



C# programming language. The beginning

Course

Programming Languages

Semester 2, FIIT

Mayer S.F.

Mikhalkovich S.S

LECTURE # 10. Matrixes

Multi-dimensional

```
int[,] a = new int[9, 9];  
for (var i = 0; i < a.GetLength(0); i++)  
    for (var j = 0; j < a.GetLength(1); j++)  
        a[i, j] = (i + 1) * (j + 1);
```

Number of rows

Number of cols

elements
generation using
rules

```
for (var i = 0; i < a.GetLength(0); i++)  
{  
    for (var j = 0; j < a.GetLength(1); j++)  
        Console.Write($"{a[i, j],3} ");  
    Console.WriteLine();  
}
```

elements output

```
int[,] b = {  
    { 1, 2, 3, 4 },  
    { 2, 3, 4, 5 },  
    { 3, 4, 5, 6 }  
};
```

If we know the
values

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Jugged arrays. Stepped array

```
int[][] a = new int[3][];
```

```
for (var i = 0; i < 3; i++)  
{  
    a[i] = new int[i+2];  
}
```

Each i -indexed row, has $i+2$ column.
Consequently, we have steps

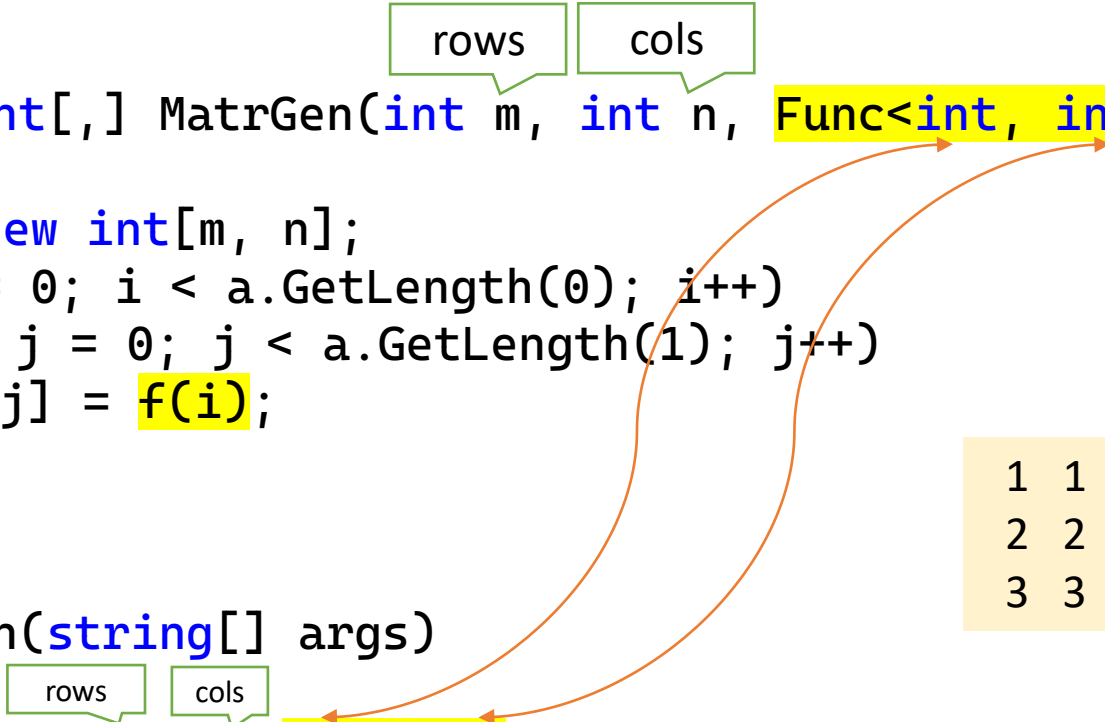
```
for (var i = 0; i < a.Length; i++)  
{  
    for (var j = 0; j < i+2; j++) {  
        a[i][j] = i;  
        Console.Write($"{a[i][j],3} ");  
    }  
    Console.WriteLine();  
}
```

```
0 0  
1 1 1  
2 2 2 2
```

Arrays generating with lambda-functions (1)

```
public static int[,] MatrGen(int m, int n, Func<int, int> f)
{
    int[,] a = new int[m, n];
    for (var i = 0; i < a.GetLength(0); i++)
        for (var j = 0; j < a.GetLength(1); j++)
            a[i, j] = f(i);
    return a;
}

static void Main(string[] args)
{
    var a = MatrGen(3, 4, i => i+1);
}
```



1	1	1	1
2	2	2	2
3	3	3	3

Arrays generating with lambda-functions (2)

```
public static int[,] MatrGen(int m, int n, Func<int, int, int> f)
{
    int[,] a = new int[m, n];
    for (var i = 0; i < a.GetLength(0); i++)
        for (var j = 0; j < a.GetLength(1); j++)
            a[i, j] = f(i, j);
    return a;
}

static void Main(string[] args)
{
    var a = MatrGen(3, 4, (i, j) => i*j);
}
```

rows cols

0	0	0	0
0	1	2	3
0	2	4	6

Extension method `.Print`

```
class Program
{
    static void Main(string[] args)
    {
        var a = MatrGen(3, 4, (i,j) => i*j);
        a.Print();
    }
}
static class MyClass
{
    public static void Print(this int[,] a) // extension
    {
        for (var i = 0; i < a.GetLength(0); i++)
        {
            for (var j = 0; j < a.GetLength(1); j++)
                Console.Write($"{a[i, j],3} ");
            Console.WriteLine();
        }
    }
}
```

Algorithms with rows, extension methods

```
public static int SumRow(this int[,] a, int k)
{
    var sum = 0;
    for (var j = 0; j < a.GetLength(1); j++)
        sum += a[k, j];
    return sum;
}
```

This – for extension
a is an array
k is a row number

```
public static int MinRow(this int[,] a, int k)
{
    var min = a[k, 0];
    for (var j = 0; j < a.GetLength(1); j++)
        if (a[k, j] < min)
            min = a[k, j];
    return min;
}
```

```
static void Main(string[] args)
{
    var a = new int[2, 3] { {1, 2, 3}, {4, 5, 6} };
    var sum = a.SumRow(1);
    Console.WriteLine(sum); // 15
}
```

How to call the method

Algorithms with cols, extension methods

```
public static int MaxCol(this int[,] a, int k)
{
    var max = a[0, k];
    for (var i = 0; i < a.GetLength(0); i++)
        if (a[i, k] > max)
            max = a[i, k];
    return max;
}
```

This – for extension
a is an array
k is a col number

```
static void Main(string[] args)
{
    var a = new int[2, 3] { {1, 2, 3 }, {4, 5, 6} };
    var max = a.MaxCol(1);
    Console.WriteLine(max); // 5
}
```

How to call the method

Lecture tasks

Lesson #10

<https://labs-org.ru/c-sharp10-eng/>

Q & A