Pattern Recognition

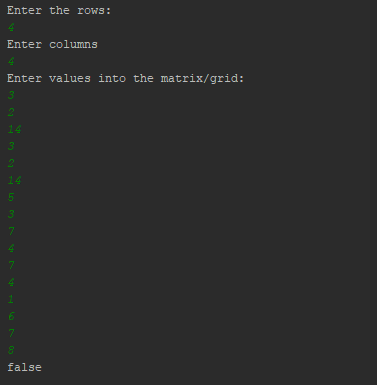
Pseudo-code:

1. Prompt user to enter rows and columns for matrix they want
2. Enter values for the matrix using scanner
   1. Nested For-loop for entering values into matrix
3. Create isConsecutiveFour method
   1. Check all rows, columns, and diagonals for 4 in a row.
      1. Return values from rows, column, and diagonal methods.
4. Create checkRows method
   1. Go through each row of the matrix starting at [0][0] and continue to only increase the rows [i][0]
   2. Create variable total that checks for four values in a row
      1. Total only increase if there are two numbers next to eachother
5. Create checkColumns()
   1. Go through each vertical column in matrix start at [0][0] and only increase [0][i]
   2. Create variable total that checks for four values in a row.
      1. Increase total by 1 for every similar value next to one another.
6. Create checkDiagonals()
   1. Check Diagonal numbers starting from every corner
      1. Top-left to bottom right
      2. Bottom left to top right

Screenshot:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cases | Input | Expected Output | Actual Output | Did Test Pass? |
| Case 1 | "Enter the rows:  4  Enter columns  4  Enter values into the matrix/grid: | True | True | Y |
| Case 2 | Enter the rows:  4  Enter columns  4  Enter values into the matrix/grid:  3 2 14 3  2 14 5 3  7 4 7 4  1 6 7 8 | false | false | Y |
| Case 4 | Enter the rows:  4  Enter columns  4  Enter values into the matrix/grid:  3 2 5 3  2 1 5 3  7 4 5 4  1 6 5 3 | true | True | Y |
| Case 3 | Enter the rows:  4  Enter columns  4  Enter values into the matrix/grid:  3 2 14 3  2 3 5 3  7 4 3 4  1 6 7 3 | true | true | Y |

Screenshots:



UML:

|  |
| --- |
| **Class Name:**  **PatternRecognition** |
| **+i: int**  **+rows: int**  **+columns: int**  **+grid[]: boolean** |
| **+ isConsecutiveFour(): boolean**  **+checkDiagonal(): Boolean**  **+checkRows(): Boolean**  **+checkColumns(): boolean** |

