

重庆师范大学

《数据结构课程设计》

课程名称： 数据结构

题 目： 停车场管理系统

学 院： 计算机与信息科学学院

专业年级： 计算机科学技术 2019 级

小组组长： 犹晓婷

小组成员： 王艳、蒋沁洁

指导教师： 张万里 职称： 讲师

2021 年 12 月 24 日

目录

一、	设计目的/需求分析	- 2 -
1.	设计目的	- 2 -
2.	需求分析	- 2 -
2.1	系统开发原因	- 2 -
2.2	系统开发目标	- 3 -
2.3	系统开发任务	- 3 -
2.4	系统总体结构	- 3 -
二、	问题描述	- 4 -
三、	基本要求	- 5 -
四、	概要设计	- 5 -
1)	算法思路	- 5 -
2)	工作分配	- 5 -
五、	程序模块	- 5 -
六、	主程序	- 10 -
七、	运行截图	- 17 -
八、	总结	- 21 -

停车场管理系统

一、设计目的/需求分析

1. 设计目的

了解栈和队列的逻辑结构和基本操作，实现栈和队列的基本功能。通过实验进一步理解栈和队列的逻辑结构和存储结构，进一步提高使用理论知识指导解决实际问题的能力。为大家解决一些生活中的实际问题，在这个过程中，自我设计的能力也在不断地提高。此次程序设计综合运用所学数据结构以及 C 语言的知识解决实际问题，将课堂的书本知识有效的在程序中体现出来，使我们更理解了 C 语言及数据结构的功能之强大，进一步让学生对面向对象的方法以及数据结构的编程思想有了较好了解和认识。通过课程设计，加深对《数据结构》这一课程所学内容的进一步理解与巩固，加深对结构化设计思想的理解，能对系统功能进行分析，并设计出合理的模块化结构，提高程序开发功能，培养分析问题、解决实际问题的能力，能运用合理的控制流程编写清晰高效的程序。

设计停车场管理系统，设计自动计算费用，避免人工出现的恶意收费，优化设施的使用，创造新的收入来源，传统停车场采用人工管理办法，不能适应社会发展的需要，一方面劳动强度大、效率低；另一方面主要弊端就是财务上造成很大的漏洞，停车场管理系统是现代化停车场车辆收费及设备自动化管理的统称，该系统将电子计算机结合起来，通过电脑管理可实施车辆出入管理、自动存储数据等功能，并且可实现脱机运行，在电脑文件出错时仍可以保证车辆的正常出入，此系统是现代化停车场车辆管理的理想系统。

2. 需求分析

2.1 系统开发原因

随着经济的发展和人民生活水平的提高，买车成为人们的奋斗目标之一，当然已经有很多人买起了车。可是随着车辆的增多，车辆的停泊问题也随之出现。这就使得各停车场需要更加先进、更加完善的车辆管理系统，为车主带来方便，使停车场的管理系统化。因此开发了停车场管理系统。

开发停车场管理系统，使管理员更加方便的众观全场停车信息，以及系统自动计费，避免人工的恶意收费以及算法失误。同时，此系统也全面体现了自动化的时代，也象征着社会正在朝人工智能发展，逐步进入自动化社会后，方便物业管理经营，采用现代化管理手段，对停车场内车车辆有效管理，包括车辆身份判断，出入控制，车位检索，车位引导，时间计

算等有效操作，提高物业管理经营效率。

2.2 系统开发目标

2.2.1 停车场内车辆信息的及时汇总，随时了解停车场车位的使用状况

2.2.2 自动化统计车辆信息，提高工作效率和工作质量

2.2.3 以停车场内的全部车辆信息为基础，动态分配停车位，尽量达到车位的最高利用率

2.2.4 停车费用结算由电脑来完成，解决用户所担心的乱收费问题

2.2.5 采用电脑自动化分配位置，减少人工耗时，增加办公效率

2.2.6 可一步到位查看停车场及便道状态及剩余量，减少人工造成的误差及失误

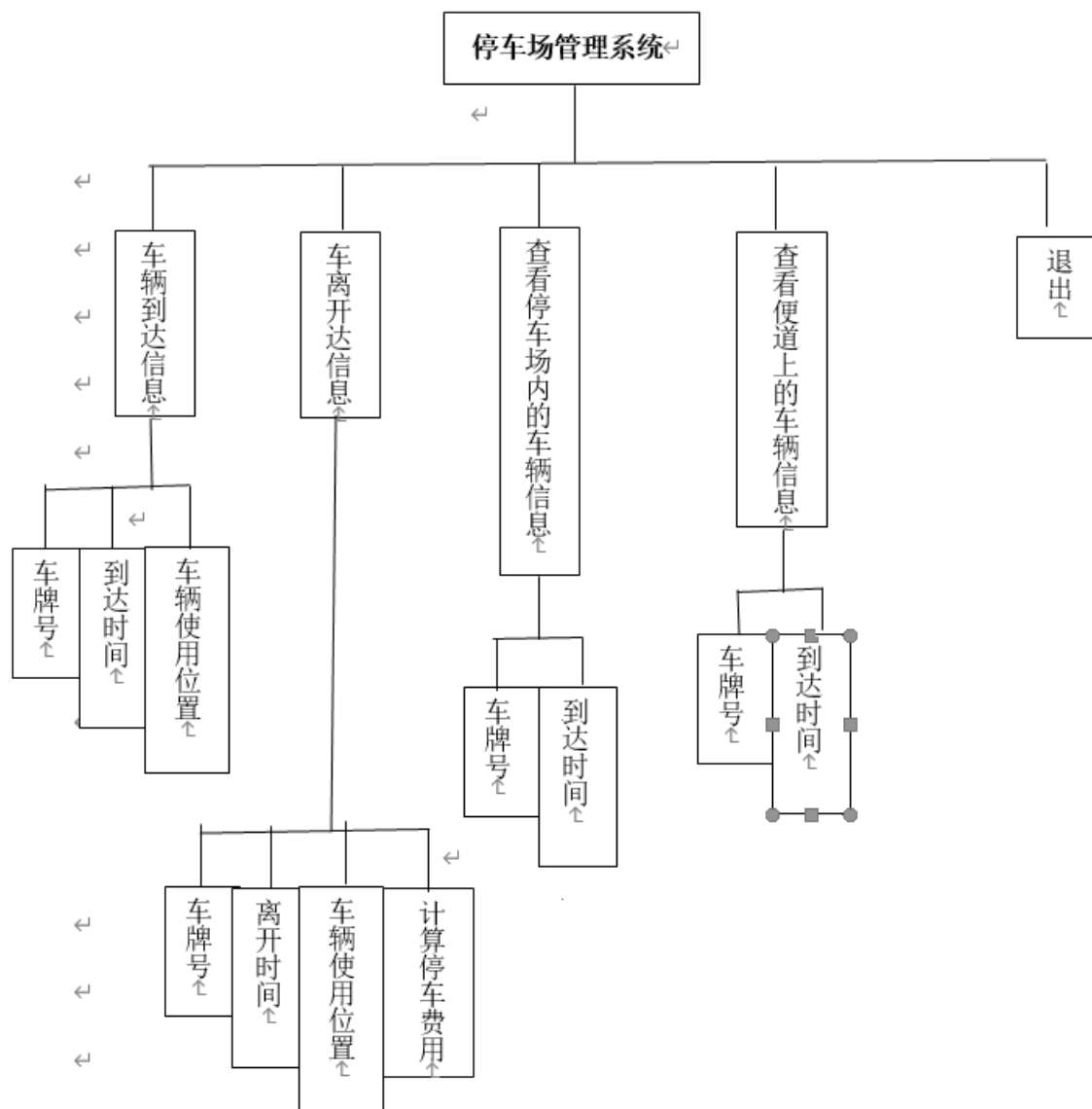
2.2.7 准确时间到达，让用户更安心及更理性的停车

2.2.8 车辆离开时自动算费，并且及时给出费用，避免了人工算费时造成的耗时，提高大家用户效率

2.3 系统开发任务

实现停车场的现代化管理，当用户进入停车场时，输入车辆信息（包括车牌号、进入停车时间年、月、日、时），系统自动分配动态车位，离开时输入车辆信息（包括车牌号、离开停车场时间年、月、日、时），离开时系统自动结算停车费用，当用户在进入停车场前，可查看停车场车位的使用状况（包括停车场和便道实用状况），停车场包含300个停车位，便道设置无限制，当用户进入停车场时，若停车场内还有余量可停车，即系统自动分配停车位置，若停车场内已停满，及系统自动将此车辆放置便道，当用户离开停车场时，可选择退出系统等，解决车辆停放难，乱收费等问题。

2.4 系统总体结构



二、 问题描述

设停车厂只有一个可停放几辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内按车辆到达的先后顺序依次排列，若车场内已停满几辆汽车，则后来的汽车只能在门外的便道上等候，便道上无限制，一旦停车场内有车开走，则排在便道上的第一辆车即可进入，停在刚离开车辆的位置，当停车场内某辆车要离开时，由于停车场是狭长的通道，在它之后进入的车辆必须先退出车场为它让路，待该辆车开出大门后，为它让路的车辆再按原次序进入车场。因此此系统的设计功能还带提升。

三、 基本要求

(1)停车场管理系统用于用户停车自动计算费用,用户进入停车场输入车辆信息(车牌),输入车辆到达时间,离开时输入车辆离开时间,离开时自动计算费用,从当前地点出发到其他所有地点的路线;管理员页面,可以增加删除地点的信息,以及地点间距离的信息。

(2)系统能实现的操作功能如下:

- ① 可录入车辆信息:车牌号,到达时间
- ② 计算停车费用:从进入停车位的那一刻开始计算时间,停车费用为每小时5元
- ③ 便道车辆信息:当停车场满位时,后续进入停车场的车辆应进入便道并为其录入车辆信息;当停车场有空位时,便道车辆应进入停车位(注意:车辆进入便道不计算费用)

四、 概要设计

1) 算法思路

- 1. 构建两个顺序栈,一个模拟停车场,一个模拟临时栈。构建一个链队模拟便道。
- 2. 当有车辆进入停车场时,若停车场容量未满,则该车辆信息入停车场栈,否则入便道链队。
- 3. 当停车场内车辆需要离开时,停车场栈内数据依次出栈并存入临时栈,若出栈的车牌号与需要离开的车辆的车牌号相同时,该车辆信息不入临时栈,临时栈中信息依次出栈并存入停车场栈。完成后,便道内的第一辆车辆进入停车场栈,便道内车辆依次前移。输入离开车辆离开的时间,调用函数计算出车辆在停车场内停留的时间和所需的停车费,输出该车辆的所有信息。

2) 工作分配

- 1. 框架设计:由犹晓婷、王艳进行构思,同时参考其他人的意见。
- 2. 系统设计:由蒋沁洁、犹晓婷进行框架的编写。
- 3. 程序设计:由王艳、蒋沁洁进行各项功能的编辑。
- 4. 程序调试:由王艳、犹晓婷进行程序的调试与调整。
- 5. 文档制作,由蒋沁洁完成文档的编辑工作。

五、 程序模块

程序的模块为：本程序包含主程序模块（如图 5.1 5.2 所示）、菜单选择模块（如图 5.3 所示）和栈及队列操作模块（如图 5.4 所示）

```
int _tmain(int argc, _TCHAR* argv[])
{
    Park park;
    int num;
exit:
    {
        cout << "操作菜单:" << endl;           //输出操作菜单
        cout << "1、" << "输入到达车辆信息:" << endl;
        cout << "2、" << "输入离开车辆信息:" << endl;
        cout << "3、" << "查看停车场内的车辆信息:" << endl;
        cout << "4、" << "查看便道上的车辆信息:" << endl;
        cout << "0、" << "退出" << endl;
        cout << "请按操作菜单选择:" << endl;
        cin >> num;                               //输入选项
        if (num>4 || num<0){                       //判断选项是否合法
            cout << "输入有误, 返回操作菜单" << endl;
            goto exit;                             //若选项不合法, 则再次输出操作菜单
        }
    }
}
```

图 5.1 主程序模块

```
exit1:
{
    switch (num){
        case 1:{                                //输入到达车辆信息
            park.arrive(); break;
        }
        case 2:{                                //输入离开车辆信息
            park.leave(); break;
        }
        case 3:{                                //查看停车场内的车辆信息
            park.sdisplay();break;
        }
        case 4:{                                //查看便道上的车辆信息
            park.ldisplay();break;
        }
        case 0:break;                           //退出
    }
    while (num != 0){                            //如果没有选择退出, 则循环输出操作菜单
        goto exit;
        goto exit1;
    }
    if (num == 0){                               //如果选择0, 则退出
        return 0;
    }
}
```

图 5.2 主程序模块

```

{
    cout << "操作菜单:" << endl;           //输出操作菜单
    cout << "1、" << "输入到达车辆信息:" << endl;
    cout << "2、" << "输入离开车辆信息:" << endl;
    cout << "3、" << "查看停车场内的车辆信息:" << endl;
    cout << "4、" << "查看便道上的车辆信息:" << endl;
    cout << "0、" << "退出" << endl;
    cout << "请按操作菜单选择:" << endl;
    cin >> num;                             //输入选项
    if (num>4 || num<0){                    //判断选项是否合法
        cout << "输入有误, 返回操作菜单" << endl;
        goto exit;                         //若选项不合法, 则再次输出操作菜单
    }
}

```

图 5.3 菜单选择模块

```

19 typedef struct{                          //设定栈
20     car data[MAXSIZE];
21     int top;
22 }SeqStack;
23 SeqStack *Init_SeqStack(){                //置空栈
24     SeqStack *s;
25     s = new SeqStack;
26     s->top = -1;
27     return s;
28 }
29 typedef struct node{                     //链队结点的类型
30     car data;
31     struct node *next;
32 }QNode;
33 typedef struct{                          //将头尾指针封装在一起的链队
34     QNode *front, *rear;
35 }LQueue;
36 LQueue *Init_LQueue(){                  //置空链队
37     LQueue *q;
38     QNode *p;
39     q = new LQueue;
40     p = new QNode;
41     p->next = NULL;
42     q->front = p;
43     q->rear = p;
44     return q;
45 }

```

图 5.4 栈队列模块

录入功能：录入车辆信息包括到达时间（如图 5.5 所示）或车辆离开时间（如图 5.6 5.7 所示）


```

int arrive() { //车辆到达
    car s;
    QNode *L;
    s.state = "到达"; //定义到达的车辆的状态
    i++; //用于记录进站的车位于第几号车位
    cout << "请输入车牌号: "; //输入车牌号
    cin >> s.number;
    cout << "请依次输入车辆到达时间(年、月、日、小时)" << endl; //输入车辆到站的时间
    cin >> s.ptime.year >> s.ptime.month >> s.ptime.day >> s.ptime.hour;
    if (s1->top < MAXSIZE - 1){ //当停车场的容量未满足时
        s1->top++; //入栈
        s1->data[s1->top] = s;
        cout << "车辆停在停车场的第" << i << "号车位" << endl;
        return 1;
    }
    else{ //当停车场容量已满足时, 停入便道
        cout << "停车场已满, 请停放在便道" << endl;
        L = new QNode; //入链队
        L->data = s;
        L->next = NULL;
        L1->rear->next = L;
        L1->rear = L;
        cout << "车辆停在便道的第" << i-MAXSIZE << "号车位" << endl;
        return 1;
    }
}

```

图 5.5 车辆到达模块

```

void leave() { //车辆离开
    car s;
    QNode *L;
    string num;
    int flag = 0, sumpri=0, ye, mo, da, ho;
    cout << "请输入需要离开的车辆的车牌号: "; //输入需要离开的车辆的车牌号用于查找
    cin >> num;
    while (s1->top != -1){ //当停车场内有车辆时
        s = s1->data[s1->top]; //出栈
        s1->top--;
        if (s.number == num){ //当停车场内有需要离开的车辆的车牌号时
            flag = 1;
            cout << "该车停在停车场内的" << s1->top + 1 << "号车位上" << endl;
            s.state = "离开";
            cout << "请依次输入车辆离开时间(年、月、日、小时)" << endl; //输入离开的车辆的信息
            cin >> s.ltime.year >> s.ltime.month >> s.ltime.day >> s.ltime.hour;
            ye = s.ltime.year;
            mo = s.ltime.month;
            da = s.ltime.day;
            ho = s.ltime.hour;
            cout << "离开的车辆的信息: " << endl;
        }
    }
}

```

图 5.6 车辆离开模块

```

17 else{                                     //将停车场栈里出栈的数据存入临时栈
18     s2->top++;                             //入栈
19     s2->data[s2->top] = s;
20 }
21
22 while (s2->top != -1){                     //当临时栈里有车辆时
23     s = s2->data[s2->top];                 //出栈
24     s2->top--;
25     s1->top++;                             //将临时栈里出栈的数据存入停车场栈
26     s1->data[s1->top] = s;                 //如果有车辆离开，后面的车辆依次前移一个车位
27 }
28
29 if (flag == 1){                           //如果停车场内有车辆离开
30     if (L1->front != L1->rear){           //就让便道里的第一辆车进入停车场，后面的车辆依次前移一个车位
31         L = L1->front->next;
32         L1->front->next = L->next;
33         L->data.ptime.year=ye;
34         L->data.ptime.month=mo;
35         L->data.ptime.day=da;
36         L->data.ptime.hour=ho;
37         s = L->data;
38         s1->top++;
39         s1->data[s1->top] = s;
40         delete L;
41         if (L1->front->next == NULL)
42             L1->rear = L1->front;
43     }
44 }

```

图 5.7 车辆离开模块（便道上的车辆进入停车场）

计算功能：车辆离开时自动计算车辆停车费用（如图 5.8 所示）

```

.....
sumpri = price*(24 * compare1(s.ptime, s.ltime) + compare(s.ptime, s.ltime)); //计算停车费
cout << "状态" << '\t' << "停车费" << '\t' << "车牌号" << '\t' << "车辆到达时间" << '\t' << "车辆离开时间" << endl;
cout << s.state << '\t' << sumpri << '\t' << s.number << '\t';
cout << s.ptime.year << "年" << s.ptime.month << "月";
cout << s.ptime.day << "日" << s.ptime.hour << "时" << '\t';
cout << s.ltime.year << "年" << s.ltime.month << "月";
cout << s.ltime.day << "日" << s.ltime.hour << "时" << endl;
break;

```

图 5.8 计算停车费用模块

查看功能：查看停车场车辆信息及便道停留车辆（如图 5.9 所示）

```

void sdisplay(){                                     //查看停车场内的车辆信息
    cout << "状态" << '\t' << "车牌号" << '\t' << "车辆到达时间" << endl;
    for (int i = 0; i <= s1->top; ++i){
        cout << s1->data[i].state << '\t' << s1->data[i].number << '\t';
        cout << s1->data[i].atime.year << "年";
        cout << s1->data[i].atime.month << "月";
        cout << s1->data[i].atime.day << "日" ;
        cout << s1->data[i].atime.hour << "时" << endl;
    }
}

void ldisplay(){                                     //查看便道上的车辆信息
    QNode *L = L1->front->next;
    cout << "状态" << '\t' << "车牌号" << '\t' << "车辆到达时间" << endl;
    while(L!=NULL){
        cout << L->data.state << '\t' << L->data.number << '\t';
        cout << L->data.atime.year << "年";
        cout << L->data.atime.month << "月";
        cout << L->data.atime.day << "日";
        cout << L->data.atime.hour << "时" << endl;
        L = L->next;
    }
}

```

图 5.9 查看停车场内或便道模块

退出功能：退出系统。

六、 主程序

程序 1. 车辆节点进栈

```

typedef struct {                                     //设定栈
    car data[MAXSIZE];
    int top;
}SeqStack;

SeqStack* Init_SeqStack() {                         //置空栈
    SeqStack* s;
    s = new SeqStack;
    s->top = -1;
    return s; }

```

程序 2. 车辆节点进队列

```

typedef struct {                                     //将头尾指针封
装在一起的链队
QNode* front, * rear;
}LQueue;

LQueue* Init_LQueue() {                             //置空链队
LQueue* q;
QNode* p;
q = new LQueue;
p = new QNode;
p->next = NULL;
q->front = p;
q->rear = p;
return q;     }

```

程序 3. 车辆到达登记

```

int arrive() {                                     //车辆到达
car s;
QNode* L;
s.state = "到达";                                //定义到达的车辆的状态
/*s.ltime = { 0, 0, 0, 0 };                      //将车辆离开的时间先设定
为 0*/
i++;                                              //用于记录进
站的车位于第几号车位
cout << "请输入车牌号：";                       //输入车牌号
cin >> s.number;
cout << "请依次输入车辆到达时间（年、月、日、小时）" << endl; //
输入车辆到站的时间

```

程序 4. 车辆离开处理

```

void leave() {                                     //车辆离开

```

```

car s;
QNode* L;
string num;
int flag = 0, sumpri = 0, ye, mo, da, ho;
cout << "请输入需要离开的车辆的车牌号:";           //输入需要离开的车辆的车牌号
用于查找
cin >> num;
while (s1->top != -1) {                                // 当
停车场内有车辆时
s = s1->data[s1->top];                                //出栈
s1->top--;
if (s.number == num) {                                //当停车场内
有需要离开的车辆的车牌号时
flag = 1;
cout << "该车停在停车场内的" << s1->top + 1 << "号车位上" << endl;
s.state = "离开";
cout << "请依次输入车辆离开时间（年、月、日、小时）" << endl; //输入离开的车
辆的信息

```

程序 5. 显示停车场内和便道上的车辆情况

```

void sdisplay() {                                       //
查看停车场内的车辆信息
cout << "状态" << &apos;\t&apos; << "车牌号" << &apos;\t&apos; << "车
辆到达时间" << endl;
for (int i = 0; i <= s1->top; ++i) {
cout << s1->data[i].state << &apos;\t&apos; << s1->data[i].number << &ap
os;\t&apos;;
cout << s1->data[i].atime.year << "年";
cout << s1->data[i].atime.month << "月";
cout << s1->data[i].atime.day << "日";

```

```

cout << s1->data[i].atime.hour << "时" << endl;}}

void ldisplay() { //
查看便道上的车辆信息

QNode* L = L1->front->next;

cout << "状态" << &apos;\t&apos; << "车牌号" << &apos;\t&apos; << "车
辆到达时间" << endl;
while (L != NULL) {
cout << L->data.state << &apos;\t&apos; << L->data.number << &apos;\t&ap
os;;
cout << L->data.atime.year << "年";
cout << L->data.atime.month << "月";
cout << L->data.atime.day << "日";
cout << L->data.atime.hour << "时" << endl;
L = L->next;}}

```

程序 6. 车辆离开时的费用

```

sumpri = price * (24 * compare1(s.atime, s.ltime) + compare(s.atime, s
.ltime)); //计算停车费

cout << "状态" << &apos;\t&apos; << "停车费" << &apos;\t&apos; << "车
牌号" << &apos;\t&apos; << "车辆到达时间" << &apos;\t&apos; << "车辆离开
时间" << endl;

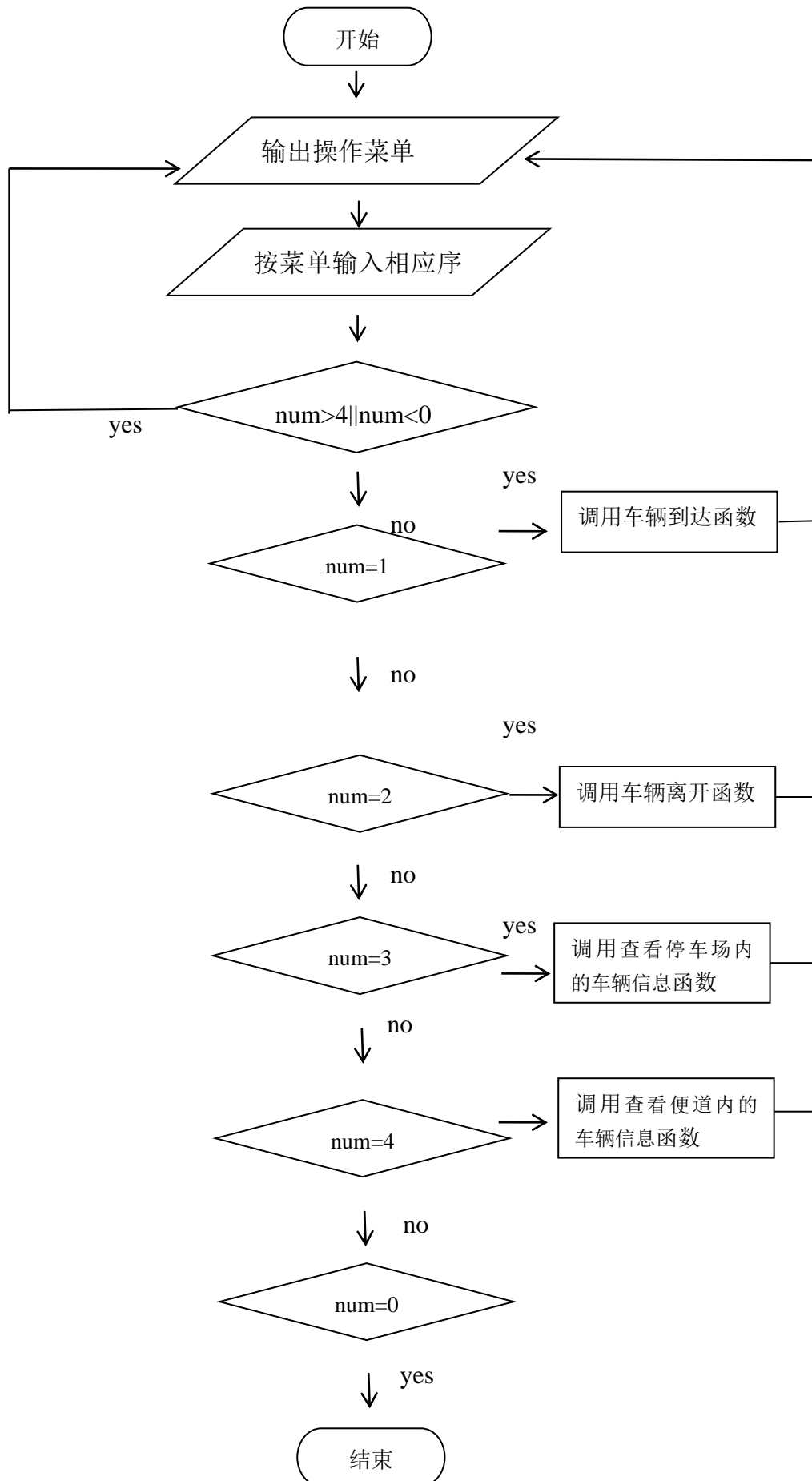
cout << s.state << &apos;\t&apos; << sumpri << &apos;\t&apos; << s.num
ber << &apos;\t&apos;;

cout << s.atime.year << "年" << s.atime.month << "月";
cout << s.atime.day << "日" << s.atime.hour << "时" << &apos;\t&apos;;
cout << s.ltime.year << "年" << s.ltime.month << "月";
cout << s.ltime.day << "日" << s.ltime.hour << "时" << endl;

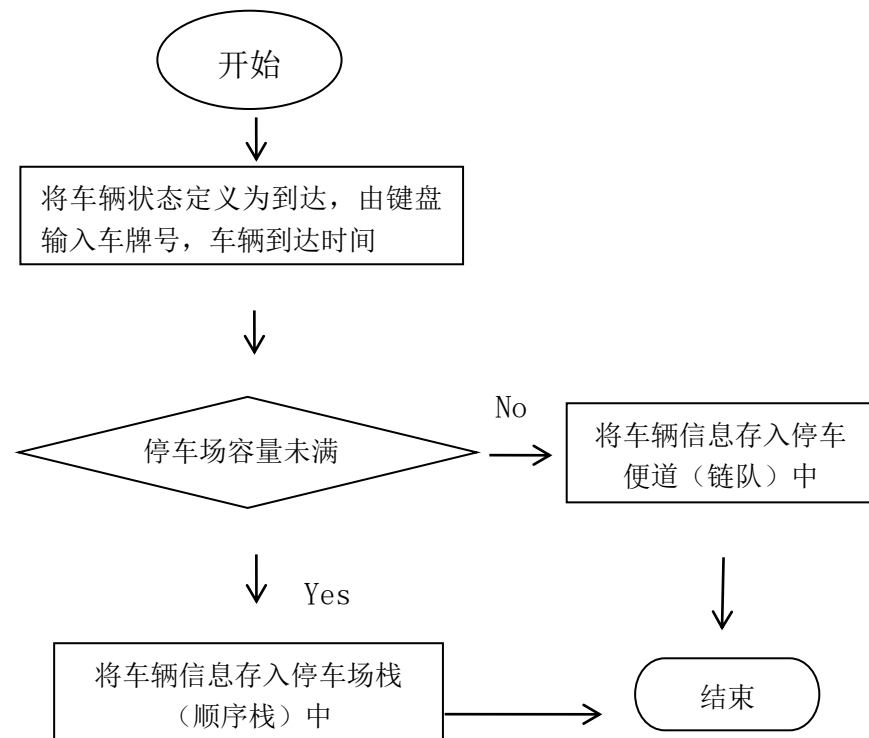
```

程序 7. 主程序画面函数

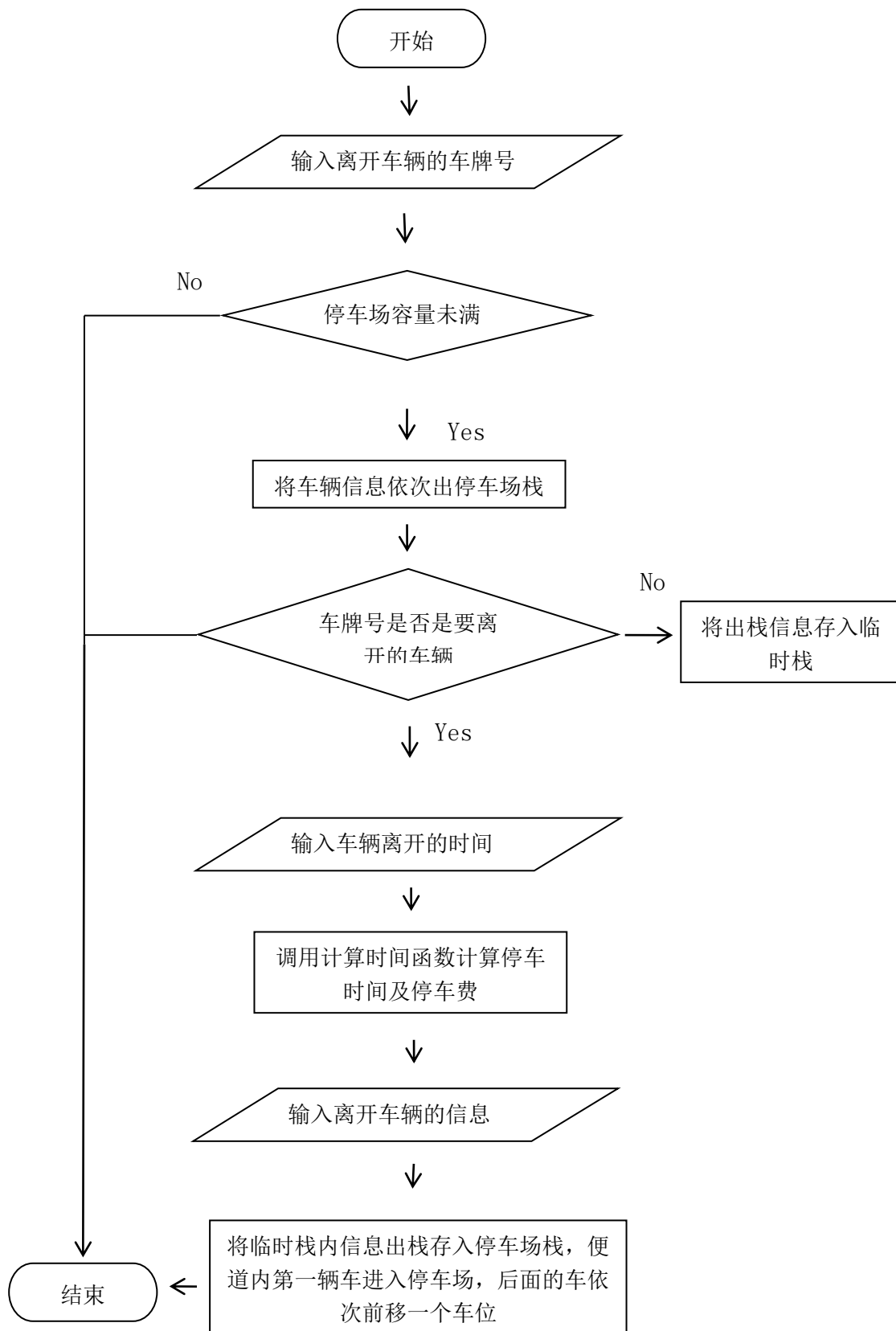
设置一个 main()函数，具体流程为：



再设置一个车辆到达函数，具体流程为：



最后设置一个车辆离开函数，具体流程为：



七、 运行截图

当用户进入停车场时，出现此系统，用户录入车辆信息，当前第一个停车用户的车牌为 A0001，进入时间为 2021 年 12 月 29 日 12 时，详情信息如图 7.1 所示：

```
操作菜单：
1、输入到达车辆信息：
2、输入离开车辆信息：
3、查看停车场内的车辆信息：
4、查看便道上的车辆信息：
0、退出
请按操作菜单选择：
1
请输入车牌号：A0001
请依次输入车辆到达时间（年、月、日、小时）
2021 12 29 12
车辆停在停车场的第1号车位
```

图 7.1

当用户离开停车场时，输入车辆信息，车牌号为 A0001 的用户离开停车场，输入离开时间 2021 年 12 月 30 日 12 时，详情信息如图 7.2 所示：

```
操作菜单：
1、输入到达车辆信息：
2、输入离开车辆信息：
3、查看停车场内的车辆信息：
4、查看便道上的车辆信息：
0、退出
请按操作菜单选择：
2
请输入需要离开的车辆的车牌号：A0001
该车停在停车场内的0号车位上
请依次输入车辆离开时间（年、月、日、小时）
2021 12 30 12
```

图 7.2

当用户离开停车场时，系统自动计算停车费用，详情信息如图 7.3 所示：

```
操作菜单：
1、输入到达车辆信息：
2、输入离开车辆信息：
3、查看停车场内的车辆信息：
4、查看便道上的车辆信息：
0、退出
请按操作菜单选择：
2
请输入需要离开的车辆的车牌号：A0001
该车停在停车场内的0号车位上
请依次输入车辆离开时间（年、月、日、小时）
2021 12 30 12
离开的车辆的信息：
状态      停车费  车牌号  车辆到达时间    车辆离开时间
离开      120    A0001   2021年12月29日12时    2021年12月30日12时
```

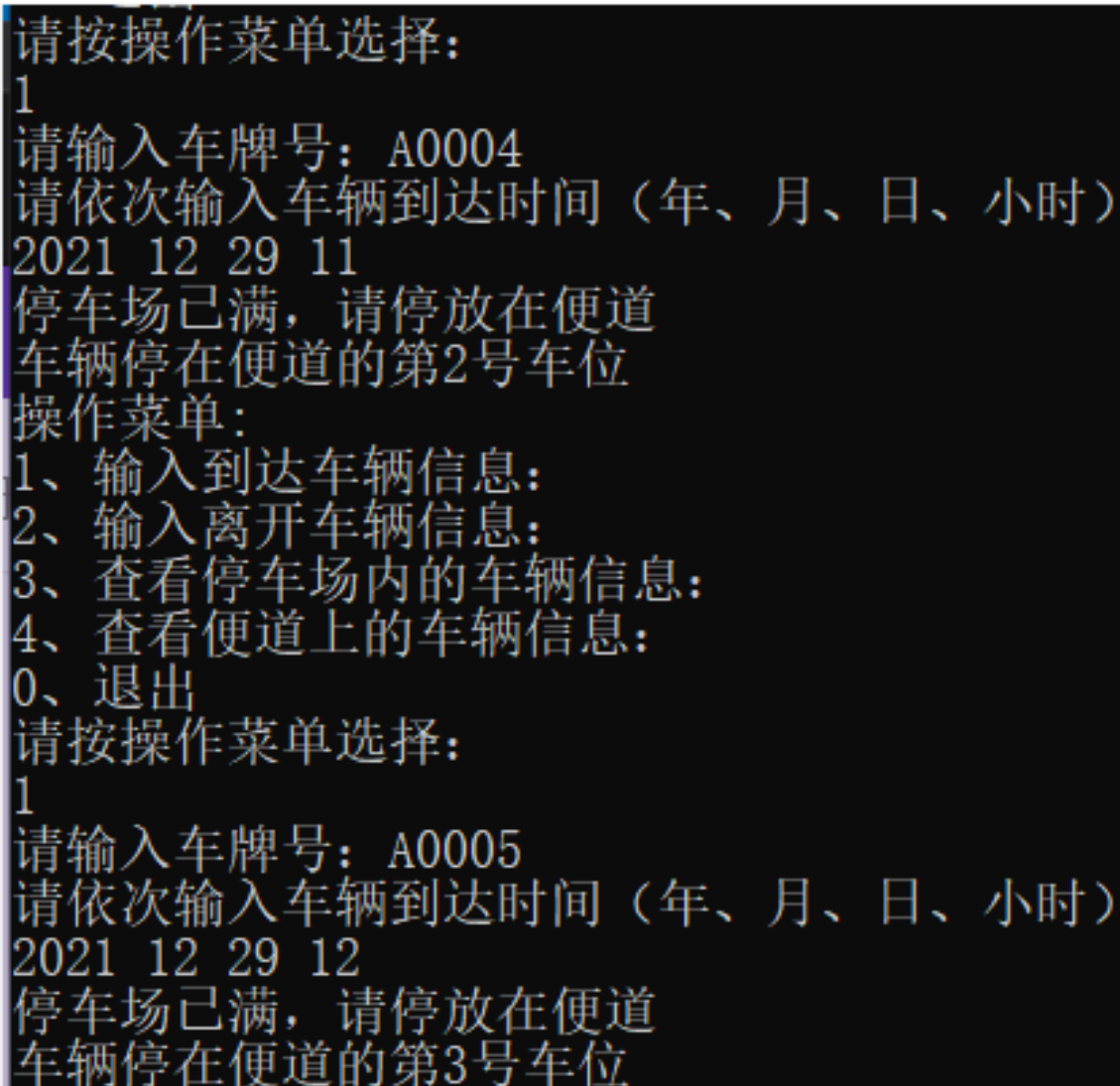
图 7.3

用户进入或者离开停车场时，可查看停车场内车辆信息，详情信息如图 7.4 所示：

```
操作菜单：
1、输入到达车辆信息：
2、输入离开车辆信息：
3、查看停车场内的车辆信息：
4、查看便道上的车辆信息：
0、退出
请按操作菜单选择：
3
状态      车牌号  车辆到达时间
到达      A0001   2021年12月29日8时
到达      A0002   2021年12月29日9时
到达      A0003   2021年12月29日10时
```

图 7.4

当停车场内满道时，再有用户进入停车场时，车辆便自动停入便道上，便道无限制，系统将此车辆录入便道停车信息，详情信息如图 7.5 所示：



```
请按操作菜单选择：
1
请输入车牌号：A0004
请依次输入车辆到达时间（年、月、日、小时）
2021 12 29 11
停车场已满，请停放在便道
车辆停在便道的第2号车位
操作菜单：
1、输入到达车辆信息：
2、输入离开车辆信息：
3、查看停车场内的车辆信息：
4、查看便道上的车辆信息：
0、退出
请按操作菜单选择：
1
请输入车牌号：A0005
请依次输入车辆到达时间（年、月、日、小时）
2021 12 29 12
停车场已满，请停放在便道
车辆停在便道的第3号车位
```

图 7.5

用户进入停车场时，可查看停车场内车道消息，详情信息如图 7.6 所示：

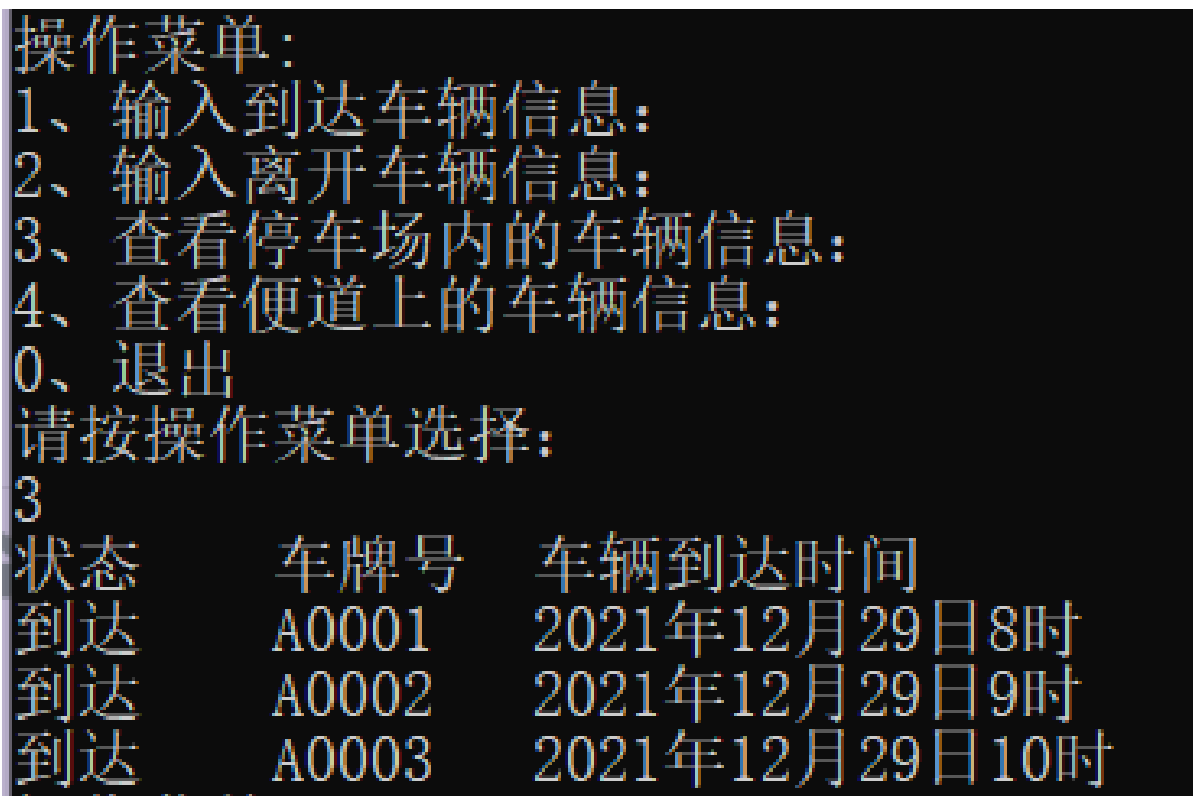


图 7.6

用户进入停车场时，当发现停车场内的所有车道均有车辆停靠时，可查看便道车辆信息，将车停靠在便道上，当停车场内有用户离开时，系统将自动将停靠在便道上的第一个车辆放置停车场内，详情信息如图 7.7 所示：

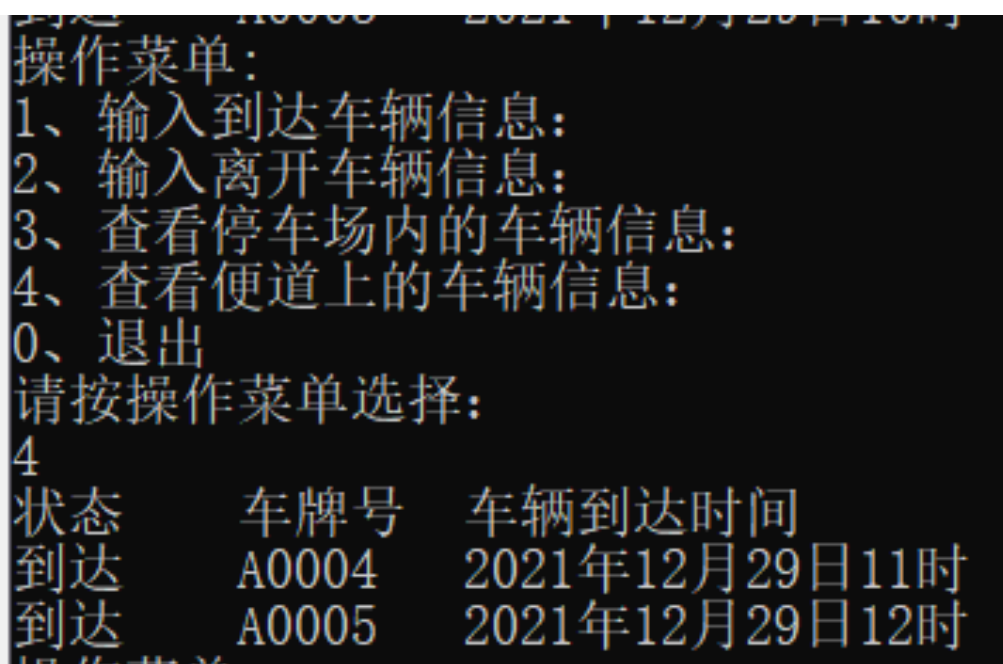


图 7.7

八、 总结

通过本次课程设计，我对数据结构的理解进一步加深，我理解和掌握了每段程序代码的功能及含义，并且能够实现模拟停车场管理的主要功能：利用栈实现车的入库问题，利用队列完成车在便道上停放的问题，主要是通过栈和队列的初始化，调用等函数来模拟停车问题。学习数据结构，我学会如何将所学的知识运用到实际中，解决一些实际的问题，这才是学习的根本。虽然我们的系统目前尚不够完善，但我相信后面我们会做的越来越好。

在确定此次我们需要完成的系统后，我们首先对这个系统构建了一个初步的框架，再通过不断的操作及讨论去一步步实现并完善这些框架；首先第一个框架：录入车辆信息。在完成这个框架时，我们主要是

此次的实训，我收获如下 1. 巩固和加深了对数据结构的理解，提高综合运用本课程所学知识的能力。2. 培养了我选用参考书，查阅手册及文献资料的能力。培养独立思考，深入研究，分析问题、解决问题的能力。3. 通过实际编译系统的分析设计、编程调试，掌握应用软件的分析方法和工程设计方法。4. 通过课程设计，培养了我严肃认真的学习作风，逐步建立正确的全局观念。刚开始觉得很难，到最后把这个做出来，付出了很多，也得到了很多，以前总以为自己对编程的地方还不行，现在才发现只要认真做，没有什么不可能。

对于编程时，也要认真仔细，出现错误要及时找出并改正，特别是对于英语的要求也体现出来了，因为它说明错误的时候也是英语的，遇到问题要及时去查相关的资料。反复的调试程序，最好是多找几个同学来对你的程序进行调试并听其对你的程序的建议，在他们不知道程序怎么写的时候完全以一个用户的身份来对你的用户界面做一些建议，正所谓当局者迷，旁观者清，同时在写代码的时候也要形成自己的编写程序与调试程序的风格，从各个细节出发，不放过每个知识点，注意与理论的联系和理论与实践的差别，另外也要注意符号的使用，所有符号均在英文状态下使用。