

计算机程序设计基础课程设计项目

机票预订系统

1. 系统描述：

分为管理员、乘客两种不同的用户。 管理员负责管理、创建、更新、删除航班，而乘客可以预订、退订航班。 要求实现一个简易的机票预订系统。

2. 系统数据：

该系统中的实体类描述如下，要求必须创建。属性类型可供参考。 另外，已给出的类与属性并无法满足整个系统设计的需要，请根据自己团队系统设计需求， 另行添加相应的类。

1) 管理员 Admin：系统的管理者。 使用前要求必须登陆

字段名	数据类型	说明
userName	String	用户名
password	String	密码

2) 乘客 Passenger：系统的使用者。 查询航班信息时可以不必登陆，查询自己订单、订票、退票需要登录，第一次登陆需要注册，用户名为身份证号，密码由乘客自己设置。

字段名	数据类型	说明
passengerID	int	乘客编号（唯一）
realName	String	真实姓名
identityID	String	身份证号
password	String	密码
orderList	ArrayList<Order>	订单列表

3) 航班 Flight： 由管理员创建

字段名	数据类型	说明
FlightID	String	航班号
startTime	String	起飞时间
arrivalTime	String	到达时间
startCity	City or String	起飞城市
arrivalCity	City or String	到达城市
departureDate	String	起飞日期
price	int	价格
currentPassengers	int	当前预定人数
seatCapacity	int	容量



flightStatus		FlightStatus or String		航班状态
passengerIDs	or	ArrayList<Integer>	or	旅客 ID 或对象列表
passengers		<Passenger>		

4) 订单 Order：旅客预订航班后产生

字段名		数据类型		说明
passengerID or passenger		Int or Passenger		旅客 ID 或 旅客对象
seat		String		座位号
flight		Flight		航班
createDate		Date		预定日期
status		OrderStatus	or	订单状态
		String		

3. 系统功能性需求：

以下是系统要求实现的最基本功能，然而这些功能是不够的，请小组讨论：如何添加其他辅助性功能，使系统顺畅运行。如何添加额外附加功能，来取得项目的加分项。

1) 创建航班 createFlight：管理员功能

- 创建航班时，需要填写有关航班的所有属性
- 确保起飞时间在当前时间 2 小时之后，到达时间在起飞时间之后。

2) 修改航班 updateFlight：管理员功能

- 对于航班状态的修改，需要系统自动完成（请参考 7)预订、退订 8) 航班功能描述）。航班起飞 2 小时前，航班状态自动变为终止（TERMINATE），管理员不可再修改终止状态的航班信息。
- 如果航班在发布前，可以任意修改
- 如果航班已发布，起飞城市、到达城市、起飞时间、起飞日期等重要信息不可修改，当前价格、容量等信息可以修改。

3) 删除航班 deleteFlight：管理员功能

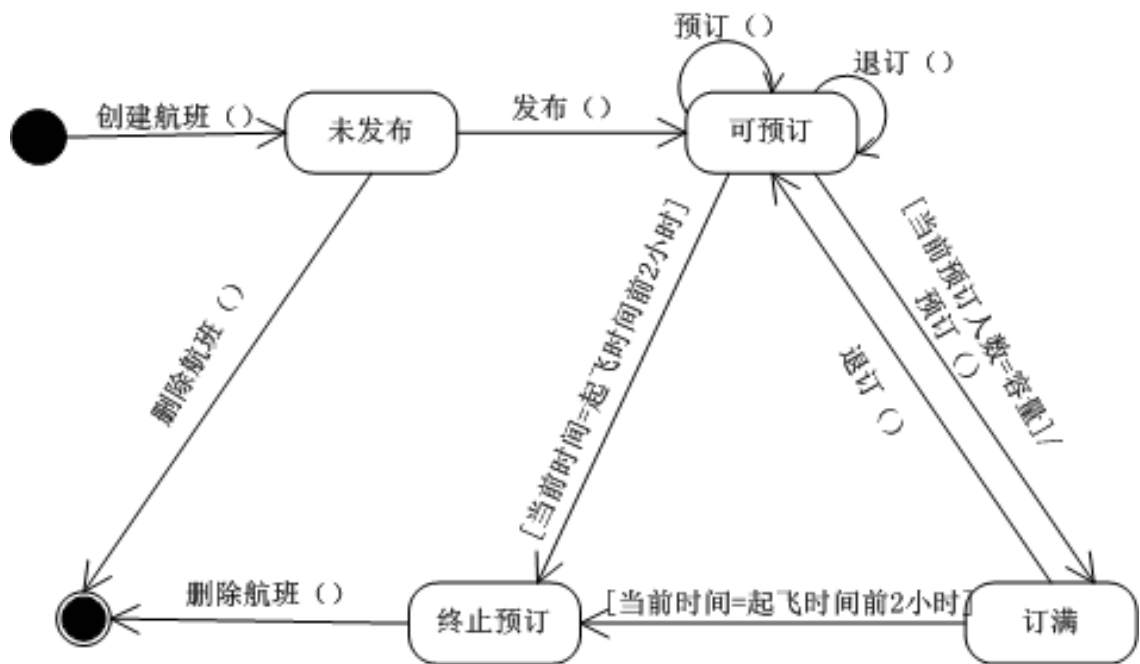
- 只有在未发布、终止状态，可以删除航班。

4) 高级查询 superQuery：管理员功能

- 查询某一航班所有的预订信息（包括乘客姓名、身份证号、座位号、预订时间以及订单状态）
 - 查询订单
 - 以及所有 6)查询航班的功能
- 5) 用户管理 `userManagement`：管理员功能
- 添加其他管理员用户
 - 更新自己的用户信息
- 6) 查询航班 `queryFlight`：管理员、乘客功能
- 可以通过起飞城市、到达城市、起飞日期查询航班，查询之后，系统可以列出所有符合条件的航班，及当前价格、航班状态
 - 可以通过航班号查询航班（可以模糊查找），查询之后，系统可以列出所有符合条件的航班，及当前价格、航班状态
 - 查询之后，乘客可以预订航班，需要密码验证
 - 查询乘客自己的订单列表，需要密码验证
- 7) 预订航班 `reserveFlight`: 乘客功能
- 如果航班为订满状态(FULL)，便不可再预订
 - 支付问题可简化。（例如，对于支付功能，可以系统提问是否支付，再手动输入一个“Y”表示已确认支付，并更改订单支付状态）
 - 每预订成功一个航班，系统需要自动进行相关的更新。（例如，当前预订人数+1）
- 8) 退订航班 `unsubscribeFlight`: 乘客功能
- 如果航班为订满状态(FULL)，退订操作后，其状态应该变为可预订状态(AVAILABLE)
 - 退订之后所执行的退款问题可简化。（例如一条输出语句：退款已返还）
 - 每退订成功一个航班，系统需要自动进行相关的更新。（例如，当前预订人数-1）

4. 项目要求：

- 1) 该项目在命令行中实现即可。
- 2) 要求完成在第 3 部分中提出的所有功能。
- 3) 每次程序运行前，要有一定的初始化数据信息：
 - 至少初始化 3 座城市
 - 每两个城市之间,每个方向至少 1 个航班(每一天至少 6 个航班)
 - 至少设置未来 3 天的航班（总共至少 18 个航班）
 - 航班状态分为 4 种，状态转化关系如下图所示，状态分为：未发布 UNPUBLISHED, 可预订 AVAILABLE, 订满 FULL, 终止 TERMINATE



- 订单状态至少分为3种：未支付 UNPAID，已支付 PAID，取消 CANCEL

4) 加分项：

并不代表简易添加一些功能、或使用 GUI、数据库等技术，就会额外加分，前提是必须设计的出色、合理、完整才能拿到额外的分数。

系统需求设计上：

- 添加中转站设计
- 添加往返功能设计
- 添加航空公司
- 设置航班与具体日期下该航班的区分
-

在技术应用上：

- 使用数据库
- 使用文件存储数据
- 图形界面设计
- 网页制作
-

5) 项目截止日期：5月21日晚11:00

答辩时间：5月22日—6月2日 各班上机课时间

5. 提示:

1) 控制台输出样例参考:

```
Please input the number:
1 Login for Admin, 2 Query Flight Information, 3 Query Order List, 4 Reserve Flight, 5 Unsubscribe Flight
2
Please input the number:
1 Query by Flight ID, 2 Query by start city, arrival city and date, 3 Return
2
```

2) 初始化数据:

以 Flight 数据为例, 仅供参考:

```
public class Admin {
    public static ArrayList<Flight> flights=new ArrayList<Flight>();
}

public static void initializeFlightData(){
    Flight f1=new
Flight("CZ1000","11:10:00","15:45:00","ShenZhen","Beijing","2017-5-15");
    Flight f2=new
Flight("CZ1001","10:30:00","12:20:00","Beijing","ShenZhen","2017-5-15");
    Flight f3=new
Flight("CZ1010","15:20:00","17:40:00","ShenZhen","ShangHai","2017-5-15");
    Admin.flights.add(f1);
    Admin.flights.add(f2);
    Admin.flights.add(f3);
    System.out.println("list all the flight ID whose start city is
ShenZhen");
    for(Flight f:Admin.flights){
        if(f.getStartCity().equals("ShenZhen")){
            System.out.println(f.getFlightID());
        }
    }
}

public static void main(String [] args){
    initializeFlightData();
}
```

Output:

```
list all the flight ID whose start city is ShenZhen
CZ1000
CZ1010
```