

Analysis

From reading the code, the code functions to display the elements of in the matrix and calculate the entire sum of those elements. The code operates on a repetition structure, specifically nested for-loops and for-each, and initially applies a selection structure to determine if there is a row in the matrix, in other words, to check if it is not empty. It also uses the primitive data type integer and an array, specifically 2D array or a matrix which is declared initially with row length 3 and column length 2.

The way it works is pretty simple: it just used a loop to traverse each element of matrix, starting from the first row of each matrix and traversing each element in that row, then the second row and vice versa. The first loop does this to assign values in a default initialized matrix. For each row, iteratively assign the value $n * i$, starting with $n = 1$ and $i = 1$, where each column step traversal increases i by 1, resetting i to 1 at the beginning of each new row, and where each row step traversal increases n by 1. As you can see, this is just a simple assignment of each element in a matrix with a particular. After assignment, each element is displayed through the use of a nested for-each and sums each element simultaneously, it uses the same procedure as the first nested for loops but using a different loop type. The following output is observed when we execute it:

Output:

```
Length of rows: 3
Length of columns: 2
1 2
2 4
3 6
Summation: 18
```

As observed, it actually does this describes it to do: it initializes, assigns, displays, and sums the elements of the matrix correctly.

In conclusion, this is a simple code that utilizes a straightforward logical structure to traverse a 2D matrix, requiring minimal analysis to understand and comprehend.