COMP201- Object Oriented Programming

Spring 2017

HOMEWORK5

In this homework, you are asked to write an application for selling tickets for the busses travelling from Adana to Kayseri.

**Program Execution:**

The application reads the list of the passengers and the list of the travels from two different input files. The example input files are given under files/week6 directory. The names of the files are *“passengers.txt”* and *“travels.txt”*

In the passengers.txt file, id/name/surname information of each passenger is listed. In the travel.txt file, the index and the capacity of each travel is listed.

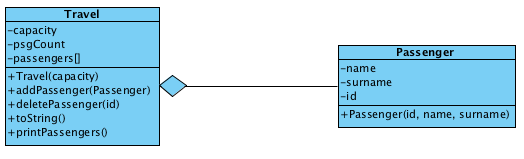
After reading passengers and travel capacities, the application lists the menu.

There are 4 menu elements, which are following:

1. Buy Ticket: The user enters the id of the passenger and the index of the travel. The system finds the passenger from the passenger array and adds the passenger to the travel with the given index.
2. Cancel Ticket: The user enters the id of the passenger and the index of the travel. The system finds the passenger from the travel with the given index and deletes the passenger from the travel.
3. List travels: The system lists all the travels and, if there is any passenger in the travel, it also lists the passengers.
4. Exit: The system terminates the application.

**Required Classes:**

The class diagram is given below:



Each travel has a capacity defined. Also, to be able to trace the number of sold tickets for the travel, psgCount variable is used. Finally, each passenger added to the travel is kept in the passengers[] array.

**Program Output:**

MENU

1:Buy Ticket

2:Cancel Ticket

3:List Travels

4: Exit

1

Passanger Id:111111

Travel Index:2

Passanger [name=Zulfiye, surname=Derin, id=111111] is added

MENU

1:Buy Ticket

2:Cancel Ticket

3:List Travels

4: Exit

1

Passanger Id:222222

Travel Index:2

Passanger [name=Irem, surname=Duran, id=222222] is added

MENU

1:Buy Ticket

2:Cancel Ticket

3:List Travels

4: Exit

1

Passanger Id:333333

Travel Index:2

The travel is FULL.

MENU

1:Buy Ticket

2:Cancel Ticket

3:List Travels

4: Exit

3

0.Travel [capacity=1, selledTickets=0]

1.Travel [capacity=10, selledTickets=0]

2.Travel [capacity=2, selledTickets=2]

Passanger [name=Zulfiye, surname=Derin, id=111111]

Passanger [name=Irem, surname=Duran, id=222222]

3.Travel [capacity=5, selledTickets=0]

4.Travel [capacity=3, selledTickets=0]

5.Travel [capacity=7, selledTickets=0]

6.Travel [capacity=4, selledTickets=0]

7.Travel [capacity=12, selledTickets=0]

Here is an example for the program output.

MENU

1:Buy Ticket

2:Cancel Ticket

3:List Travels

4: Exit

2

Passanger Id:222222

Travel Index:2

Passanger [name=Irem, surname=Duran, id=222222] is deleted

MENU

1:Buy Ticket

2:Cancel Ticket

3:List Travels

4: Exit

3

0.Travel [capacity=1, selledTickets=0]

1.Travel [capacity=10, selledTickets=0]

2.Travel [capacity=2, selledTickets=1]

Passanger [name=Zulfiye, surname=Derin, id=111111]

3.Travel [capacity=5, selledTickets=0]

4.Travel [capacity=3, selledTickets=0]

5.Travel [capacity=7, selledTickets=0]

6.Travel [capacity=4, selledTickets=0]

7.Travel [capacity=12, selledTickets=0]

MENU

1:Buy Ticket

2:Cancel Ticket

3:List Travels

4: Exit

4

**Programming Tips:**

* In the main method, you will need two arrays to be able to trace all the passengers in the system and also all the travels in the system.
* To be able to make the programmers life easier, the programmer directly enters the index of the travel so that the programmer can directly use that index.