**课程设计（实训）**

**资 料**

|  |  |  |
| --- | --- | --- |
| 学年学期 | ： | 2021-2022-1 |
| 课程名称 | ： | 面向过程程序设计（C语言）课程设计 |
| 题目 | ： | 航班信息管理系统 |
| 起止日期 | ： | 2021年12月25日 ～ 2021年12月31日 |
| 学院 | ： |  |
| 班级 | ： |  |
| 学生姓名 | ： |  |
| 学号 | ： |  |

目 录 清 单

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 材 料 名 称 | 资料数量 | 备 注 |
| 1 | 课程设计任务书 | 1 |  |
| 2 | 课程设计（实训）报告 | 1 |  |
|  |  |  |  |
|  |  |  |  |

课程设计（实训）任务书

学 院： 学生姓名： 专业班级：

课程名称： 面向过程程序设计（C语言）课程设计 设计题目： 航班信息管理系统

完成期限：自 2021 年 12 月 25 日至 2021 年 12 月 31 日共 1 周

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 内  容  及  任  务 | **一、课题的主要内容**  综合运用C语言程序设计课程的主要知识，设计并实现一个航班信息管理系统。主要内容包括：  ⑴ 增加航班记录。将新的航班记录增加到原有的航班数据文件中。在进行处理时必须检查所要增加的航班记录是否存在，如果已经存在，应给出提示信息后停止增加；  ⑵ 航班取消。如果某次航班的乘客数太少（已订票的少于本次航班最大载客数的10%），将取消该航班，但该航班的记录仍然保存在原有的航班数据文件中；  ⑶ 航班查询。应该有以下几种基本的查询方式：按航班号、按起飞地点和起飞时间、按降落地点，按起飞地点和降落地点；  ⑷ 航班订票。按上述问题描述中的乘客订票方式完成航班订票处理。  ⑸ 设计一个菜单，至少具有上述操作要求的基本功能。  **二、主要任务及说明**  ① 航班信息：航班号、最大载客数、起飞地点、起飞时间、降落地点、降落时间，单价；  ② 乘客信息：航班号、身份证号码、姓名、性别、出生年月、座位号。  乘客订票的主要方式是：乘客提出航班号、起飞地点、起飞时间、降落地点、订票数等订票要求，根据事先保存的航班数据决定乘客能否订票？只有全部满足了乘客的订票要求并且所订航班有足够的未订座位之后才能完成订票处理，并且修改该航班的未订座位数（每个航班的未订座位数的初始值就是该航班的最大载客数）；否则，订票失败，并且给出不能订票的原因。  要求将航班数据保存在数据文件中，在处理时按航班的起飞地点建立不同的链表。 | | | | |
| 进  度  安  排 | 起止日期 | | 工作内容 | | |
| 2021.12.20~2021.12.24 | | 问题分析和设计； | | |
| 2021.12.25~2021.12.30 | | 熟悉软件开发工具、编码系统、系统测试； | | |
| 2021.12.30~2021.12.31 | | 答辩，撰写并提交课程设计报告 (含电子文档)、源程序等。 | | |
| 主  要  参  考  资  料 | [1]谭浩强.C程序设计设计题解与上机指导(第5版) [M].清华大学出版社，2017.08  [2]颜晖,张泳.C语言程序设计实验与习题指导(第3版) [M].高等教育出版社，2015.08  [3]刘强，童启.C语言程序设计实验教程[M].电子工业出版社，2018.02  [4]刘博.C语言课程设计[M].机械工业出版社，2013.04  [5]明日科技. C语言函数参考手册[M].清华大学出版社，2012.02 | | | | |
| 同组信息 | 学号 | 姓名 | | 分工负责的内容 | 备注 |
| 无 |  | |  |  |

2021年12月17日

课程设计（实训）报告

|  |  |  |
| --- | --- | --- |
| 学年学期 | ： | 2021-2022-1 |
| 课程名称 | ： | 面向过程程序设计（C语言）课程设计 |
| 题目 | ： | 航班信息管理系统 |
| 起止日期 | ： | 2021年12月25日 ～ 2021年12月31日 |
| 学院 | ： |  |
| 班级 | ： |  |
| 学生姓名 | ： |  |
| 学号 | ： |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 成绩评定 | ： |  |  |
| 课程目标 | | 成绩 | 评定人签名 |
| 目标1 | |  |  |
| 目标2 | |  |
| 目标3 | |  |

**2021年 12 月**

目 录

[1 课题简介 1](#_Toc28164390)

[1.1 系统简介 1](#_Toc28164391)

[1.2 开发环境及开发工具 1](#_Toc28164392)

[2 问题分析 2](#_Toc28164393)

[2.1 需求调查 2](#_Toc28164394)

[2.1.1 功能要求 2](#_Toc28164395)

[2.1.2 技术要求 2](#_Toc28164396)

[2.2 数据分析 2](#_Toc28164397)

[3 系统设计 3](#_Toc28164398)

[3.1 系统功能设计 3](#_Toc28164399)

[3.2 系统的功能分析 3](#_Toc28164400)

[3.3 系统的数据结构设计 4](#_Toc28164401)

[4 功能模块的设计与实现 5](#_Toc28164402)

[4.1 系统主函数设计与实现 5](#_Toc28164403)

[4.1.1 主函数功能说明 5](#_Toc28164404)

[4.1.2 实现流程 5](#_Toc28164405)

[4.1.3 函数功能描述 6](#_Toc28164406)

[4.1.4 函数的调用关系 6](#_Toc28164407)

[4.1.5 关键源代码 7](#_Toc28164408)

[4.2 航班信息增加模块设计与实现 8](#_Toc28164409)

[4.2.1 功能说明 8](#_Toc28164410)

[4.2.2 实现流程 8](#_Toc28164411)

[4.2.3 关键源代码 9](#_Toc28164412)

[4.3 订票模块设计与实现 11](#_Toc28164413)

[5 系统测试 1](#_Toc28164414)3

[5.1 软件测试 1](#_Toc28164415)3

[5.1.1 单元测试 1](#_Toc28164416)3

[5.1.2 集成测试 1](#_Toc28164417)3

[5.2 系统有效性 1](#_Toc28164418)4

[6 总结 1](#_Toc28164419)5

[7 参考文献 1](#_Toc28164420)6

# 课题简介

## 系统简介

随着信息产业的飞速发展，信息化管理及查询已经进入并应用到各行各业，影响着人们的价值观念和生活方式。因此，要提高企业信息化建设，利用先进的办公自动化系统来实现企业内部信息管理、共享及交流，从而提高企业综合实力。

本次设计是针对航班的查询系统，该设计要求对飞机航班信息进行排序和查询。可按航班的航班号、起点站、终点站等信息进行航班信息的查询。

## 开发环境及开发工具

1. win10操作系统
2. 小熊猫Dev-c++
3. Visio

# 问题分析

## 需求调查

### 功能要求

⑴ 增加航班记录。将新的航班记录增加到原有的航班数据文件中。在进行处理时必须检查所要增加的航班记录是否存在，如果已经存在，应给出提示信息后停止增加；

⑵ 航班取消。如果某次航班的乘客数太少（已订票的少于本次航班最大载客数的10%），将取消该航班，但该航班的记录仍然保存在原有的航班数据文件中；

⑶ 航班查询。应该有以下几种基本的查询方式：按航班号、按起飞地点和起飞时间、按降落地点，按起飞地点和降落地点；

⑷ 航班订票。按上述问题描述中的乘客订票方式完成航班订票处理。

⑸ 设计一个菜单，至少具有上述操作要求的基本功能。

### 技术要求

1. 熟练运用结构体存储航班和乘客信息
2. 使用switch语句实现多功能操作
3. 自定义函数简化程序操作，实现代码可读性，美观性
4. 使用while循环实现重复操作
5. 建立链表方便遍历
6. 使用文件读取写入操作，方便数据增删改查

## 数据分析

1. 数据流图

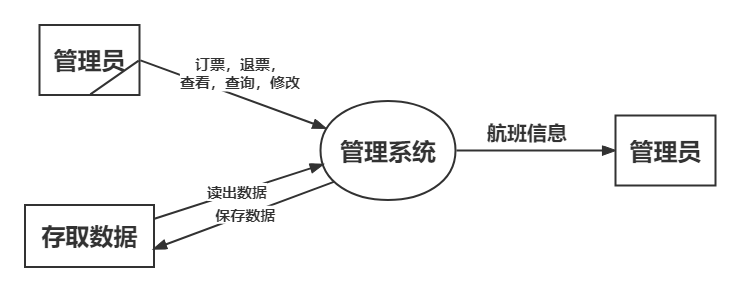


图2-1 数据流图

2、数据字典

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 航班号 | 最大载客量 | 起飞地点 | 起飞时间 | 降落地点 | 降落时间 | 单价 | 剩余票数 |
| MH870 | 160人 | 北京 | 11：00 | 乌鲁木齐 | 18：00 | ￥320 | 160 |
| ZM567 | 240人 | 哈尔滨 | 0：00 | 黎巴嫩 | 16：00 | ￥1210 | 240 |
| AQ857 | 180人 | 洛杉矶 | 14：00 | 休斯敦 | 18：00 | ￥360 | 180 |

图2-2数据字典

# 系统设计

## 系统功能设计

根据对航班管理的总体调研及分析，航班信息管理系统分为增加航班记录、航班查询、航班订票、航班退票、订票信息查询、删除航班记录、退出系统等7个功能模块。

## 系统的功能分析

基础信息管理主要是满足管理员管理需要，主要有以下功能。

（1）增加、删除航班记录

该功能分为两个模块，分别用于增加和删除。增加时支持多条航班记录同时添加。添加过程为先判断数据库中是否有该航班，如果存在则提示已经存在退出添加，不存在则将该条航班记录添加到链表末尾；删除时需要提供航班号，经过再次确认后即可删除

（2）航班查询

该功能模块主要是查询链表中的航班信息，查询方式主要有根据航班号查询、根据起飞地点和起飞时间查询、根据降落地点查询、根据起飞地点和降落地点查询等。若该航班订票人数不足最大载客量的10%，则提示该航班可能面临取消。

（3）航班订票、退票

该功能分为两个模块，分别用于订票和退票。订票时需要提供乘客的个人信息进行登录后再满足所有乘客的要求才能成功订票，若订票失败则返回失败原因，例如航班不存在或者航班没有余票。退票需要提供乘客信息和航班号，经过再次确定后方可退票。

1. 订票信息查询

该功能模块需要输入乘客的姓名及身份证号，如果存在该乘客信息，则输出该乘客的所有订单，如果不存在则输出未查询到。

## 系统的数据结构设计

（1）每个航班信息记录由多个不同的数据项构成，是一个结构体数据，因此需要定义结构体数据类型。根据题目要求，确定每个记录包括的数据项目为：航班号、最大载客量、起飞地点、起飞日期、降落地点、降落时间、单价、剩余票数等。据此定义如下结构体数据类型：

typedef struct planemessage {  
 char number[16];//航班编号  
    int maxbear;//最大载客量  
    char beginplace[64];//起飞地点  
    char endplace[64];//降落地点  
    char begintime[16];//起飞时间  
    char endtime[16];//降落时间  
    double price;//价格  
    int planesum;//剩余票数

struct planemessage \*next;//建立链表  
} planemessage;

每个乘客信息记录由多个不同的数据项构成，是一个结构体数据，因此需要定义结构体数据类型。根据题目要求，确定每个记录包括的数据项目为：航班号、姓名、性别、身份证号、出生日期、座位号等。据此定义如下结构体数据类型：

struct passengermessage {  
    char planenumer[16];//航班编号  
    char idnumber[32];//身份证号  
    char name[16];//姓名  
    char sex[8];//性别  
    char born[16];//出生日期  
    int sitnumber;//座位号  
};

struct passengermessage passenger[MAXN];

（2）考虑到struct planemessage、struct passengermessage是航班信息管理程序中的通用数据类型，将在多个函数中使用，故将其在靠前位置定义。

（3）使用文件test.txt存储数据类型为stuct  planemessage的航班信息数据。

（4）为了方便增加删除航班信息记录，用链表形式存储航班信息，初始化如下：

planemessage \* initLink() {  
    planemessage \* p = (planemessage\*)malloc(sizeof(planemessage));//建立头指针  
    p->next = NULL;  
    return p;  
}  
planemessage \*head = initLink();//head为航班信息的头指针

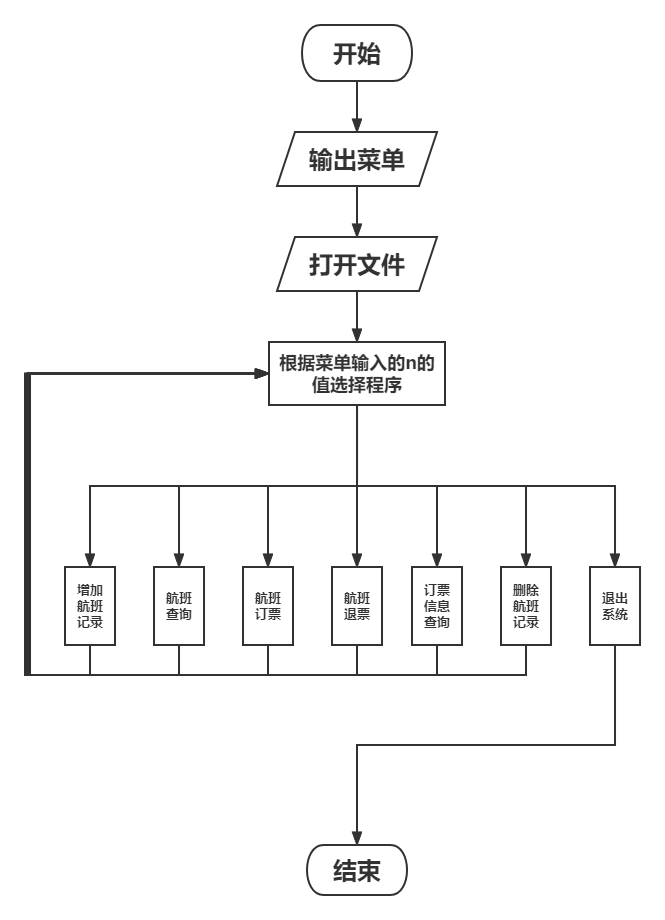
# 功能模块的设计与实现

## 系统主函数设计与实现

### 主函数功能说明

首先将test.txt文件中的航班信息数据读入到结构体链表中，再在界面上显示菜单界面，用户通过输入不同的操作号实现相应的功能。

### 实现流程



### 函数功能描述

（1）main()函数

航班信息管理程序主函数，实现程序功能的主菜单显示，通过各功能函数的调用，实现整个程序的功能控制。

（2）add()函数

函数原型为void add()，是输入数据函数，它实现航班信息数据的键盘输入。

（3）cancel()函数

函数原型为void cancel()，是删除航班信息的函数，它实现航班信息数据的删除。

（4）menu()函数

函数原型为void menu()，是显示菜单的函数，它实现在界面打印菜单的功能。

（5）order()函数

函数原型为void order()，是订票的函数，它实现乘客订票的功能。

（6）cancelticket()函数

函数原型为void cancelticket()，是取消订票的函数，它实现乘客退票的功能。

### 函数的调用关系

（1）main()函数直接调用的函数：menu()函数、writein()函数、add()函数、checkplane()函数、order()函数、cancelticket()函数、checkticket()函数、cancel()函数。

（2）checkplane()函数直接调用的函数：checkplan1()函数、checkplan2()函数、checkplan3()函数、checkplan4()函数。

（3）checkticket()函数直接调用的函数：prints()函数。

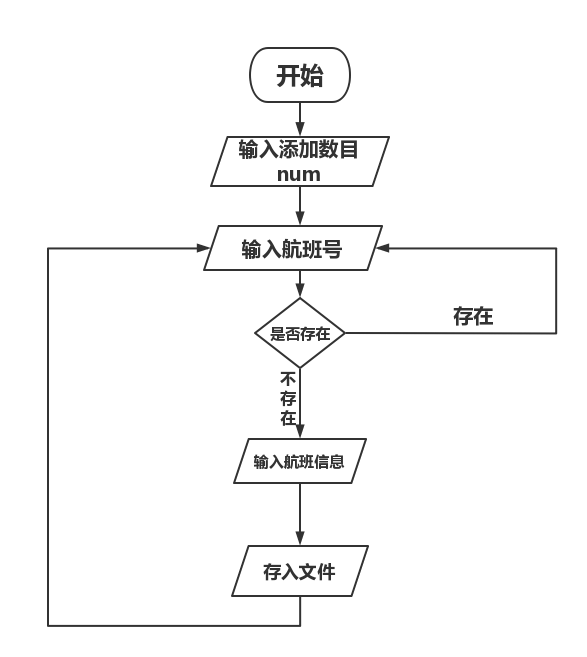
### 关键源代码 int main() {     writein();     while (1) {         menu();         do {             printf("请输入操作编号1~7:");             scanf("%s", n);         } while (n[0] < '1' || n[0] > '7' || strlen(n) > 1);         switch (n[0]) {             case '1':                 system("cls");                 add();                 break;             case '2':                 system("cls");                 checkplane();                 break;             case '3':                 system("cls");                 order();                 break;             case '4':                 system("cls");                 cancelticket();                 break;             case '5':                 system("cls");                 checkticket();                 break;             case '6':                 system("cls");                 cancel();                 break;             case '7':                 return 0;         }         do {             printf("是否返回菜单?(Y/N):");             scanf("%s", n);         } while ((n[0] != 'Y' && n[0] != 'N') || strlen(n) > 1);         if (n[0] == 'Y') {             system("cls");             continue;         } else             return 0;     } }

## 航班信息增加模块设计与实现

### 功能说明

此模块用add()函数实现，函数原型为void add()，是输入数据函数，它实现航班信息数据的键盘输入。

### 实现流程



### 

### 关键源代码

void add() {  
    int sum;  
    char number[24];  
    planemessage \*plane = head;  
    printf("请输入添加数目:");  
    scanf("%d", &sum);  
    for (int i = 0; i < sum; i++) {  
        int flag = 0;  
        printf("请输入航班编号:");  
        scanf("%s", number);  
        for (; plane->next != NULL; plane = plane->next)  
            if (!strcmp(plane->next->number, number)) {  
                flag = 1;  
                break;  
            }  
        if (flag == 1) {  
            printf("已存在该航班信息！\n");  
            continue ;  
        }  
        plane->next = (planemessage \*)malloc(sizeof(planemessage));  
        strcpy(plane->next->number, number);  
        printf("请输入最大载客量:");  
        scanf("%d", &plane->next->maxbear);  
        printf("请输入起飞地点:");  
        scanf("%s", plane->next->beginplace);  
        printf("请输入起飞时间(格式xx:xx):");  
        scanf("%s", plane->next->begintime);  
        printf("请输入降落地点:");  
        scanf("%s", plane->next->endplace);  
        printf("请输入降落日期:");  
        scanf("%s", plane->next->endtime);  
        printf("请输入单价:");  
        scanf("%lf", &plane->next->price);  
        plane->next->planesum = plane->next->maxbear;  
        FILE \*fp;  
        fp = fopen("test.txt", "a+");  
        if (fp == NULL) {  
            printf("打开航班数据库失败!\n");  
            exit(0);  
        }  
        fprintf(fp, "%s %d %s %s %s %s %lf %d\n", plane->next->number, plane->next->maxbear, plane->next->beginplace, plane->next->begintime, plane->next->endplace, plane->next->endtime, plane->next->price, plane->next->planesum);  
        fclose(fp);  
        plane->next->next=NULL;  
    }  
}

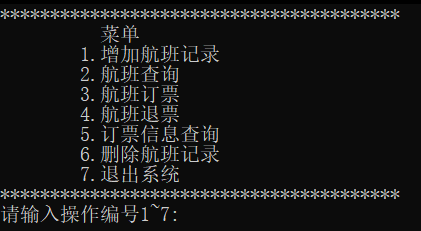
## 订票模块设计与实现

void order() {  
    struct passengermessage pass;  
    printf("请输入姓名:");  
    scanf("%s", pass.name);  
    printf("请输入身份证号:");  
    scanf("%s", pass.idnumber);  
    printf("性别(男/女):");  
    scanf("%s", pass.sex);  
    printf("出生年月(xxxx年xx月):");  
    scanf("%s", pass.born);  
    printf("登录成功!\n");  
    printf("---------------------------------------\n");  
    int sum;  
    printf("请输入订票数目:");  
    scanf("%d", &sum);  
    for (int i = 0; i < sum; i++) {  
        struct planemessage plane1;  
        printf("请输入航班号:");  
        scanf("%s", plane1.number);  
        printf("请输入起飞地点:");  
        scanf("%s", plane1.beginplace);  
        printf("请输入起飞日期:");  
        scanf("%s", plane1.begintime);  
        printf("请输入降落地点:");  
        scanf("%s", plane1.endplace);  
        int flag = 0;  
        for (planemessage \*plane = head; plane->next != NULL; plane = plane->next) {  
            if (!strcmp(plane->next->number, plane1.number) && !strcmp(plane->next->beginplace, plane1.beginplace) && !strcmp(plane->next->begintime, plane1.begintime) && !strcmp(plane->next->endplace, plane1.endplace)) {  
                flag = 1;  
                if (plane->next->planesum) {  
                    flag = 2;  
                    strcpy(pass.planenumer, plane->next->number);  
                    pass.sitnumber = plane->next->maxbear - plane->next->planesum + 1;  
                    printf("订票成功!您的座位号是:%d\n", pass.sitnumber);  
                    (plane->next->planesum)--;  
                    passenger[++passengernum] = pass;  
                }  
            }  
        }  
        if (!flag)  
            printf("订票失败，未找到符合条件的航班\n");  
        else if (flag == 1)  
            printf("订票失败，航班票数不足!\n");  
    }  
}

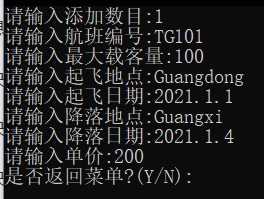
# 系统测试

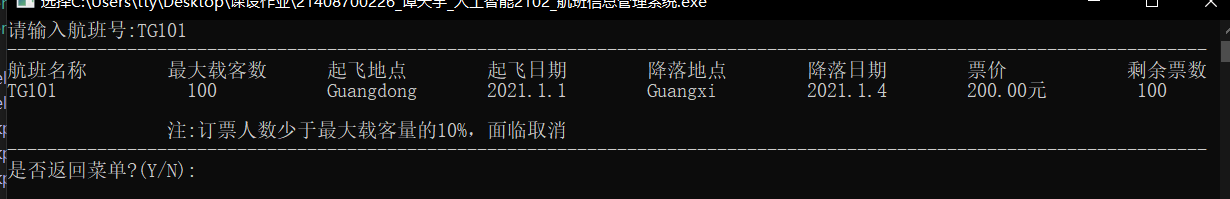
## 软件测试

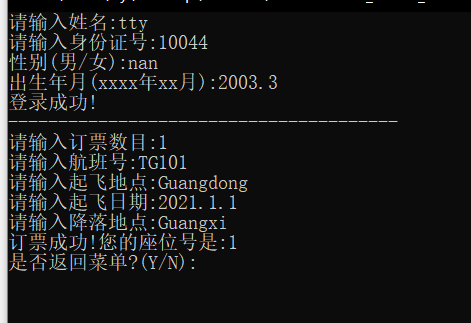
1. 编译运行后显示菜单



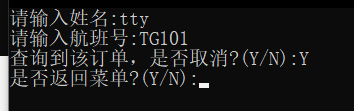
1. 输入菜单号“1”，开始航班信息的录入



1. 输入菜单号“2”，开始航班信息的查询
2. 输入菜单号“3”，开始航班订票



1. 输入菜单号“4”，开始航班退票



## 系统有效性

# 总体上，该航班信息管理系统可以满足大部分航班信息管理的功能，能够实现精确地查找和快速地订票退票。但是，也存在很多不足。例如：

# 1.该系统我未设计管理员登陆账号密码，在信息安全保护上不到位；

# 2.在订票环节，需要乘客已知的信息过于繁琐，降低了乘客及时订票的效率；

# 3.在时间上，不够具体明了，容易造成乘客的困惑；

# 4.未能及时把更改后的数据存入文件中，容易造成断电数据丢失的后果；

5.文件中若存在中文信息读入到链表中会变成乱码

# 因此，我还需要不断地完善该系统。总结

经过这几天的C语言课程设计，我收获很大，在实现系统功能的过程中，也遇到很多问题，及时的向周围的同学请教，寻求帮助，设计修改，编写调试，每一步都渗透着沟通和分析，每一次沟通之后，都会对系统功能有更深层次的理解和阐述，这也促使我一步步完成了整个系统，并使之可以运行。时间太短暂了，我没有办法把全部学到的知识运用到具体的系统实现中，深深感受到数据结构这门课程的重要性，以及它在实际应用中的广泛性。这个飞机订票系统由于种种原因仍然存在需要改进，需要完善的地方，我对于这个系统的认识和对数据结构这门课程的理解已经更加深入了。在实现系统的过程中，我感受到自己的知识的薄弱零散，经过这次做系统，我觉得自己的知识得到巩固和整合，知识体系更加系统化。我也深深体会到实践检验真理，只有实践中，才能出真知。我懂得了学习不只是纸上谈兵，更重要的是实践，就比如平常我们在学习中经常会遇到一些问题，看似无关紧要跳过去，不求甚解，在得知答案的时候，总是感觉心领神会，久而久之，对于的知识点就模棱两可，形成知识漏洞。但这次课程设计所要编写的程序很长，涉及的知识点也很多，选择此题目的时候我心里面还底气十足，自认为没什么问题，对于程序的步骤仿佛了然于胸，但在实际操作中才发现并非如此，在编写程序的过程中，我不断地上网查阅相关资料进行学习，积极地思考该怎样实现系统的每一个功能。

通过这次数据结构的课程设计，我巩固和加深对数据结构基本知识的理解，锻炼了综合运用课本知识的能力，自己也真正地完整而系统地设计并编写了一个程序，使得自己的编程能力大为提高。通过对我所选择飞机订票系统的设计和编写，我初步掌握简单软件的分析方法和设计方法，第一次体会到了数据结构编程知识在实际生活中的重要性和应用的广泛性，最后，我要感谢老师和同学们，正是因为他们的热心帮助，我才能正确的完成这次课程设计。在今后的学习和工作中，会更加深入地把知识和思想体会更广泛地应用到其他学科的学习中。

# 参考文献

1. 何钦铭.C语言程序设计 (第3版) [M].高等教育出版社，2015.08
2. 谭浩强.C程序设计(第5版) [M].清华大学出版社，2017.08
3. ]颜晖,张泳.C语言程序设计实验与习题指导(第3版) [M].高等教育出版社，2015.08
4. 刘强，童启.C语言程序设计实验教程[M].电子工业出版社，2018.02
5. 刘博.C语言课程设计[M].机械工业出版社，2013.04
6. 明日科技. C语言函数参考手册[M].清华大学出版社，2012.02
7. 韩先峰.脑动力：C语言函数速查效率手册[M]. 电子工业出版社，2012.07