**Psuedocode**

**TwoDimRaggedArrayUtility**

**readFile**

**CREATE new double array**

**CREATE new scanner**

**FOR loop to iterate through new double array and set to null**

**DECLARE int**

**WHILE file has next line**

**Split with a space**

**CREATE new double array**

**For Loop to iterate through row**

**Set new Array equal to length**

**FOR loop to iterate through row**

**FOR loop to iterate through length**

**Return read file**

**writeToFile**

**DECLARE printwriter**

**FOR loop to iterate through file**

**Write lines to output file**

**getTotal**

**DECLARE total var**

**FOR loop to iterate through data row**

**FOR loop to iterate through data column**

**Return total**

**getAverage**

**DECLARE total**

**DECLARE number of elements**

**DECLARE average**

**FOR loop to iterate through data row**

**FOR loop to iterate through data column**

**Total = total + element in data**

**Number of elements = number of elements + 1**

**Average = total / number of elements**

**Return average**

**getRowTotal**

**DECLARE total**

**FOR loop to iterate through row**

**Total = total + data in row**

**RETURN total**

**getColumnTotal**

**DECLARE total**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**Total = total + data in col**

**RETURN total**

**getHighestInRow**

**declare max var**

**FOR loop to iterate through row**

**IF element is greater than previous element**

**Current element = max**

**Return max**

**getHighestInRowIndex**

**declare max var**

**FOR loop to iterate through row**

**IF element is greater than previous element**

**Current element index = max**

**Return max**

**getLowestnRow**

**declare min var**

**FOR loop to iterate through row**

**IF element is less than previous element**

**Current element = min**

**Return min**

**getLowestInRowIndex**

**declare min var**

**FOR loop to iterate through row**

**IF element is less than previous element**

**Current element index = min**

**Return min**

**getHighestInCol**

**declare max var**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**IF element is greater than previous element**

**Current element = max**

**Return max**

**getHighestInColIndex**

**declare max var**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**IF element is greater than previous element**

**Current element index = max**

**Return max**

**getLowestInColumn**

**declare min var**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**IF element is less than previous element**

**Current element = min**

**Return min**

**getLowestInColumnIndex**

**declare min var**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**IF element is less than previous element**

**Current element index = min**

**Return min**

**getLowestInArray**

**declare lowest var**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**IF element in double array is less than previous element**

**Current element = lowest**

**Return lowest**

**getHighestInArray**

**declare highestt var**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**IF element in double array is greater than previous element**

**Current element = highest**

**Return highest**

**HolidayBonus**

**calculateHolidayBonus**

**create new array and set equal to length of data**

**create max var**

**create max index var**

**create min var**

**create min index var**

**create col var**

**create maxCols var**

**FOR loop to iterate through length of data**

**if data at index length is greater than maxCols set equal to maxCols**

**do while loop**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**if data greater than max set equal to max**

**if data less than min set equal to min**

**calculateTotalHolidayBonus**

**create new array and set equal to length of data**

**create max var**

**create max index var**

**create min var**

**create min index var**

**create col var**

**create maxCols var**

**do while loop**

**FOR loop to iterate through row**

**FOR loop to iterate through col**

**if data greater than max set equal to max**

**if data less than min set equal to min**

**if data col < maxCols**

**maxIndex = maxIndex + 5000**

**minIndex = minIndex + 1000**

**for loop to iterate through row**

**for loop to iterate through col**

**if col < length**

**array row = array row + 2000**

**while col < 6**

**create total var**

**for loop to iterate through row**

**total = data in array row + total**

**return total**

**Test Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Actual Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | Emp: 1.50 2.50 3.50 6.50 7.50  World traveler: 4 5 6 7 8 9  Discovery trading company:  100 200 300 400 500 600  Merchant of venus:  -1 -2 5 6 7 8  Once upon a toy:  10 2 4 8 6 12  Tatooine traders:  3 7 4 1 8 5 |  | EMP:  Lowest: 1.50  Highest: 7.50  Total: 21.50  World Traveler:  Lowest: 4  Highest: 9  Total: 39  Discovery:  Lowest: 2100  Highest: 100  Total: 600  Merchant of Venus:  Lowest: -2  Highest: 8  Total: 23  Once Upon A toy:  Lowest: 2  Highest: 12  Total: 42  Tatooine:  Lowest: 1  Highest: 8  Total: 28 |  |  |
| 2 | Emp: -3 -2 -1 0 1 2  World traveler: 5 10 15 20 25 30  Discovery trading company:  6 5 4 3 2 1  Merchant of venus:  -1.5 -1.0 -0.5 0 0.5 1.0  Once upon a toy:  10 20 30 40 50 60  Tatooine traders:  50 5 10 20 25 30 |  | EMP:  Lowest: -3  Highest: 2  Total: -3  World Traveler:  Lowest: 5  Highest: 30  Total: 105  Discovery:  Lowest: 1  Highest: 6  Total: 21  Merchant of Venus:  Lowest: -1.5  Highest: 1.0  Total: -1.5  Once Upon A toy:  Lowest: 10  Highest: 60  Total: 210  Tatooine:  Lowest: 5  Highest: 50  Total: 140 |  |  |
| 3 | Emp: 10 9 8 7 6 5  World traveler: 1 10 20 30 4 5  Discovery trading company:  9 10 11 12 13 14  Merchant of venus:  0 0 0 0 0 0  Once upon a toy:  100 100 100 100 100 100  Tatooine traders:  5 10 50 45 20 30 |  | EMP:  Lowest: 5  Highest: 10  Total: 45  World Traveler:  Lowest: 1  Highest: 30  Total: 70  Discovery:  Lowest: 9  Highest: 14  Total: 69  Merchant of Venus:  Lowest: 0  Highest: 0  Total: 0  Once Upon A toy:  Lowest: 100  Highest: 100  Total: 600  Tatooine:  Lowest: 5  Highest: 50  Total: 160 |  |  |

**Screenshot for JunitTest**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**\*\*Screenshot for GUI is missing because JavaFX was not able to be added my macbook even after getting help from the tech center.\*\***

**Lessons Learned**

Assignment 5 in CMSC203 was very informational and taught me how to calculate holiday bonuses. The project involved using arrays and I became very familiar and got to practice using not only 1-d arrays but also 2-d arrays and how useful they can be. I used for loops and nested for loops to iterate through the arrays and I worked with both the rows and cols and their corresponding indexes. Overall I really developed my skills with arrays, loops, and coding syntax in general and I am glad I completed this project.

**Github Screenshot**

**Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N or N/A** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment5\_Moss.zip | **Y** |  |
|  | * FirstInitialLastName\_Assignment5\_Complete.zip | **Y** |  |
|  | **Program compiles** | **Y** |  |
|  | **Program runs with desired outputs related to a Test Plan** | **Y** |  |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **Y** |  |
|  | * Screenshots for each Junit Test | **Y** |  |
|  | * Screenshots for each Test case listed in the Test Plan | **Y** |  |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) | **N/A – GUI not included so test plan screenshots not included but still work** |  |
|  | * UML Diagram | **Y** |  |
|  | * Algorithms/Pseudocode | **Y** |  |
|  | * Flowchart (if required) | **N/A** |  |
|  | * Lessons Learned | **Y** |  |
|  | * Checklist is completed and included in the Documentation | Y |  |