CS 2336-504

PROJECT 0 – Disney's Jungle Cruise Ticket Reservation System

Project Due: January 31st by 11:55 PM

KEY ITEMS: Key items are marked in red. Failure to include or complete key items will incur additional deductions as noted beside the item.

Submission and Grading:

- **The file containing `main` must be named Main.java. (-5 points)**
- **The project files must be in a package named `Tickets`. (-5 points)**
- All project deliverables are to be submitted in eLearning.
- Zip the contents of the `src` directory into a single zipped file
    - Make sure the zipped file has a .zip extension (not .tar, .rar, .7z, etc.) (-5 points)
    - Please review the submission testing information in eLearning on the Course Homepage
- Projects submitted after the due date are subject to the late penalties described in the syllabus.
- Programs must compile and run with JDK 8.
- Each submitted program will be graded with the rubric provided in eLearning with the same set of test cases. Each student is responsible for developing sample test cases to ensure the program works as expected.
- **Type your name and netID in the comments at the top of all files submitted.** (-5 points)

Objectives: Create a Java program using programming fundamentals (file I/O, loops, conditional statements, arrays, functions)

Problem: In preparation for the release of Disney's Jungle Cruise movie next summer, you have been hired by the owner of a very small movie theater to develop the backend for an online ticket reservation system. The program should display the current seating arrangement and allow the patron to select seats. To simplify, you may assume that there is one auditorium. A report should be generated at the end of the program to specify for the auditorium how many seats were sold/unsold and how much money was earned.

Details

- **To avoid potential errors when grading, do not create multiple Scanner objects for System.in. If the Scanner object is needed in multiple functions, please pass the Scanner object into the function.**
- The seating arrangement for the auditorium will be stored in a file named *A1.txt*
- Each line in the file will represent a row in the auditorium. The number of rows in the auditorium is unknown to you.
    - There will be a newline character after each line in the file except for the last line which may or may not have a newline character.
- The number of seats in each row of the auditorium will be the same.
- No row will have more than 26 seats.

- There will not be more than 10 rows.
- The auditorium will be held in a two-dimensional array. (-5 points if not)
- Empty seats are represented by a period (.).
- Reserved seats are represented by a letter (A, C or S) **in the file**
    - This will be used to create reports
    - A =adult
    - C = child
    - S = senior
- Reserved seats will be represented by a hashtag (#) **on the screen**
    - The user does not need to know what type of ticket was sold, just that a seat is reserved.
- There is no need to worry about multiple screenings or reserving seats on a specific day.
- Ticket prices are as follows:
    - Adult - $10
    - Child - $5
    - Senior - $7.50

User Interface and Input: Present a user-friendly menu system for the user.

```
1. Reserve Seats
2. Exit
```

Loop the menu until the user decides to quit. Imagine this is for a ticket kiosk in the lobby of the theater.

If the user wants to reserve seats, display the current seating availability for that auditorium. An example seating chart is provided below for an auditorium with 5 rows and 20 seats per row.

```
  ABCDEFGHIJKLMNOPQRST
1 ...##..#####........
2 ########....####..##
3 .........##.........
4 #.#.#.#.#.#.#.#.#.#.
5 ########.#####.#####
```

The rows are numbered and the seats are identified in each row by a letter of the alphabet

After the seating chart has been displayed, prompt the user for the following information in the order below:

- Row number
- Starting seat letter
- Number of adult tickets
- Number of child tickets
- Number of senior tickets

Assume that the user wants to reserve sequential seats to the right of the first seat entered. Adult tickets will be reserved first, followed by child and then senior. All seats must be open for a reservation to be processed.

If the desired seats are not available, offer the user the best available seats that meet their criteria **on that row only**. The best available seats are the seats closest to the middle of the row measured by the distance from the center of the selection to the middle of the row. Prompt the user to enter a **Y** to reserve the best available or **N** to refuse the best available. Once the selection has been processed, return to the main menu. If there are no alternate seats available, display an appropriate message to the user instead of a prompt and return to the main menu.

All input will be of the valid data type. You do not have to worry about the user entering a letter when a number is expected or a floating-point number when an integer is expected. You are responsible for validating that the data falls within the proper range and that a user does not try to reserve a seat that is already reserved. If the user's selection extends past the end of a row, consider this invalid.

User Interface Workflow: Please do not add extra prompts since this will cause a mismatch in the input which will either force the program to throw an exception or cause the program to perform an unintended operation.

- Display main menu
- Prompt for input
- If user is reserving tickets
    - Prompt for row
    - Validate – loop until valid
        - Valid row = row number listed in auditorium display
    - Prompt for starting seat
    - Validate – loop until valid
        - Valid seat = seat number listed in auditorium display
    - Prompt for number of adult tickets
    - Validate – loop until valid
        - Valid ticket number = number >= 0
    - Prompt for number of child tickets
    - Validate – loop until valid
        - Valid ticket number = number >= 0
    - Prompt for number of senior tickets
    - Validate – loop until valid
        - Valid ticket number = number >= 0
    - If seats unavailable
        - Display best available
        - Prompt user to reserve (Y/N)
    - If reserved, confirm reservation
    - Return to main menu
- Loop to top of workflow until user selects exit

Output: At the end of the program, write the current status of the auditorium back to the file. Remember to write the auditorium reservations using A, C and S to identify the type of ticket sold. Also, display a formatted report to the console. Make sure each column lines up properly (no jagged columns). Include the following information in the order given:

- Total Seats in Auditorium
- Total Tickets Sold
- Adult Tickets Sold
- Child Tickets Sold
- Senior Tickets Sold
- Total Ticket Sales – total amount of money collected for tickets in the auditorium

All values except total ticket sales will be an integer value. Total ticket sales will be a floating-point value rounded to 2 decimal places and formatted with a dollar sign before the first digit of the number.