Running Head: 2D Linear Data Set 1

2D Linear Data Set

Isaak Henry

Columbia Gorge Community College

Running Head: 2D Linear Data Set 2

2D Linear Data Set

**Introduction**

This problem that this program will solve is the need to print an image correctly, while keeping the picture dynamic in case of change. This program will solve this problem by creating a 2D array, printing the 2D array to create a 2D image, use logic to dynamically fill the array, create a nested loop, fill the array with the proper amount of \*’s.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** |  | **Functional Requirement** |  | **Value** | **Stakeholder** |
| FID001 | **I want to** | Create a 2D Array | **so that** | there is an array to fill in multiple directions | Student |
| FID002 | **I want to** | Create a nested loop | **so that** | It’s possible to access the 2D array. | Student |
| FID003 | **I want to** | Use logic to dynamically fill the array | **so that** | Not every aspect of the 2D array has to be hard coded. | Student |
| FID004 | **I want to** | Fill the 2D array with the proper amount of \*’s | **so that** | The array can match the image given in the assignment | student |
| FID004 | **I want to** | Print the array to the console | **So that** | A 2D image of the array is presented that matches the picture. | Student |
| FID004 | **I want to** | Make sure the console doesn’t close. | **So that** | The instructor may see the result of the code | Instructor |

These are the functional requirements for this program, there are many ways to go about implementing these functional requirements, but the easiest way is to break each problem down into smaller steps.

**Analysis**

For this program, there are a few design choices that needed to be made. Abstraction will be used, a class called TwoDArray will be made so that the complexity of the MainEntry.java is as low as possible. The picture that is going to be printed will be easier to work with if cut in two. The first functional requirement will require a function that will call two other functions, each which will fill out part of the image. This function may be public as it is only calling two other functions. The second functional requirement will be to create a nested loop of two for() loops to access each point of a 2D array. There will be two of these loops as there will be a function for each half of the image, both will be private, as to encapsulate the functions from being tampered with by the MainEntry class.

Running Head: 2D Linear Data Set 3

For the left half of the image, the loop will logically go through each space using less than and an if else loop to examine the first 8x8 array, placing an asterisk in the correct locations by logically comparing the location of each intersection of column and row in the array. The second half of the image will have a similar function, comparing locations of each intersection logically using logical less than, greater than, along with logical and within a line of if else loops, examining each intersection of the array to put an asterisk in the correct place. With these two functions finished, a print function can be made, which will use a nested for() loop with System.out.print(arr[r][c]) attached to the final for loop to print out each asterisk and System.out.println() outside of the second for loop to reset the cursor drawing each loop through. In the main class, since abstraction was used, it’s possible to simply instantiate the TwoDArray class and call the printArray function to print the image to the console without closing the console, meeting all of the functional requirements.

**Conclusion**

When running the program, an image will print out the same image that is on the assignment. Tackling the entire problem in one chunk is rather difficult, so the key to this program is breaking each functional requirement down to the smallest pieces possible. Each small piece is still an interesting problem in themselves, but it’s no where near as big an obstacle as the problem as a whole.

There were a lot of struggles with this assignment: mainly syntax. I didn’t know that it was possible to refer to the componants of a 2D array as arr[r][c], which may have eliminated a lot of the struggle. Practicing breaking a large problem into smaller problems is something that I really need to do. It will be easier to identify how to solve each functional requirement with practice, which is an issue for me right now. Having tools does no good if I can’t use each one effectively. Going over this in class really helped, and seeing the thought process that was used to break it down during each line of code was an excellent resource for me, so that’s something that I’m very grateful for.