Florida Polytechnic University
Department of Computer Science

COP 3330C Computer Programming 2, Fall 2018

Project: Airline Reservation System

Due Date: Sunday, December 2 at 11:59 pm

Objectives

By completing this project, you will gain experience with the following:

- JavaFX (GUI)
- File input/output
- Exception handling
- ArrayList and 2D array
- Using Scanner for console input
- Defining classes
- Defining methods
- Loops
- Conditions

Project Description

In this project, you will develop a JavaFX application for an Airline Reservation System. The airline manages the information of several flights. The information of each flight consists of the following:

- flight number
- flight Date
- departure time
- arrival time
- departure city
- destination city
- number of available seats

The passenger seat reservation information consists of the following:

- passenger ID number (Driver License ID, passport ID, or ID)
- passenger name
- seat number
- flight number

A file named “flights.txt” contains the information of all flights of the airline. The file is stored in the format shown below [Figure 1].

Flight#	FDate	DTime	ATime	DepartCity	DestCity	AvailableSeats
AA1150	20Dec15	23:00	02:00	FORT WAYNE	ORLANDO	70
AA1230	5Nov15	11:30	14:00	BLACKSBURG	BOCA RATON	25
AA1140	4Jan15	07:00	11:00	SEATTEL	PHONEIX	42

Figure 1. Contents of flights.txt (fields are separated by Tabs, not spaces)

Another file named “reservations.txt” contains the information of all passenger reservations. The file is stored in the format shown below [Figure 2].

ID	Name	SeatNumber	Flight#
1035093287	George Eric	2A	AA1230
E8742784	Tom Ed	1C	AA1140
1035027864	Sara Shafer	4E	AA1140

Figure 2. Contents of reservations.txt (fields are separated by Tabs, not spaces)

Phase A:

When the application is started, it should display a Main Menu with seven items, as shown in Table 1 below.

Main Menu Item	Description
1. Add a new flight	a. Ask the user to enter data for a new flight b. Create new text file that contains the passengers’ seats map as described in Phase B. Name the file to be the flight number. c. Add flight information to the "flights.txt" file
2. Make a new reservation	a. Ask the user for the flight number in which seat is to be reserved b. Ask the user for new passenger ID and name c. Display passengers seats map as described in Phase B d. Ask the user to enter the seat number (e.g. 2A, 1C, 4D….etc) e. Update the number of available seats in this flight f. Add passenger reservation information to the "reservations.txt" file g. Update passengers’ seats map for this flight in the text file which has flight number as file name

3. Display passengers seats map	Display the seat pattern for a given flight number
4. List all flights	Display information of all flights
5. List all passenger reservations for all flights	Display information of all passenger reservations for all flights
6. List all passenger reservations for specific flight	Display information of all passenger reservations for a given flight number
7. Exit	Exit the program

Table 1. Main Menu items and their description

Phase B:

In this project, you should assign passengers seats in an airplane. Assume a small airplane with seat numbering as follows:

```
1     A B   C D E   F G
2     A B   C D E   F G
3     A B   C D E   F G
4     A B   C D E   F G
5     A B   C D E   F G
6     A B   C D E   F G
7     A B   C D E   F G
8     A B   C D E   F G
9     A B   C D E   F G
10    A B   C D E   F G
```

Figure 3. Initial passengers' seats in an airplane

The program should display the seat pattern, with an 'X' marking the seats already assigned. For example, after seats 1A, 2B, and 4C are taken, the display should look like:

```
1     X B   C D E   F G
2     A X   C D E   F G
3     A B   C D E   F G
4     A B   X D E   F G
5     A B   C D E   F G
6     A B   C D E   F G
7     A B   C D E   F G
8     A B   C D E   F G
9     A B   C D E   F G
10    A B   C D E   F G
```

Figure 4. Updated passengers' seats in an airplane

Analysis and Design

- You have to use JavaFX to design and create the GUI for the application. We will leave the GUI design of the application to you.
- You have to use object-oriented concepts, i.e. classes and methods. We will leave the design of the application to you.
- Your application starts by reading the data from two text files, namely “flights.txt” and "reservations.txt". Data in "flights.txt" should be stored in an **ArrayList of flight objects**. Similarly, data in "reservations.txt" should be stored in an **ArrayList of passenger objects**.
- In menu item # 1 “Add a new flight”, you should create a text file that contains the airplane seats as shown in Figure 3. The name of the file should be the flight number.
- In menu item # 2 “Make a new reservation”, you should display the seats available, ask the user to enter the seat desired, the user can type in a seat, and then the display of available seats should be updated as shown in Figure 4.
- If the user types in a seat that is already assigned, the application should say that the selected seat is occupied and ask for another choice.
- When the program reads data from other text files that contain the seats maps, the data for each seats map should be stored in **2D array of char objects of size 10x8.**
- **The program must be able to detect all possible errors and exceptions, e.g. entering a number of a nonexistent menu, giving a negative value for the number of available seats, etc.**
- You should always return to the menu after completing any operation. The application should **not** exit until the user chooses the “Exit” option.

Grading Policy

Activity	Weight
Algorithm, *i.e. your problem solving logic.*	20 pts
Use of Object Oriented Concepts	25 pts
Running Program	40 pts
Style; *i.e. good naming, use of methods, etc.*	10 pts
Documentation; *i.e. use of comments.*	5 pts
Total	100 pts

Submission Instructions

Zip up your project folder and upload the zip file to Canvas no later than the due date. In the main class, write the names of all team members as a comment. Only one member of the team can submit the project on Canvas.