装 订 线

班级： 学号： 姓名：

课程名称： **数据结构** 考核日期： 2019年12月18日

**题目：航班查询订票系统**

# 一、需求分析与功能描述

设计并实现一个航班信息查询和检索系统。实现俩类登陆，用户登陆和业务员登陆，对飞机航班信息进行排序和查找，可按照航班号、起点站、到达站、起飞时间和到达时间等信息进行查询。

# 二、数据结构的选择与设计

## （一）系统中的数据定义

面向的数据有航班号、起始站、终点站、班期、起飞时间、到达时间、机型、票价。

数据元素

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 名字 | 航班号 | 起始站 | 终点站 | 班期 | 起飞时间 | 到达时间 | 机型、 | 票价 |
| 变量名 | number | staddress | arraddress | DATE | stime | atime | TYPE | value |
| 类型 | char | char | char | char | char | char | char | Int |
| 长度 | 10 | 10 | 10 | 10 | 10 | 10 | 4 | 无 |

数据项Data\_type

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 名字 | 航班号 | 起始站 | 终点站 | 班期 | 起飞时间 | 到达时间 | 机型、 | 票价 |
| 变量名 | number | staddress | arraddress | DATE | stime | atime | TYPE | value |
| 类型 | char | char | char | char | char | char | char | Int |
| 长度 | 10 | 10 | 10 | 10 | 10 | 10 | 4 | 无 |

数据项Node\_list

|  |  |  |
| --- | --- | --- |
| 名字 | 数据项 | 指向下一个数据项 |
| 变量名 | \*info | \*next |
| 类型 | Data\_type | node |

## （二）数据结构选择

选择单链表存储数据，项目中用到的数据，每一个项中包含多个数据元素，并且要实现插入和查找等功能，所以使用单链表实现这些功能更加方便快捷，并且方便之后进行查找算法的实现。

在实现单链表储存数据时，先建立第一个结构体Data\_type，保存基本数据。然后建立第二个结构体Node\_list，构造单链表，在其中定义一个Data\_type类型指针\*info指向其基本数据，定义一个node类型指针\*next指向下一个数据项

可以通过画结构图或流程图阐述系统模块之间的关系以及运行的先后顺序。

三、算法的选择与设计

## （一）排序

1.冒泡排序

2. 选择原因及理由

起泡排序使我们接触的第一种排序方式，起泡排序的外层for的循环次数为n-1，内层For循环的循环次数为i，可知内层的总比较次数为O（n^2）所以起泡排序的最好、最坏和平均情况下的时间复杂程度相同，程序执行相对稳定。

## （二）查找

1.顺序查找

void inquire\_by\_number(Node\_list \*L, char num[] ) {

L=L->next;

int flag=0;

while(L != NULL)

{

if(strcmp(num,L->info->number) == 0)

{

show\_one(L);

flag=1;

break;

}

L=L->next;

}

if( flag == 0)

cout<<"对不起！没有此航班!\n";

}

2.选择原因及理由

顺序查找与其他查找方法相比，查找成功的平均查找长度比较长，但是对表没有什么特别的要求，并且线性链表只能进行顺序查找，由于程序使用了单链表，所以选择顺序查找。

# 四、主要代码

typedef struct node //定义存储航班信息结构体链表

{

Data\_type \*info;

struct node \*next;

} Node\_list;

Node\_list \*L; //定义存储链表

Data\_type \*E; //定义航班信息结构体

/\*

用于存储链表的初始化

\*/

bool init\_N(Node\_list \*\*L)

{

(\*L)=(Node\_list \*)malloc(sizeof(Node\_list));

if( (\*L)!=NULL )

{

(\*L)->next=NULL;

return true;

}

else

{

cout<<"malloc failed!\n";

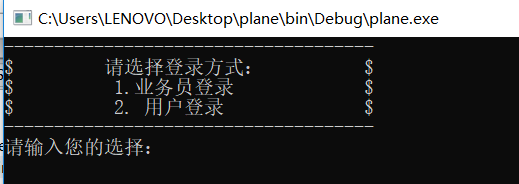
return false;

}

}

# 测试结果

# 登陆界面测试



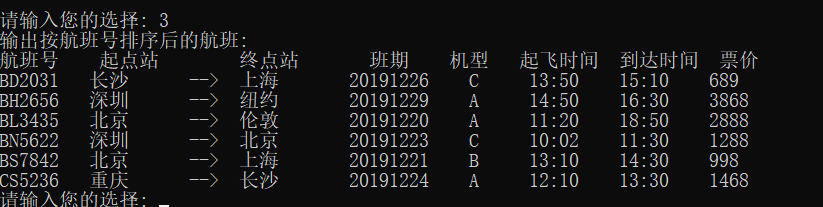
插入界面测试



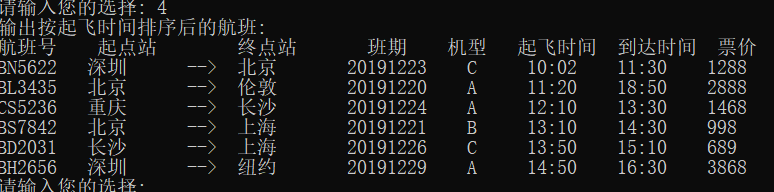
删除界面



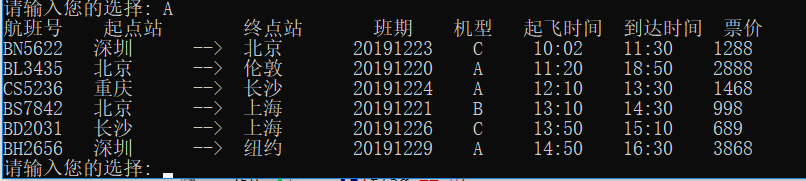
按照航班号排序



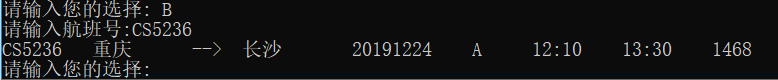
按照起飞时间排序



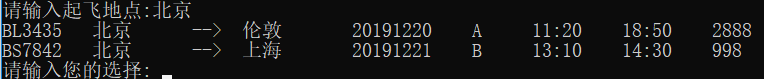
查询所有航班



按照航班号查找



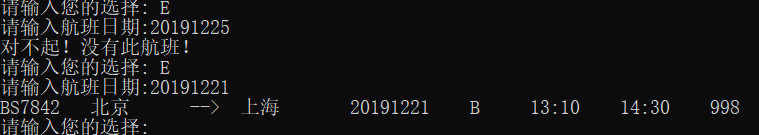
按照起点站查找



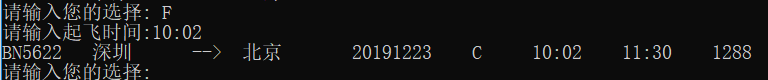
按照终点查询



按照班期查询



按照起飞时间查询



按照到达时间查询



六、分析与总结

此次实验中，所做程序满足了，程序所预定的需求，实现一个航班信息查询和检索系统。实现俩类登陆，用户登陆和业务员登陆，对飞机航班信息进行排序和查找，可按照航班号、起点站、到达站、起飞时间和到达时间等信息进行查询等功能。使用了单链表储存数据信息，使用顺序查找，对信息进行检索，使用冒泡排序，对数据进行排序。因为对于单链表使用的不熟练，没有更多的使用学到的知识，是一个遗憾。如果有可能，我会进一步改进，可以将数据进行写入文件，使其更加具有实际意义。

在本次实验中，我深刻体验到了，拥有解决问题能力的重要性，问题总会层出不穷，只有掌握了解决问题的方法，才能应对未来可能出现的问题，坚持和专注，使我们应该具备的品质，在我们的学习之中。之后的学习，应该更加专注其现实意义，与实际相结合，才能记得更加牢固，更加灵活，更加深刻**。**