

# 数据结构与算法实验综合实验题

## ——高铁订票系统

### 一、 基本功能（必做）

#### 1. 高铁车次信息管理

- 需要管理的信息包括：车次、始发站、终到站、发车时间、票价、运行情况（是否停运），及停靠车站、各站点进站时间、出站时间、里程（单位公里）等，还可以包括高铁运行时间、各种坐席数量、票价等
- 说明及要求：
  - ◆ 需要选择合适的数据结构存放高铁信息，包括逻辑结构和物理结构
  - ◆ 高铁车次不少于 10 趟
  - ◆ 高铁信息管理具有以下功能：
    - （1）增开一趟列车（插入）
    - （2）停开一趟列车（不是删除）
    - （3）高铁车次信息查询。至少能够根据车次、起点站、终到站、发车时间、停靠站进行查询
    - （4）可以修改除车次之外的信息
    - （5）从文件中读入高铁车次信息
    - （6）输出所有高铁信息
  - ◆ 高铁的票价按如下规则定价：商务座，每公里 1.2 元，一等座，每公里 0.6 元，二等座，每公里 0.46 元。如果不区分坐席等级，统一按二等座数量和价格管理。

#### 2. 乘客信息管理

- 数据项：证件类别、证件号、姓名、手机号、紧急联系人，紧急联系人电话等
- 说明及要求：
  - ◆ 乘客人数不少于 20 人
  - ◆ 乘客信息管理至少具有以下功能
    - （1）增加一个乘客（插入）
    - （2）删除一个乘客（删除）
    - （3）乘客信息查询。至少能根据姓名、证件号、手机号、紧急联系人等进行查询
    - （4）乘客信息修改（如手机号更新、紧急联系人更新等，不可更改证件号）
    - （5）从文件中获得乘客数据
    - （6）输出所有乘客信息

#### 3. 高铁订票信息管理

- 订票信息应包括订单号、乘车人姓名、乘车日期、起点站、终点站、证件号、开车时间、票价、座位号等，还可以包括订票时间、坐席等级等信息
- 说明及要求：
  - ◆ 车票可以预定，也可以退票
  - ◆ 只能预定 2 个月内的车票
  - ◆ 座位不记录车厢号，仅用序号（从 1 开始）表示

- ◆ 每个乘客一次只能预定一趟高铁的车票，可以最多为 2 位同行乘客订票
- ◆ 预定的车票在开车前 2 个小时可以免费退票，否则需要收取 5% 的退票费
- ◆ 每张车票起点站可以不是高铁的始发车站，终到站也可以不是终点站
- ◆ 系统中至少包含 50 个订票数据，至少有 3 张以上的车票出发日期、起点站、终点站相同
- ◆ 订票信息管理至少具有以下功能
  - (1) 订票（插入）
  - (2) 退票（删除）
  - (3) 车票信息查询。至少能根据姓名、证件号、乘车日期、车次等进行查询
  - (4) 余票查询（根据日期和车次，还可以根据日期和起点、终点站）
  - (5) 输出所有车票信息
  - (6) 打印一张车票（输出）

4. 给出的数据全部为模拟数据，仅用于本次练习，可以按需选用

## 二、 系统功能自行设计，只要合理都可以

## 三、 考核要求

- 1、要有分级菜单，程序运行简洁、可行，程序可读性强
- 2、注释不少于 30%，高铁、乘客、车票结构体的每个成员含义和主要函数的功能必须用注释说明
- 3、2 个同学一组，自由组合
- 4、所有同学录汇报视频（视频格式为 mp4，时长不超过 10 分钟），介绍程序功能、程序运行情况等
- 5、每个小组需要上交实验报告一份，包括
  - (1) 高铁、乘客、车票信息的物理结构
  - (2) 整个程序的框架结构、设计思路、主要函数的功能
  - (3) 订票程序的算法思路说明
  - (4) 自认为最难/最出彩的部分
  - (5) 主要功能运行截屏
  - (6) 小组成员分工等
- 6、各种方式的抄袭，均以零分计。

## 四、 其他要求

- 1、小组合作也是考核内容
- 2、重修数据结构与算法实验的同学必须完成本作业，需在 12 月 19 日前告知任课教师小组成员
- 3、已经有大作业成绩的重修同学可以不做大作业，但必须在 12 月 19 日前跟任课教师明确答辩成绩。
- 4、上交材料包括：源程序、数据文件、实验报告、汇报视频，打包在一个压缩文件中，压缩文件以小组两个成员学号的后四位命名
- 5、大作业提交截止时间：2023 年 1 月 8 日，提交邮箱：zhongljxxxxy@126.com

附：获得系统时间函数

```
#include<time.h>
typedef struct date
```

```

{
    int year,month,day;
}DATE;           //定义日期结构体
DATE Get_Sys_Time() //返回当前系统时间
{
    struct tm today; //存放时间的结构体
    time_t one;      //存放时间的类型
    DATE now;
    one=time(&one);   //获得系统时间
    today=*(gmtime(&one)); //将 time_t 格式系统时间转为 struct tm 格式
    now.year=today.tm_year-100+2000; //2011 年返回 111
    now.month=today.tm_mon+1;        //9 月返回 8;
    now.day=today.tm_mday;
    return now;
}

```