**CMSC203 Project 5 Implementation**

Class: CMSC203 CRN 31648

Program: Project 5

Instructor: Gary Thai

Summary of Description: This assignment required me to create a property management company which manages individual properties they will build to rent, and charges them a management fee as the percentages of the monthly rental amount.

Due Date: 04/07/2023

Integrity Pledge: I pledge that I have completed the programming assignment independently.

I have not copied the code from a student or any source.

**Part 1: UML Diagrams**

|  |
| --- |
| **TwoDimRaggedArrayUtility** |
| **- MAX\_ROW: int = 10**  **- MAX\_COLUMN: int = 10** |
| **+ readFile(file: File): double[][]**  **+ writeToFile(data: double[][], outputFile: File): void**  **+ getTotal(data: double[][]): double**  **+ getAverage(data: double[][]): double**  **+ getRowTotal(data: double[][], row: int): double**  **+ getColumnTotal(data: double[][], col: int): double**  **+ getHighestInRow(data: double[][], row: int): double**  **+ getHighestInRowIndex(data: double[][], row: int): int**  **+ getLowestInRow(data: double[][], row: int): double**  **+ getLowestInRowIndex(data: double[][], row: int): int**  **+ getHighestInColumn(data: double[][], col: int): double**  **+ getHighestInColumnIndex(data: double[][], col: int): int**  **+ getLowestInColumn(data: double[][], col: int): double**  **+ getLowestInColumnIndex(data: double[][], col: int): int**  **+ getHighestInArray(data: double[][]): double**  **+ getLowestInArray(data: double[][]): double** |

|  |
| --- |
| **HolidayBonus** |
| **- HIGHEST\_SALES: double = 5000.0**  **- OTHER\_SALES: double = 2000.0**  **- LOWEST\_SALES: double = 1000.0** |
| **+ calculateHolidayBonus(data: double[][]): double[]**  **+ calculateTotalHolidayBonus(data: double[][]): double** |

**Part 2: Comprehensive Test Plan**

A good test plan should be comprehensive. This means you should have a few test cases that test when the input is in and out of range, division by 0, incorrect Data type, etc. (Provide valid and invalid input)

HolidayBonusGFATest.java was successful for all available test cases.

Text

Description automatically generated

TwoDimRaggedArrayUtilityGFATest.java was successful for all available test cases.

Text

Description automatically generated

HolidayBonusTestStudent.java was successful for all available test cases.

Text

Description automatically generated

TwoDimRaggedArrayUtilityTestStudent.java was successful for all available test cases.

Text

Description automatically generated

**Part 3: Examples of the Code Running**

When application starts:

A picture containing calendar

Description automatically generated

File containing sales data (district3.txt):

Text, letter

Description automatically generated

Result after selecting Load Sales Data (district3.txt):

A picture containing timeline

Description automatically generated

File containing sales data (district4.txt):

Text, letter

Description automatically generated

Result after selecting Load Sales Data (district4.txt):

A picture containing graphical user interface

Description automatically generated

File containing sales data (district5.txt):

Text, letter

Description automatically generated

Result after selecting Load Sales Data (district5.txt):

A picture containing timeline

Description automatically generated

Additional input file containing sales data (personalDataSet1.txt):

Text

Description automatically generated

Result after selecting Load Sales Data (personalDataSet1.txt):

A picture containing graphical user interface

Description automatically generated

Additional input file containing sales data (personalDataSet2.txt):

Text, letter

Description automatically generated

Result after selecting Load Sales Data (personalDataSet2.txt):

Calendar

Description automatically generated with medium confidence

**Part 4: Github Repository S****creenshot**

**Lessons Learned** <Provide answers to the questions listed below>**:**

Write about your Learning Experience, highlighting your lessons learned and learning experience from working on this project.

What have you learned?

Through this project, I learned a lot about property management companies and how they function. Getting to work with classes to build a property management company was fun because I got to put my code towards something that is seemingly useful in areas that I wouldn’t have previously considered.

What did you struggle with?

I found that this project was rather difficult to understand with the instructions being somewhat jumbled. I had difficulties trying to understand what was being asked of me while reading through the provided documents. It did not help that most of the document had strikethroughs and some parts were misplaced. It felt as though it were a previous version history of the document rather than the finished version. This gave me some difficulties with trying to complete the documentation which I did to the best of my ability.

What would you do differently on your next project?

I really wanted to make another method for the addProperty method in the ManagementCompany class that would do the return -1,-2.-3.-4 part when called so that I would only have to write it once rather than for all 3 overloaded methods. If I were to repeat this project, that is definitely something I would look into.

What parts of this assignment were you successful with, and what parts (if any) were you not successful with?

I was successful with all parts of the assignment, except for the issues I had with having to compile the code through Command Prompt for JavaFX to work properly.

Provide any additional resources/links/videos you used to while working on this assignment/project.

**Check List:** <Provide answers to the column Y/N or N/A >**:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment#\_Moss.zip | **Y** |  |
|  | * FirstInitialLastName\_Assignment#.docx/.pdf | **Y** |  |
|  | * Source java files | **Y** |  |
|  | **Program compiles** | **Y** |  |
|  | **Program runs with desired outputs related to a Test Plan** | **Y** |  |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **Y** |  |
|  | * Screenshots related to the Test Plan | **Y** |  |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) | **Y** |  |
|  | * UML Diagram (if required) | **Y** |  |
|  | * Algorithms/Pseudocode (if required) | **N/A** |  |
|  | * Flowchart (if required) | **N/A** |  |
|  | * Lessons Learned | **Y** |  |
|  | * Checklist is completed and included in the Documentation | **Y** |  |