Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 4
Part-5
REPORT

HAMZA YOĞURTCUOĞLU
171044086

Course Assistant:AYŞE ŞERBETÇİ TURAN

1 INTRODUCTION

1.1 Problem Definition

- There is a 2D arrays of non negative integers.

- Coding an iterator class for these data that will traverse a given 2D array spirally clockwise starting at the top left element.

- Recursively iterator is changing recursively 2D to 1D array.

1.2 System Requirements

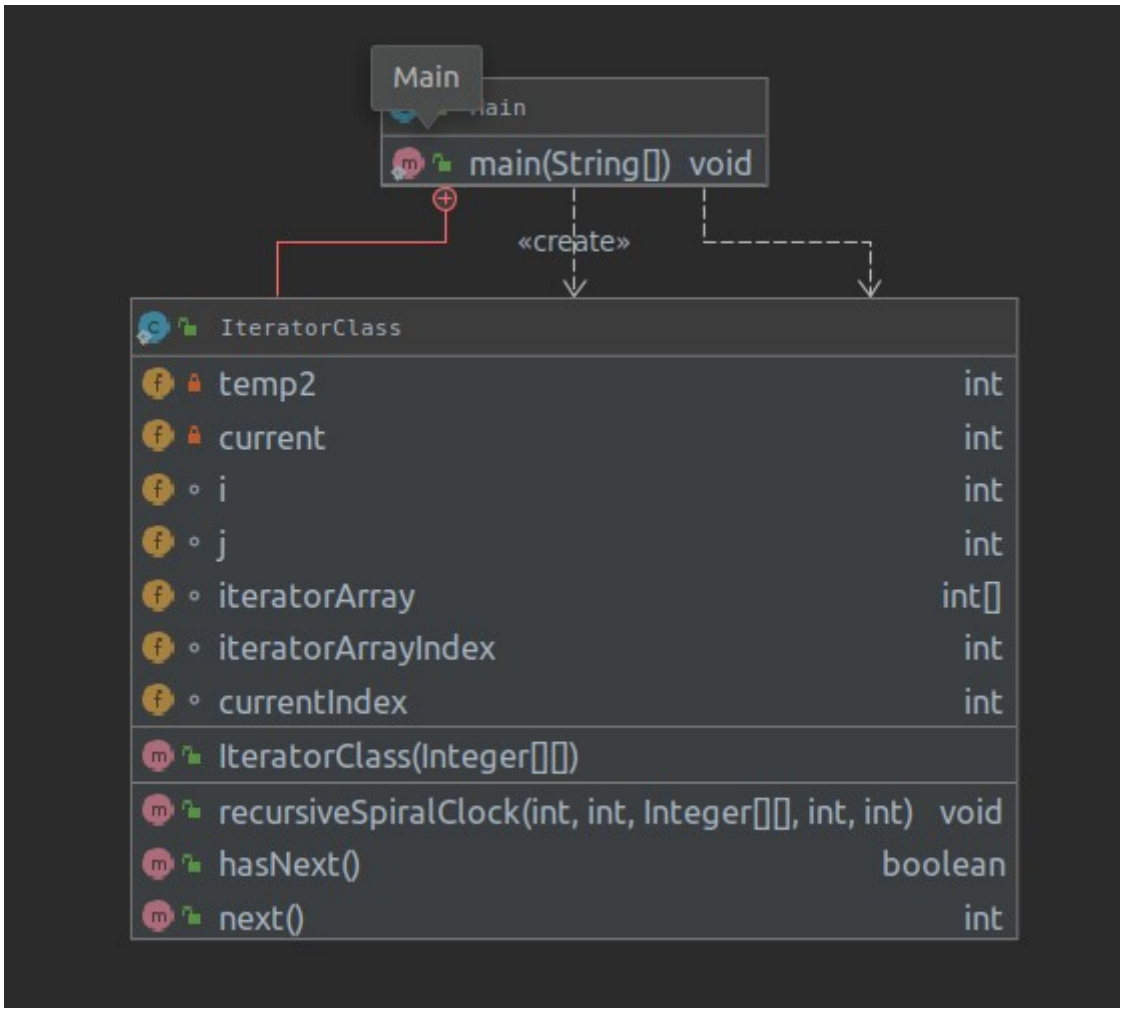
+ Total of Algoirthm Complexity is work with O(n).
+ All Operating Systems handle this program.
+ Doesnt need a lot of memory acually if have alot of data , memory usage will raise linearly.
+ Pogram can work 128KB of memory (That can be change your data.)
+ This program doest make properly in smartphone it can be work in computer.
+ Doesnt need a specific of hardware just a computer work.
+ Just take executable file for directly execute program. Then , it works efficently.

2 METHOD

2.1 Class Diagrams

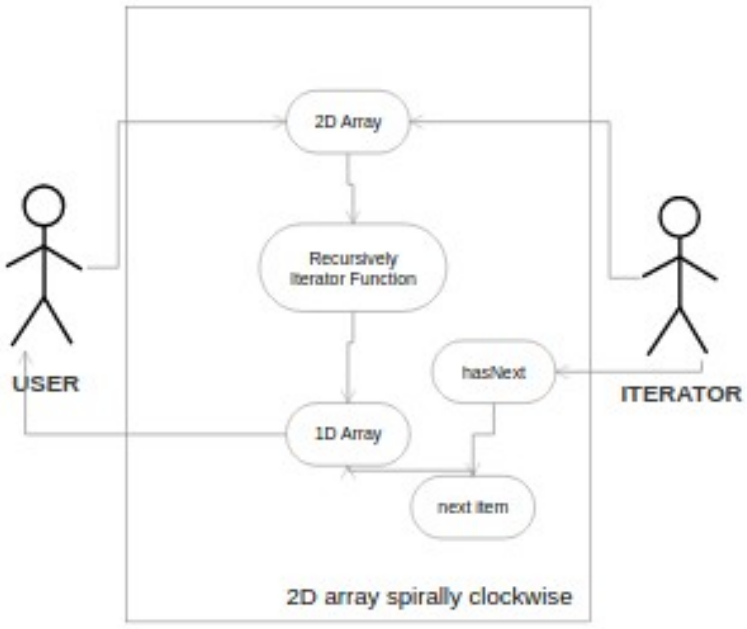
1)- IteratorClass is inner class of main class

2)- Main class can creat many IteratorClass object.



2.2 Use Case Diagrams

1) User creates a 2D array.
2) Iterator convert 2D array recursively to 1D array.
3) if user use next fucntion of iterator , orderly clockwise starting at the a element.



2.3 Problem Solution Approach

1-) Coded my iterator Class.

2-) This iterator class constructor is taken a 2D array.

3-) Converted recursively 2D array to 1D array.

4-) Called recursive function for each row and column.

5-) Each function callee is keeping on bound of row or column.

6-) If user used next of iterator . It traverses next integer.

RecursiveSpiralClock : T(n) = n + T(n/4)

Master Teorem : a = 1 , b = 4 , d = 1 => T(n) = Θ(n) (Okey)
next : O(2)

hasNext : O(1)

3 RESULT

3.1 Test Cases

```
public static void main(String[] args) {


   //------------------1 - 16 Clock --------------
    Integer[][] arrInt = {{1, 2, 3, 4},
            {5, 6, 7, 8},
            {9, 10, 11, 12},
            {13, 14, 15, 16}};

    IteratorClass<Integer> iter = new IteratorClass<>(arrInt);
    while (iter.hasNext())
        System.out.print (iter.next() + " ");

    System.out.println("\n");
    // ------------------  1 - 100 Clock-------------
    Integer[][] arrInt2 =
            {{1, 2,   3,  4, 5, 6, 7, 8, 9,10},
            {11, 12, 13, 14,15,16,17,18,19,20},
            {21, 22, 23, 24,25,26,27,28,29,30},
            {31, 32, 33, 34,35,36,37,38,39,40},
            {41, 42, 43, 44,45,46,47,48,49,50},
            {51, 52, 53, 54,55,56,57,58,59,60},
            {61, 62, 63, 64,65,66,67,68,69,70},
            {71, 72, 73, 74,75,76,77,78,79,80},
            {81, 82, 83, 84,85,86,87,88,89,90},
            {91, 92, 93, 94,95,96,97,98,99,100}

    };
    IteratorClass<Integer> iter2 = new IteratorClass<>(arrInt2);
    while (iter2.hasNext())
        System.out.print (iter2.next() + " ");

    System.out.println("\n");

}
```

First Test : 1 – 16 2D array iterator take 2D array

Second Test : 1 – 100 2D array iterator take 2D array

3.2 Running Results

Test 1 Result :

```
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
```

Test 2 Result : All result side by side

```
1 2 3 4 5 6 7 8 9 10 20 30 40 50 60 70 80 90 100 99 98 97 96 95 94 93 92 91 81 71 61 51
41 31 21 11 12 13 14 15 16 17 18 19 29 39 49 59 69 79 89 88 87 86 85 84 83 82 72 62 52 42 32 22 23 24 25
26 27 28 38 48 58 68 78 77 76 75 74 73 63 53 43 33 34 35 36 37 47 57 67 66 65 64 54 44 45 46 56 55
```