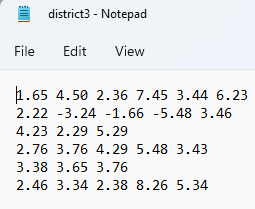
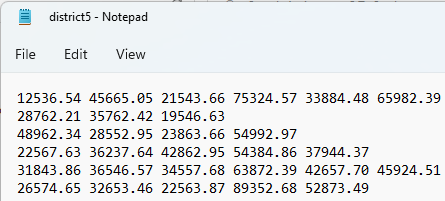
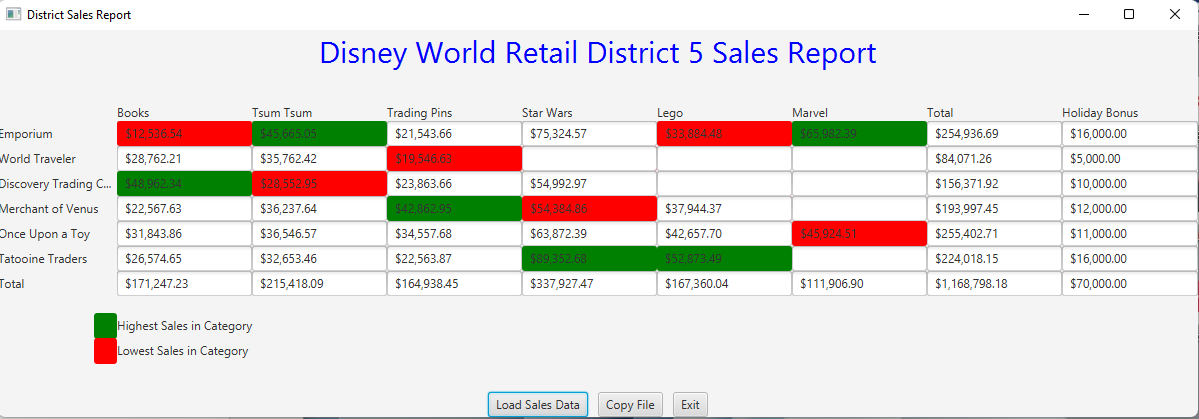
Assignment 5

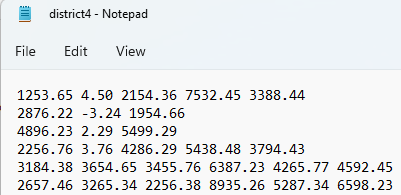
**Screenshots**

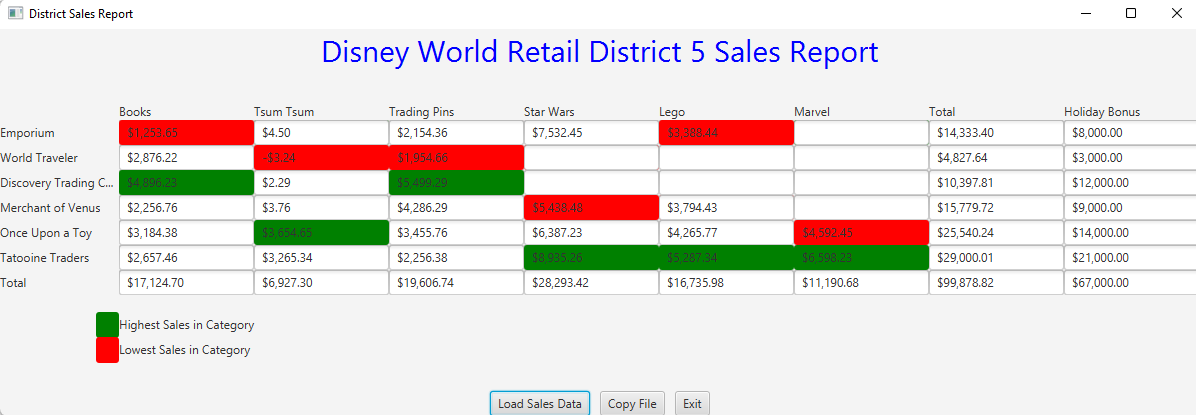




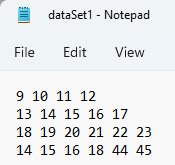




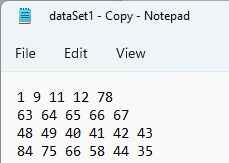




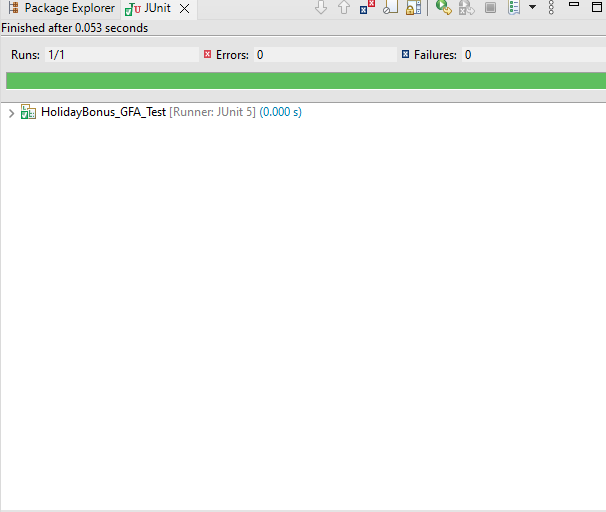
Additional Data

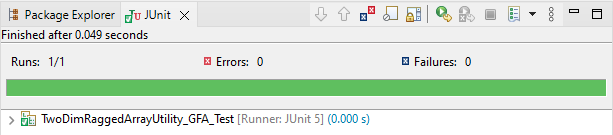


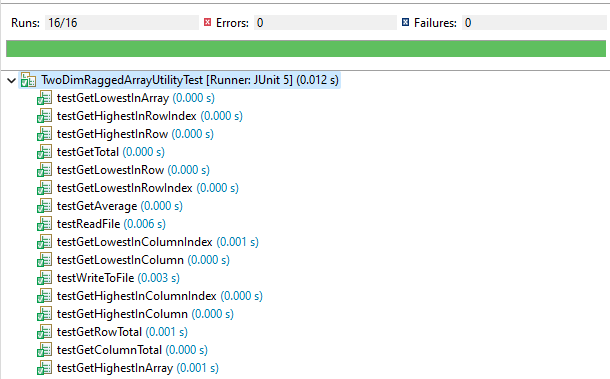


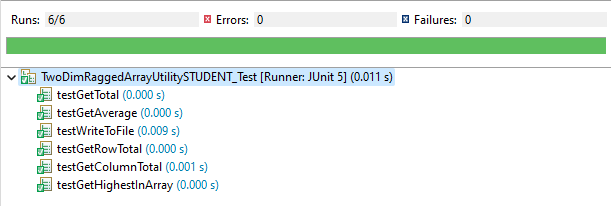


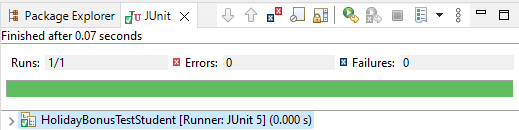




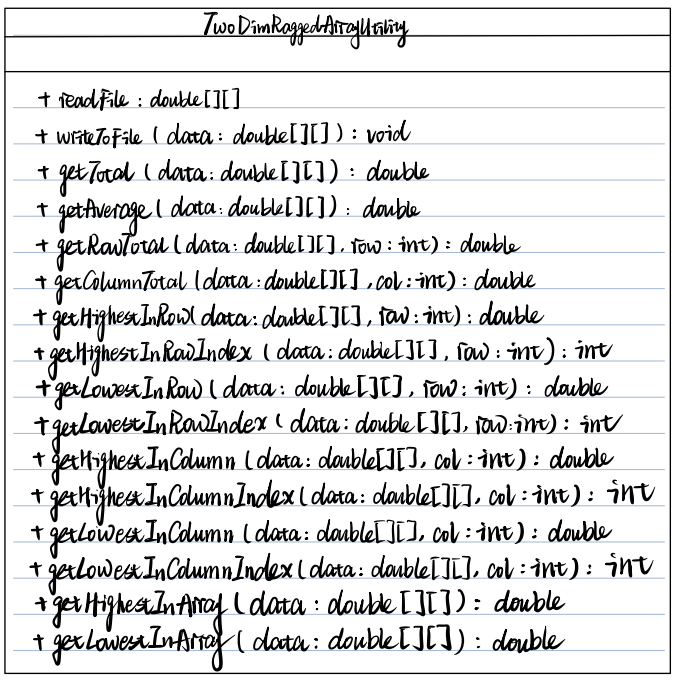


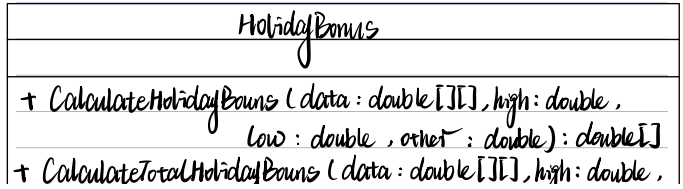






**UML Diagram**







**Algorithms/Pseudocode**

Create an instance of the File object

Open the file

Read from the file

If the file has next String

Read the numbers and store them in a ragged of array double

Return the ragged of array double

Create an instance of the PrintWriter object

If the number of the row is less than 10

If the number of the column is less than 10

Read the String from the File and Store them in the double array

End if

End if

Return the array

If the number of the row is less than the length of the row

If the number of the column is less than the length of the column

Declare a double total

Add up the numbers

End if

End if

Return total

Declare a double average

Divide total by the number of elements

Return average

If the number of the row is less than the length of the row

Declare a double total

Add up the numbers

End if

Return total

If the number of the column is less than the length of the column

Declare a double total

Add up the numbers

End if

Return total

Declare a double highest, and assigns it to the first element of the row

Declare an int index, and assigns it to zero

If index is smaller than the length of the row

If the value of the array[index] is larger than highest

Assign highest to array[index]

End if

End if

Return highest

Declare an integer index

Get the highest number from the previous method

Assign the index to the location of the number

Return index

Declare a double lowest, and assigns it to the first element of the row

Declare an int index, and assigns it to zero

If index is smaller than the length of the row

If the value of the array[index] is lowest than highest

Assign lowest to array[index]

End if

End if

Return lowest

Declare an integer index

Get the highest number from the previous method

Assign the index to the location of the number

Return index

Declare a double highest, and assigns it to the first element of the column

Declare an int index, and assigns it to zero

If index is smaller than the length of the column

If the value of the array[index] is larger than highest

Assign highest to array[index]

End if

End if

Return highest

Declare an integer index

Get the highest number from the previous method

Assign the index to the location of the number

Return index

Declare a double lowest, and assigns it to the first element of the column

Declare an int index, and assigns it to zero

If index is smaller than the length of the column

If the value of the array[index] is lowest than lowest

Assign lowest to array[index]

End if

End if

Return lowest

Declare an integer index

Get the highest number from the previous method

Assign the index to the location of the number

Return index

Declare a double highest, and assigns it to the first element of the row

Declare an int index, and assigns it to zero

If index is smaller than the length of the row

Declare an int index\_1, and assigns it to zero

If index\_1 is smaller than the length of the column

If the value of the array[index] is larger than highest

Assign highest to array[index][index\_1]

Increase index\_1 by one

End if

Increase index by one

End if

End if

Return highest

Declare a double lowest, and assigns it to the first element of the row

Declare an int index, and assigns it to zero

If index is smaller than the length of the row

Declare an int index\_1, and assigns it to zero

If index\_1 is smaller than the length of the column

If the value of the array[index] is smaller than lowest

Assign lowest to array[index][index\_1]

Increase index\_1 by one

End if

Increase index by one

End if

End if

Return lowest

Declare the getHighestInRowIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Declare the getHighestInColumnIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Declare the getLowestInRowIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Declare the getLowestInColumnIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Add getHighestInRowIndex and getLowestInRowIndex

Add getHighestInColumnIndex and getLowestInColumnIndex

Declare the getHighestInRowIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Assigns the rest of the data to 2000

Declare the getHighestInColumnIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Assigns the rest of the data to 2000

Declare the getLowestInRowIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Assigns the rest of the data to 2000

Declare the getLowestInColumnIndex method from the TwoDimRaggedArrayUtility and assigns it to 5000

Assigns the rest of the data to 2000

Add getHighestInRowIndex and getLowestInRowIndex and the rest of the 2000

Add getHighestInColumnIndex and getLowestInColumnIndex and the rest of the 2000

**Lesson Learned**

Since it’s been a while since the time we learned Junit, it took me a while to go back and go over the use of Junit Test. So it’s vital for me to keep everything in mind so that it would be easier for me to do the assignments.

I will say the most difficult part of doing this program are the methods related to column. With the help of tutor, I learned that I can not directly assign data[0][col] to highest since I have to make sure there must be a number in there, thus, I have to use try catch method to assign the first element of that column to highest.

Since there are some circumstances that only one data in the category, so I have to assign that number to highest which means that store will receive the highest bonus, in order for that to work successfully, I have to write if(data[i][j] == TwoDimRaggedArrayUtility.getHighestInColumn(data, j)&&data[i][j] == TwoDimRaggedArrayUtility.getLowestInColumn(data, j)) {HolidayBonus[i][j] = high;}, otherwise, the output won’t be successful

**Assignment 5 Check List (include Yes/No or N/A for each item)**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N or N/A** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment5\_Moss.zip | Yes |  |
|  | * FirstInitialLastName\_Assignment5\_Complete.zip | Yes |  |
|  | **Program compiles** | Yes |  |
|  | **Program runs with desired outputs related to a Test Plan** | Yes |  |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | N/A |  |
|  | * Screenshots for each Junit Test | Yes |  |
|  | * Screenshots for each Test case listed in the Test Plan | Yes |  |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) |  |  |
|  | * UML Diagram | Yes |  |
|  | * Algorithms/Pseudocode | Yes |  |
|  | * Flowchart (if required) | N/A |  |
|  | * Lessons Learned | Yes |  |
|  | * Checklist is completed and included in the Documentation | Yes |  |