Testat 3

Programming in Java

SS 2020

Learning objectives:

- **You deepen and consolidate your object-oriented thinking**
- **You build classes and you use objects**
- **You structure your program processes with suitable classes and methods**

Administration of basic data for an airplane caterer

Before starting your programming, create an UML class diagram and place it as a PDF file named `Testat-3_ClassDiagram.pdf` in the project folder `docu` (a folder as in Testat 1 and 2)

Basic data management is an important thing in all business applications. One situation is that a caterer offers various dishes and menus for airline companies. So, for example, Lufthansa might book some menus/dishes for a flight from Frankfurt to Tokyo.
For each flight you can book several menus/dishes, each in a certain quantity.
So here is given a n:m - relationship (one flight, several dishes and also: one dish, several flights)
Remember: First, you must build an ERD-diagram to understand the scenario.

Your task now is to build a first console program (without a graphical interface) that manages the dishes, the flights, the customers, and so on...
Your job: Take **two entity types**
that you have recognized and do the basic actions for them.

The first step of the console program is to display a main menu:

```
EAF: EAT&FLY-Management (Version 1.0 (C) 2020 by Group n)  (n= your
Group number ☺)
MAIN MENU
Please select:

Managing flights      1
Managing dishes       2

Program exit          0

Please enter your choice:  _
```

The cursor is positioned behind the text "`Please enter your choice:`"
and waits for the input of the user.
When the user enters 1 or 2, the program starts a corresponding submenu to serve the objects.

As long as you have not yet programmed a corresponding submenu, this message should appear:

"This program section is currently not implemented
Press ENTER to return to the menu"

When the user then presses any key, the program returns to the main menu.

If the user enters 0, the program terminates.
If a crazy user enters something else, the main menu is simply re-displayed without comment until user enters 0,1,2
So this menu is presented in a loop and will redisplay until user enters 0.
The submenus for objects look similar to the main menu.
Here you see as an example the flight's menu:

```
EAF: EAT&FLY-Management (Version 1.0 (C) 2020 by Group 13)

MANAGING FLIGHTS MENU
Please select:

Create new flight          1
Update a flight            2
Delete a flight            3
Show flight     list       4

Back to main menu          0

Please enter your choice:  _
```

When user enters 0, the program returns to the main menu to allow user to select other items to work on. If user enters 1, 2, 3 or 4, the program starts the respective method to do that job. Again:
As long as you have not yet programmed a method, this message should appear:

"This program section is currently not implemented
Press ENTER to return to the menu"

When the user then presses any key, the program returns to the **submenu**.

Each new flight is an object stored in an ArrayList. You will need 2 ArrayLists to handle these 2 types of objects. Each new flight has an ID, a name and some other necessary attributes that you need to program.
AGAIN: First, you should build up an ERD or better an UML-Class-Diagram to understand the relationships between your classes and their attributes and methods!

In a first step try to understand that you now need several classes:
A class for each entity type you want to handle with your program, for example class Flight

You also need a class Eaf with the main()- method in it. This method produces the main menu, asks for user input (0,1,2) and switches to the desired sub menu by calling the corresponding method.
Main and the 2 corresponding methods may be static.

Then you need a class FlightManager that handles flights. So it has the methods addFlight(), updateFlight(), deleteFlight() and listFlights(). You have to create an object of that class, that will handle all flights:

```
FlightManager myFlightManager = new FlightManager();
```

In the submenu this objects methods are called whenever user selects it.
For example, if user enters 3 in the flights submenu, then your program will call:
myFlightManager.deleteFlight();

to do that job.

It is th same with other jobs and other entity types.

Bonus task: (For participants seeking maximum points)

How could you filter the list displayed in item 4? With a well-thought-out filter, the program only displays the flights, dishes, … you have filtered, and not all of them!