Java → Basic syntax and simple programs → Arrays → Multi-dimensional array

Multi-dimensional array → Fill the matrix by numbers

2554 users solved this problem. Latest completion was 1 day ago.

Medium 16 minutes

Given the number n, not greater than 100, create the matrix of size n×n and fill it using the following rule. Numbers 0 should be stored on the primary (main) diagonal. The two diagonals, adjacent to the primary one, should contain numbers 1. The next two diagonals should contain numbers 2; etc.

Note: the primary diagonal runs from the top left corner to the bottom right corner.

Report a typo

Sample Input 1:

```
5
```

Sample Output 1:

```
0 1 2 3 4
1 0 1 2 3
2 1 0 1 2
3 2 1 0 1
4 3 2 1 0
```

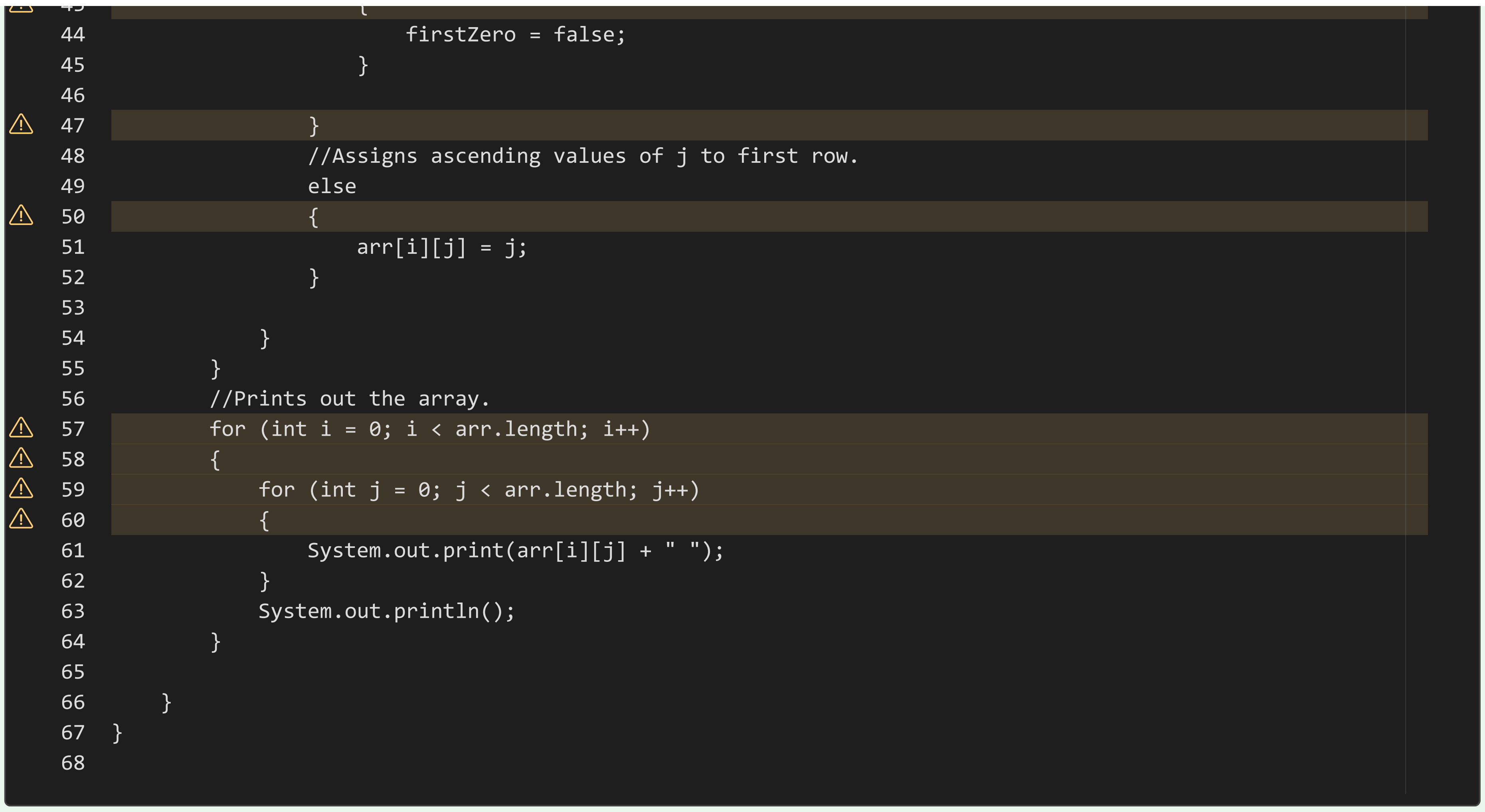
Write a program

Code Editor IDE

Java

```
import java.util.*;
class Main {
    // Function print matrix in spiral form
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        int size = scanner.nextInt();
        int[][] arr = new int[size][size];
        int start = 0;
        boolean firstZero = false;
        for (int i = 0; i < arr.length; i++)
        {
            for (int j = 0; j < arr.length; j++)
            {
                //Increase element by one for first number of each row.
                if (j == 0)
                {
                    arr[i][j] = start;
                    start++;

                }
                //Assign elements for remaining rows based on specific rules.
                else if (j >= 1 && i >= 1)
                {
                    //If previous element isn't 0 and firstZero boolean is false, decrease current element
                    if (arr[i][j - 1] != 0 && !firstZero)
                    {
                        arr[i][j] = arr[i][j - 1] - 1;
                    }
                    //If previous element is zero, increase current element by one and set firstZero boolea
                    if (arr[i][j - 1] == 0)
                    {
                        arr[i][j] = arr[i][j - 1] + 1;
                        firstZero = true;
                    }
                    //If an element in the row was zero, increase the remaining elements by one.
                    else if (firstZero)
                    {
                        arr[i][j] = arr[i][j - 1] + 1;
                    }
                    //Set the firstZero boolean to false once the row is complete.
                    if (j == arr.length - 1)
                    {
```



```
                    firstZero = false;
                }

            }
            //Assigns ascending values of j to first row.
            else
            {
                arr[i][j] = j;
            }

        }
    }
    //Prints out the array.
    for (int i = 0; i < arr.length; i++)
    {
        for (int j = 0; j < arr.length; j++)
        {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }

}
}

```

✓ Correct, but can be improved.

275 users liked this problem. **13** didn't like it. What about you?

Run Continue Solve again Solutions (305)

Time limit: 1 seconds Memory limit: 256 MB

Comments (32) Hints (9) Useful links (0) Solutions (305) Show discussion