Due Date: 21.10.2024 23:55

CENG211 – Programming Fundamentals
Homework #1

In this homework, you are expected to implement a "Ticket Booking Application" in Java.
You should fulfill the concepts of:

- Defining Classes
- CSV file I/O
- Arrays
- 2-dimensional Arrays
- Constructors, Getters & Setters

In the Ticket Booking Application, there are records of customer names and the number of booked tickets for each customer. Each customer books random tickets from one section of a venue. The venue has 4 sections. Each section is built as 10 rows and 60 seats (in each row). Therefore, each section has 600 tickets. Each ticket has a section number, row number, seat number, price, and booking status.

In this application, the pricing of the tickets is made randomly. The following are applied to ticket pricing:

- Each section has a maximum price and a minimum price.
- The price of the tickets in the first row is equal to the maximum price (of each section).
- The price of the tickets in the second row is equal to 80% of the maximum price (of each section).
- Other rows' tickets have **random prices** between minimum price and maximum price.

Additionally, each section's maximum price and minimum price are determined randomly. For Section 0, the boundaries of the random maximum price are [4000,5001), and the boundaries of the random minimum price are [3000,4000). Remember that '[' is inclusive, and ')' is exclusive. For each section, **the boundaries are decreased by 500,** and it goes on like this. All the boundaries are given in the below table. (In your application, please, calculate them in a **for or while loop**). Also, please pay attention that these are only the **boundaries**, so the actual maximum and minimum prices should be determined **randomly** in the application.

Section	Max Price Boundaries	Min Price Boundaries
Section 0	[4000,5001)	[3000,4000)
Section 1	[3500,4501)	[2500,3500)
Section 2	[3000, 4001)	[2000, 3000)
Section 3	[2500, 3501)	[1500, 2500)

In this application, customers book random tickets for a section. In the attached CSV file, each customer's name and the number of tickets that will be booked are given. The section of the customer will book tickets is also determined randomly. Each customer keeps the information of tickets that she/he booked.

In this homework, you are expected to implement the necessary classes to load the data from the given CSV file and create the desired queries. You are expected to implement classes for **Ticket, Section, Venue, Customer, Query, TicketBookingApplication (the class with main method),** and the helper class **FileIO** with the information given below:

Ticket:

- Section Number
- Row Number

- Seat Number
- Price
- Booking Status

Section:
- ID
- Number of Rows: 10
- Number of Seats (in each row): 60
- Max Price
- Min Price
- A two-dimensional (10x60) array that holds **Ticket** objects.

Venue:
- Number of Sections: 4
- A one-dimensional array of **Section** objects

Customer:
- Name
- Number of Booked Tickets
- A one-dimensional array that holds **Ticket** objects

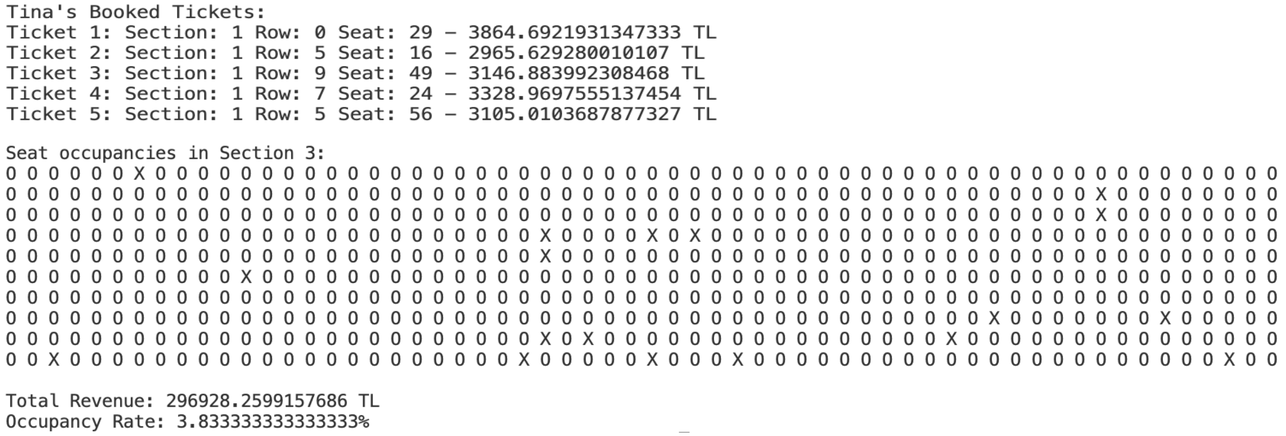
Implement necessary methods to respond to the following queries in **Query** class:

1- The section that has the highest revenue.
2- The total revenue of the venue.
3- The occupancy rate of the venue.
4- The tickets of the customer that pays the highest price for all of her/his tickets.
5- The most expensive ticket.

If the result of any of the queries is more than one, please display one result. Since almost all data are randomly determined, the results of your projects will differ each time you execute your code and, also from each other. This is completely normal.

In your application, try to determine the methods of your classes correctly and call these methods in your queries.

Also, please display **each section's occupancies** by printing the booked tickets by 'X' and the available tickets by 'O'. An example output is in the following, the output is given for **some** of the queries.



```
Tina's Booked Tickets:
Ticket 1: Section: 1 Row: 0 Seat: 29 – 3864.6921931347333 TL
Ticket 2: Section: 1 Row: 5 Seat: 16 – 2965.629280010107 TL
Ticket 3: Section: 1 Row: 9 Seat: 49 – 3146.883992308468 TL
Ticket 4: Section: 1 Row: 7 Seat: 24 – 3328.9697555137454 TL
Ticket 5: Section: 1 Row: 5 Seat: 56 – 3105.0103687877327 TL

Seat occupancies in Section 3:
O O O O O O X O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O
O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O X O O O O O O O O
O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O X O O O O O O O O
O O O O O O O O O O O O O O O O O O O O O O O O O X O O O O X O X O O O O O O O O O O O O O O O O O O O O O O O O O O O
O O O O O O O O O O O O O O O O O O O O O O O O O X O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O
O O O O O O O O O O O X O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O
O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O
O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O X O O O O O O O X O O O O O
O O O O O O O O O O O O O O O O O O O O O O O O O X O X O O O O O O O O O O O O O O O O X O O O O O O O O O O O O O O O
O O X O O O O O O O O O O O O O O O O O O O O O X O O O O O X O O O X O O O O O O O O O O O O O O O O O O O O O O X O O

Total Revenue: 296928.2599157686 TL
Occupancy Rate: 3.833333333333333%
```

Important Notes:

1. Do NOT request inputs in your app. Printing the results of the queries will be enough. You should print names of the results instead of printing IDs or indices.

2. You are NOT allowed to use **List** / **ArrayList** interfaces in this homework.

3. You can use standard **java.io** packages to read files. Do NOT use other 3rd party libraries.

4. You should use **relative** paths (e.g. Files/sample.csv) instead of **absolute** paths (e.g. C:\\user\\eclipse-workspace\\MyProject\\Files\\sample.csv). Please be sure of it, otherwise there will be **no output** of your application and you certainly will **lose points**.

5. To support **Turkish characters,** you may need to change your project's text file encoding to UTF8: Right click on your project (in package explorer) → Properties → Text file encoding → Other → UTF8 → Apply.

6. You are expected to write clean, readable, and tester-friendly code. Please try to maximize reusability and prevent from redundancy in your methods.

References

Assignment Rules:

1. In this lecture's homework, there is no cheating allowed. If any cheating has been detected, they will be graded as 0 and there will be no further discussion on this.
2. You are expected to submit your homework in groups. Therefore, only one of you will be able to submit your homework.
3. Make sure you export your homework as an Visual Studio Code Java project. You can use other IDEs as well, however, you must test if it **can be executed** in Visual Studio Code.
4. Submit your homework through Microsoft Teams.
5. Your exported **Java Project** should have the following naming format with your assigned group ID (which will be announced on MS Teams) as the given below:

   G05_CENG211_HW1

   Also the **zip folder** that your project in should have the same name

   G05_CENG211_HW1.zip

6. Please beware that if you do not follow the assignment rules for exporting and naming conventions, you will lose points.
7. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.