)

{

cin>>i;

cout<<endl;

if(i=='1'||i=='2'||i=='3')

{

break;

}

else

cout<<"输入错误,重输：";

}

if(i=='1')

{

cout<<"请输入需查询车辆的制造公司：";

cin>>t;

}

else if(i =='2')

{

cout<<"请输入需查询的车辆编号：";

cin>>n;

}

else if(i == '3')

{

cout<<"请输入需查询的车辆型号(1-大客车,2-小轿车,3-卡车)：";

cin>>m;

}

for(p=head;p!=NULL;p=p->next)

{

if(p->made==t || p->num==n || p->type==m) //string字符串比较.

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"编号 车牌号 制造商 购买时间 型号 特性 "<<endl;

a++; //使a不为0

break;

}

}

for(p=head;p!=NULL;p=p->next)

{

int j=0;

if(p->made==t || p->num==n || p->type==m)

{

cout<<setw(2)<<p->num<<setw(14)<<p->pnum<<setw(15)<<p->made<<setw(14);

cout<<p->buytime;

if(p->type==1 )

cout<<setw(14)<<"大客车"<<setw(14)<<"载客量:"<<people[j]<<endl; //输出特性

else if(car[j].type == 2 )

cout<<setw(14)<<"小轿车"<<setw(14)<<"车厢数: "<<coach[j]<<endl;

else

cout<<setw(14)<<"卡 车"<<setw(14)<<"载重量:"<<weight[j]<<endl;

cout<<"----------------------------------------------------------------------------"<<endl;

cout<<"----------------------------------------------------------------------------"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"总公里 养路费 维护费 油耗 总费用"<<endl;

cout<<setw(2)<<p->km<<setw(15)<<p->RoadFee<<setw(16)<<p->cost;

cout<<setw(14)<<p->OilKm<<setw(16)<<p->AllCount<<endl;

}

j++;

}

if(a!=0)

{

cout<<"---------------------------------------------------------"<<endl;

cout<<"| 是否进行修改?(1-是,2-否)： |"<<endl;

cout<<"---------------------------------------------------------"<<endl;

cin>>choose;

if(choose == 1)

edit(); //调用编辑功能函数

}

else //上述a++的函数未成功，最上面1，2，3的查询结果没有

{

if(i=='1')

{

cout<<"--------------------------------------------------"<<endl;

cout<<"| 车辆制造公司不存在！ |"<<endl;

cout<<"--------------------------------------------------"<<endl;

}

if(i=='2')

{

cout<<"---------------------------------------------------"<<endl;

cout<<"| 车辆编号不存在！ |"<<endl;

cout<<"---------------------------------------------------"<<endl;

}

else if (i=='3')

{

cout<<"----------------------------------------------------"<<endl;

cout<<"| 车辆型号不存在！ |"<<endl;

cout<<"----------------------------------------------------"<<endl;

}

}

cout<<"---------------------------------------------------------------"<<endl;

cout<<"| 按任意键返回功能主界面 |"<<endl;

cout<<"---------------------------------------------------------------"<<endl;

char zz=getch(); //无回显的取一个字符

Sleep(700);

system("cls") ;//清屏函数

jiemian(); //调用界面函数

}

void Manager::edit() //编辑车辆，既修改车辆

{

int i, j, a = 0,sm,q;

lianbiao();

Cars \*p,\*head;

head=&car[0];

if(top == 0) //车库为空无法编辑

{

cout<<"---------------------------------------------------------------"<<endl;

cout<<"| 车辆信息库为空，按任意键返回功能主界面 |"<<endl;

cout<<"---------------------------------------------------------------"<<endl;

char zz=getch(); //无回显的取一个字符

Sleep(500);

system("cls") ;//清屏函数

jiemian(); //调用界面函数

}

cout<<"------------------------------------------------------------"<<endl;

cout<<"| 车辆信息库中有以下车辆编号 |"<<endl;

cout<<"------------------------------------------------------------"<<endl;

for(int i=0;i<top;i++)

{

cout<<setw(10)<<car[i].num;

}

cout<<endl;

cout<<"-------------------------------------------------"<<endl;

cout<<"| 请依次输入需要修改的车辆信息： |"<<endl;

cout<<"-------------------------------------------------"<<endl;

while(1)

{

cout<<"车辆编号：";

cin>>i;

for(j = 0; j < top; j++)

{

if(car[j].num == i) //车库有此编号的车

{

cout<<"车牌号：";

cin>>temp\_plus.temp\_pnum;

cout<<"制造公司：";

cin>>temp\_plus.temp\_made;

cout<<"购买时间: 年/月/日"<<endl;

car[j].set\_time();

cout<<"型号(1-大客车，2-小轿车，3-卡车)：";

cin>>temp\_plus.temp\_types;

while(!(temp\_plus.temp\_types==1||temp\_plus.temp\_types==2||temp\_plus.temp\_types==3)) //判断是否有误

{

cout<<"---------------------------------------------------"<<endl;

cout<<" 输入信息有误，请输入1-3： "<<endl;

cout<<"---------------------------------------------------"<<endl;

cin>>temp\_plus.temp\_types;

}

cout<<"总公里数：";

cin>>temp\_plus.temp\_km;

cout<<"养路费：";

cin>>temp\_plus.temp\_RoadFee;

if( temp\_plus.temp\_types == 1 ) //此时是1-大客车

{

cout<<"最大载客量：";

cin>>temp\_plus.people;

temp\_plus.temp\_cost = 2000;

temp\_plus.temp\_OilKm = 0.1;

}

else if(temp\_plus.temp\_types == 2 ) //此时是2-小轿车

{

cout<<"车厢数：";

cin>>temp\_plus.coach;

temp\_plus.temp\_cost = 1000;

temp\_plus.temp\_OilKm = 0.05;

}

else //此时是3-卡车

{

cout<<"载重量：";

cin>>temp\_plus.weight;

temp\_plus.temp\_cost = 1500;

temp\_plus.temp\_OilKm = 0.15;

}

cout<<endl;

cout<<"---------------------------------------------"<<endl;

cout<<"| 信息确认 !!! |"<<endl;

cout<<"---------------------------------------------"<<endl;

cout<<"车辆编号："<<temp\_plus.temp\_num<<" ";

cout<<"车牌号: "<<temp\_plus.temp\_pnum<<" "<<endl;

cout<<"车辆制造公司："<<temp\_plus.temp\_made<<" ";

cout<<"购买时间: "<<temp\_plus.temp\_buytime<<endl;

cout<<"养路费： "<<temp\_plus.temp\_RoadFee<<" "<<"总公里数："<<temp\_plus.temp\_km<<endl;

cout<<"车型：";

if( temp\_plus.temp\_types==1)

{

cout<<temp\_plus.temp\_type[0]<<" ";

cout<<"最大载客量："<<temp\_plus.people<<endl;

}

if( temp\_plus.temp\_types==2)

{

cout<<temp\_plus.temp\_type[1]<<" ";

cout<<"车厢数："<<temp\_plus.coach<<endl;

}

if( temp\_plus.temp\_types==3)

{

cout<<temp\_plus.temp\_type[2]<<" ";

cout<<"载重量："<<temp\_plus.weight<<endl;

}

cout<<"--------------------------------------------------"<<endl;

cout<<"| 1.确认 2.取消 |"<<endl;

cout<<"--------------------------------------------------"<<endl;

cin>>sm;

while(!( sm==1 || sm==2 ))

{

cout<<"-------------------------------------------"<<endl;

cout<<"| 输入指令号错误，重输 : |"<<endl;

cout<<"-------------------------------------------"<<endl;

cin>>sm;

}

if(sm==1)

{

car[j].pnum=temp\_plus.temp\_pnum;

car[j].made=temp\_plus.temp\_made;

car[j].buytime.year=temp\_plus.temp\_buytime.year;

car[j].buytime.month=temp\_plus.temp\_buytime.month;

car[j].buytime.day=temp\_plus.temp\_buytime.day;

car[j].RoadFee=temp\_plus.temp\_RoadFee;

car[j].km=temp\_plus.temp\_km;

car[j].type=temp\_plus.temp\_types;

people[j]=temp\_plus.people;

coach[j]=temp\_plus.coach;

weight[j]=temp\_plus.weight;

car[j].cost=temp\_plus.temp\_cost;

car[j].OilKm=temp\_plus.temp\_OilKm;

write(); //调用存盘函数

a++; //成功后加一

}

else if(sm==2)

{

cout<<"------------------------------------------------------"<<endl;

cout<<"| 取消修改 ！即将返回至页面..... |"<<endl;

cout<<"------------------------------------------------------"<<endl;

Sleep(700);

system("cls") ; //清屏函数

jiemian();

}

}

}

if(a == 0) //未存成功，说明有错误

{

cout<<"-------------------------------------------------"<<endl;

cout<<"| 此编号不存在，不能进行修改，请重新输入! |"<<endl;

cout<<"-------------------------------------------4------"<<endl;

}

else

{

cout<<"--------------------------------------------------------"<<endl;

cout<<"| 修改成功！系统将返回主界面.... |"<<endl;

cout<<"--------------------------------------------------------"<<endl;

Sleep(700);

system("cls") ; //清屏函数

break; //跳出while循环

}

}

Sleep(700); //延迟0.7秒

jiemian();

}

void Manager::delet() //删除车辆

{

int i, a = 0; //a用来判断循环是否成功

int j;

Cars \*p,\*head;

head=&car[0];

while(1)

{

if(top == 0) //车库为空无法删除

{

cout<<"-----------------------------------------------------------"<<endl;

cout<<"| 车辆信息库为空！系统将返回功能界面.... |"<<endl;

cout<<"-----------------------------------------------------------"<<endl;

break; //跳出while循环

}

else //不为空

{

cout<<"------------------------------------------------------------"<<endl;

cout<<"| 车辆信息库中有以下车辆编号 |"<<endl;

cout<<"------------------------------------------------------------"<<endl;

for(int i=0;i<top;i++)

{

cout<<setw(10)<<car[i].num;

}

cout<<endl;

cout<<"请输入需要删除的车辆编号：";

cin>>i;

for( j = 0; j <top; j++) //for循环用来删除所有存盘的信息

{

if(car[j].num == i) //车辆编号存在

{

if(top==1)

car[0].type=0; //就一辆车，使其变为0

else //top不等于1

{

if( car[top - 1].type == 1 ) //判断出此时car数组里储存在最后一组的是1-大客车

{

people[j] = people[j+1]; //people载客量

car[j].cost = 2000;

car[j].OilKm = 0.1;

}

else if(car[top - 1].type == 2 ) //判断出此时car数组里储存在最后一组的是2-小轿车

{

coach[j] = coach[top - 1];

car[j].cost = 1000;

car[j].OilKm = 0.05;

}

else if(car[top - 1].type == 3 ) //判断出此时car数组里储存在最后一组的是3-卡车

{

weight[j] = weight[j+1];

car[j].cost = 1500;

car[j].OilKm = 0.15;

}

car[j].num = car[top-1].num; //使所有的数据均减一

car[j].pnum=car[top-1].pnum;

car[j].made=car[top-1].made;

car[j].buytime = car[top-1].buytime;

car[j].type = car[top-1].type;

car[j].km = car[top-1].km;

car[j].RoadFee = car[top-1].RoadFee;

car[j].AllCount = car[top-1].AllCount;

car[j].type=0;

}

//top--;

a++;

write(); //存盘函数

lianbiao();

}

}

if(a == 0)

{

cout<<"----------------------------------------------------------"<<endl;

cout<<"| 此编号不存在，不能进行删除，请重新输入！ |"<<endl;

cout<<"----------------------------------------------------------"<<endl;

}

else

{

cout<<"------------------------------------------------------------"<<endl;

cout<<"| 删除成功！系统将返回功能界面.... |"<<endl;

cout<<"------------------------------------------------------------"<<endl;

Sleep(800); //延迟0.8秒。

system("cls") ; //清屏函数

break;

}

}

}

Sleep(700); //延迟0.7秒

system("cls") ; //清屏函数

jiemian();

}

void Manager::write() //车辆信息存盘

{

ofstream write;

write.open("d:\\b.txt",ios::out); //以输出方式打开文件

if(write)

{

for(int i = 0; i <=top; i++)

{

write<<endl;

write<<car[i].num<<" "<<car[i].pnum<<" "<<car[i].made<<" ";

write<<car[i].buytime.year<<" "<<car[i].buytime.month<<" "<<car[i].buytime.day;

write<<" "<<car[i].type<<" ";

if(car[i].type == 1)

write<<people[i];//存入 载客量

else if(car[i].type == 2)

write<<coach[i];//存入 车厢数

else

write<<weight[i];//存入 载重量

write<<" "<<car[i].km<<" "<<car[i].RoadFee<<" "<<car[i].cost<<" "<<car[i].OilKm<<" "<<car[i].AllCount;

}

}

}

void Manager::read() ///读出车辆信息

{

while(1)

{

ifstream read("d:\\b.txt",ios::in);

if(read)

{

while(!read.eof()) //文件不结束，还有内容

{

read>>car[top].num>>car[top].pnum>>car[top].made;

read>>car[top].buytime.year>>car[top].buytime.month>>car[top].buytime.day>>car[top].type;

if(car[top].type == 1)

read>>people[top];//读取 载客量

else if(car[top].type == 2)

read>>coach[top];//读取 车厢数

else

read>>weight[top];//读取 载重量

read>>car[top].km>>car[top].RoadFee>>car[top].cost>>car[top].OilKm>>car[top].AllCount;

if(car[top].type==1||car[top].type==2||car[top].type==3)

top++;

}

read.close(); //关闭文件

break;

}

else

{

ofstream outfile("d:\\a.txt");

outfile.close();

}

}

}

void Manager::jiemian()

{ int c1=0,c2=0,c3=0;

for(int j = 0; j <= top; j++)

{

if( car[j].type == 1 ) c1++; //统计大客车的数量

else if(car[j].type == 2) c2++; //统计小轿车的数量

else if(car[j].type==3) c3++; //统计卡车的数量

}

top=c1+c2+c3;

lianbiao();

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\* ------------------------- \*"<<endl;

cout<<"\* ------------------------- \*"<<endl;

cout<<"\* ------------------------- \*"<<endl;

cout<<"\* | 欢迎进入车辆管理系统 | \*"<<endl;

cout<<"\* | 1.添加车辆 | \*"<<endl;

cout<<"\* | 2.查询车辆 | \* \t\t总车辆\t"<<top<<endl;

cout<<"\* | 3.显示车辆 | \* \t\t大客车\t"<<c1<<endl;

cout<<"\* | 4.编辑车辆 | \* \t\t小轿车\t"<<c2<<endl;

cout<<"\* | 5.删除车辆 | \* \t\t卡 车 \t"<<c3<<endl;

cout<<"\* | 0.退出系统 | \*"<<endl;

cout<<"\* ------------------------- \*"<<endl;

cout<<"\* ------------------------- \*"<<endl;

cout<<"\* ------------------------- \*"<<endl;

cout<<"\* \*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<endl;

cout<<"------------------------------------------------------"<<endl;

cout<<" 请输入你需要操作的选项 "<<endl;

cout<<"------------------------------------------------------"<<endl;

char i;

cin>>i;

while(!(i=='1'||i=='2'||i=='3'||i=='4'||i=='5'||i=='0'))

{

cout<<"-----------------------------------------------"<<endl;

cout<<"| 输入错误,请重新输入 |"<<endl;

cout<<"-----------------------------------------------"<<endl;

cin>>i;

}

switch(i)

{

case '1':add();

break;

case '2':search();

break;

case '3':show();

break;

case '4':edit();

break;

case '5':delet();

break;

case '0':menu();

return;

}

}

int main()

{

int temp\_num; //编号

char temp\_pnum[20]; //车牌号

char temp\_made[20]; //制造公司

Time temp\_buytime; //购买时间（多少个月）

int temp\_type; //型号(1-大客车,2-小轿车,3-卡车) 公共变量

double temp\_km; //总公里数

int temp\_cost; //基本维护费用

int iemp\_RoadFee; //养路费

double temp\_OilKm; //耗油量/公里

double temp\_AllCount; //累计总费用

Cars car[N];

Manager a;

a.jiemian();

return 0;

}

return 0;