06 December 2024 20:06

2D Array

```
[ [1,2,3,4],
  [10,20,30,40],
  [15,25,35,45],
  [80,90,100,105] ]
```

Matrix1 →

Column

Row

1	2	3	4
10	20	30	40
15	25	35	45
80	90	100	105

Matrix2 →

5	15
20	40
6	80
16	34

rows = 4
columns = 2

Declaration

```
int arr[ ][ ]
```

Initialization

```
int arr[ ][ ] = new int [row] [column]
int arr[ ][ ] = new int [4][2];
```

Index

← columns

Index

← columns

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 17_{00} & 27_{01} & 35_{02} \\ 1 & 4_{10} & 51_{11} & 65_{12} \\ 2 & 20_{20} & 40_{21} & 90_{22} \\ 3 & 6_{30} & 60_{31} & 100_{32} \end{array}$$

rows →

rows = 4 = n
Columns = 3 = m

Initialize with values :-

```
int arr[][] = { {17, 27, 35},
                {4, 51, 65},
                {20, 40, 90},
                {6, 60, 100} }
```

arr[0][0], arr[0][1], arr[0][2]
arr[1][0], arr[1][1], arr[1][2]
arr[2][0], arr[2][1], arr[2][2]
arr[3][0], arr[3][1], arr[3][2]

Taking input from user

n → rows

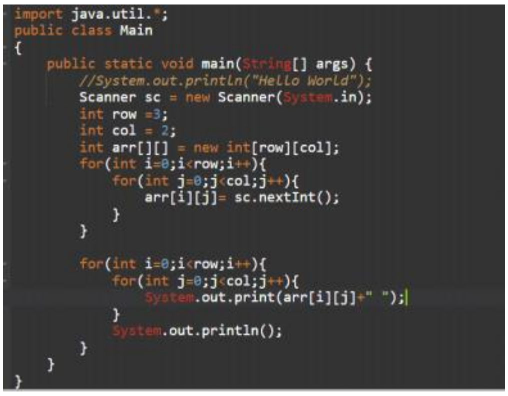
```
for(int i=0; i<row; i++) {
    for(int j=0; j<column; j++) {
    arr[i][j] = sc.nextInt();
    } }
```

Printing matrix :-

```
for(int i=0; i<row; i++) {
  for(int j=0; j<column; j++) {
      S.o.p(arr[i][j]+" ");
    }
   S.o.pln();
 }
```

Output

```
17 27 35
4  51 65
20 40 90
6  60 100
```



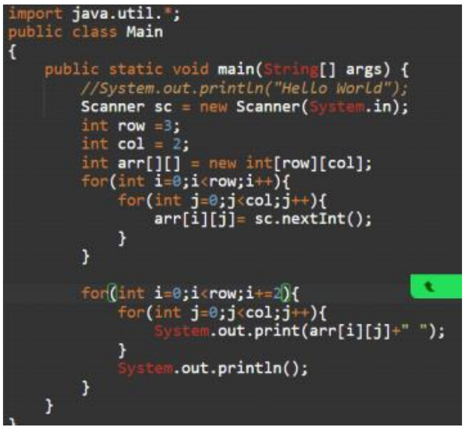
```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        //System.out.println("Hello World");
        Scanner sc = new Scanner(System.in);
        int row =3;
        int col = 2;
        int arr[][] = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                arr[i][j]= sc.nextInt();
            }
        }

        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                System.out.print(arr[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

Time complexity – O(nm)

06 December 2024 20:47

Print Alternate row in Matrix



```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        //System.out.println("Hello World");
        Scanner sc = new Scanner(System.in);
        int row =3;
        int col = 2;
        int arr[][] = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                arr[i][j]= sc.nextInt();
            }
        }

        for(int i=0;i<row;i+=2){
            for(int j=0;j<col;j++){
                System.out.print(arr[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

To print alternate row i will be incremented by 2 to skip next row

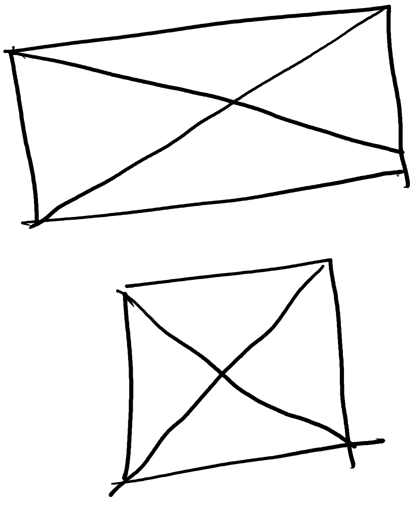
To print alternate column j will be incremented by 2 to skip next column

Time Complexity → O(nm)

Print Matrix Left Diagonally

Input

10	15	16
62	20	36
86	40	30



Output

10 20 30

Solution :-

	0	1	2
0	10 00	15 01	16 02
1	62 10	20 11	36 12
2	86 20	40 21	30 22

00, 11, 22

arr[0][0], arr[1][1], arr[2][2]

```
for ( i=0; i<row; i++) {
```

```
for(i=0; i<row; i++){
   s.o.p(arr[i][i]+" ");
   }
```

Time Complexity -> O(n)

Print Upper Triangular Matrix

Example 1

	0	1	2
0	100 00	6 01	41 02
1	90 10	80 11	24 12
2	56 20	37 21	25 22
3	71 30	76 31	55 32

```
100  6  41
     80 24
        25
```

```
00  01  02
    11  12
        22
```

Example 2.

	0	1	2
0	16 00	15 01	14 02
1	90 10	80 11	70 12
2	24 20	56 21	78 22

```
16  15  14
    80  70
        78
```

```
00  01  02
    11  12
        22
```

i=0, j=0 to 2

i=1, j=1 to 2

i=2, j=2 to 2

Output. scenario 1.

16 15 14 80 70 78

i=0, j=0 to 2

i=1, j=1 to 2

i=2, j=2 to 2

```
for(int i=0; i<n; i++) {
    for(int j=i; j<m; j++) {
        S.o.p(arr[i][j]+" ");
    }
}
```

Scenario 2 .

```
Output  00         01        02
        16 __ 15 __ 14
        10         11        12
        ____ 80 __ 70
                             22
                             78
        __________
```

```
for(int i=0; i<n; i++){
for(int j=0; j<m; j++){
    if(j>=i){
    s.o.p(arr[i][j]+" ");
    }else{
    s.o.p("    "); //Double space
      }
    }
    s.o.pln();
  }
```