

Lecture **#8**

Lesson **# 15**

Basics of Programming.

Matrices

Course Basics of Programming Semester 1, FIIT

Mayer Svetlana Fyodorovna

Schedule

- **Consultation** before the exam (Q & A): **9.01.2021** at **10 a.m.** (sat)
- **Exam**: **11.01.2021** at **9 a.m.** (mon)



Matrices

Definition

- A **matrix** or two-dimensional array is a rectangular table of numbers (or other elements of the same type). Each element of the matrix has two indices (**row** number and **column** number).

A

	0	1	2	3	4
0	1	4	7	3	6
1	2	-5	0	15	10
2	8	9	11	12	20

Diagram illustrating a 3x5 matrix **A** with row and column indices. The matrix is shown with rows 0, 1, 2 and columns 0, 1, 2, 3, 4. The element at row 1, column 3 (value 15) is highlighted in green. Callouts indicate "column 2" pointing to column index 2, "row 1" pointing to row index 1, and "cell **A**[2, 3]" pointing to the cell at row 2, column 3 (value 12).

```
var a: array [,] of integer;  
a := new integer[3, 5];
```

Two-dimensional Arrays

i,j

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3

Declaration

```
begin
  var a2: array [,] of integer;
  a2 := new integer[3,4];
  var a := new integer[3,4];

  for var i:=0 to a2.RowCount-1 do
    for var j:=0 to a2.ColCount-1 do
      a2[i,j] := i + j;

  a2.Println;
  Println(a2);
end.
```

Loop over the rows

Loop over the columns

Output

2 positions for
output element

```
0  1  2  3
1  2  3  4
2  3  4  5
[[0,1,2,3],[1,2,3,4],[2,3,4,5]]
```

```
a.Println(2);
```

Example

To do: Initialize a two-dimensional array named **a** with given integer values, they are **[[1, 2], [3, 4], [5, 6]]**

1) First, output all the elements of the array. **2)** After, output the following elements to the Output window (the elements to output are marked in red color):

[1, 2]

[3, 4]

[5, 6]

```
begin  
  var a := new integer[3,2];  
  
  for var i:=0 to a.RowCount-1 do  
    for var j:=0 to a.ColCount-1 do  
      a[i,j]:=readinteger;  
    println('The array:');  
    a.Println();  
    Println(a[1,0], a[2,0]);  
end.
```

i,j

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3

Matrix filling

i,j

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3

```
begin
  var a := Matr(3,4,1,2,3,4,5,6,7,8,9,10,11,12);
  a.Println(3);

  var a1 := MatrGen(3,4,(i,j)->i+j+0.5);
  a1.Println(5,1);
end.
```

5 positions for element, 1 position for fractional part

Matrix dimensions

```
1  2  3  4
5  6  7  8
9 10 11 12
0.5 1.5 2.5 3.5
1.5 2.5 3.5 4.5
2.5 3.5 4.5 5.5
```

Filling with random numbers

```
begin  
  var a := MatrRandomInteger(3, 4);  
  a.Println(4);  
  
  var a1 := MatrRandomReal(3, 4, 1, 9);  
  a1.Println(6, 2);  
end.
```

```
22  32  10  41  
11  25  50  50  
81  19  25  73  
7.58  1.99  4.99  2.09  
7.39  2.82  3.04  7.39  
5.86  8.64  1.33  5.63
```


Processing all matrix elements

Traditional code

Problem. Calculate sum of the elements.

```
begin
  var a := MatrRandomInteger(3, 4);
  a.Println;
  var sum:=0;
  for var i := 0 to a.RowCount - 1 do
    begin
      for var j := 0 to a.ColCount - 1 do
        sum += a[i, j];
      end;
    println(sum);
  end.
```

```
42  54  68   9
81  36  79  34
75  27  64  75
644
```

Tasks

- Task 1,2,3,4

Rows and columns of matrices

```
begin
  var a := MatrRandomInteger(3,4);
  a.Println;
  Println;

  var k := 1;
  for var j:=0 to a.ColCount-1 do
    Print(a[k,j]);
    Println;

    a.Row(k).Println;

  k := 2;
  for var i:=0 to a.RowCount-1 do
    Print(a[i,k]);
    Println;

    a.Col(k).Println;
  end.
```

k-th row of matrix (one-dimensional array)

k-th column of matrix (one-dimensional array)

```
31 28 56 60
45 65 70 19
29 32 72 88
```

```
45 65 70 19
45 65 70 19
56 70 72
56 70 72
```

Operations by rows and columns

```
begin  
  var a := MatrRandomInteger(3,4);  
  a.Println;  
  
  a.Row(0).Sum.Println;  
  a.Row(1).Average.Println;  
  a.Row(2).Product.Println;  
  
  a.Col(0).Min.Println;  
  a.Col(1).Max.Println;  
end.
```

```
77 51 30 61  
29 10 34 19  
83 40 79 57  
219  
23  
14949960  
29  
51
```

Processing all matrix elements

Traditional code

Problem. Calculate sum in every row.

```
begin
  var a := MatrRandomInteger(3,4);
  a.Println;

  var Sums := new integer[a.RowCount];
  for var i:=0 to a.RowCount-1 do
    begin
      var sum := 0;
      for var j:=0 to a.ColCount-1 do
        sum += a[i,j];
      Sums[i] := sum
    end;

    Sums.Println;
  end.
```

← one-dimensional array to store the sums of rows

```
45  13  80   5
18  62  48  41
 8  77  63  26
143 169 174
```

Mass operations by rows and columns

Problem. Calculate sum in each row.

```
begin
  var a := MatrRandomInteger(3,4);
  a.Println;

  var Sums := ArrGen(a.RowCount, r -> a.Row(r).Sum);
  Sums.Println;

  for var i:=0 to a.RowCount-1 do
    a.Row(i).Sum.Print;
end.
```

Two ways to calculate the sum



```
15  21   9  78
 54  61   2  19
 19  28  50  61
123 136 158
123 136 158
```

Mass operations by rows and columns (2)

Problem. Calculate minimal element in every column.

```
begin  
  var a := MatrRandomInteger(3, 4);  
  a.Println;  
  
  var Mins := ArrGen(a.ColCount, c -> a.Col(c).Min);  
  Mins.Println;  
end.
```

```
98  65  84  45  
46   7  18  50  
82  17   6  26  
46  7  6  26
```

Mass operations on rows and columns (3)

Problem. Calculate a number of evens in each row

```
begin
  var a := MatrRandomInteger(3, 4);
  a.Println;

  var EvensCount := ArrGen(a.ColCount, c -> a.Col(c).Count(x -> x.IsEven));
  EvensCount.Println;
end.
```

```
49  98  78  27
44  65  50  74
68   1  75  59
2  1  2  1
```


Mass operations on rows and columns (4)

Problem. Calculate the minimum among maximum elements of rows.

```
begin
  var a := MatrRandomInteger(3, 4);
  a.Println;
  ArrGen(a.RowCount, r -> a.Row(r).Max).Min.Println;
end.
```

```
27  1  92  0
48  68 10  25
93  58 65  51
68
```

Tasks

- Task 5,6,7

Search in matrix

Problem. Does the matrix contain the element x?

Solution (bad).

```
begin
  var a := MatrRandomInteger(3,4,1,10);
  a.Println;

  var found := False;
  for var i:=0 to a.RowCount-1 do
    for var j:=0 to a.ColCount-1 do
      if a[i,j] =5 then
        found := True;

  Println(found);
end.
```

```
2  1  3 10
1  6  7  7
1  9  2  3
False
```

Search in matrix, with function

Problem. Does the matrix contain the element x?

Solution (good).

```
function Contains<T>(a: array [,] of T; x: T): boolean;  
begin  
    Result := False;  
    for var i:=0 to a.RowCount-1 do  
        for var j:=0 to a.ColCount-1 do  
            if a[i,j]=x then  
                begin  
                    Result := True;  
                    exit;  
                end;  
            end;  
        end;  
    end;  
  
begin  
    var a := MatrRandomInteger(3,4,1,10);  
    a.Println;  
  
    var found := Contains(a,5);  
    Println(found);  
end.
```

5 is to find

2	1	3	10
1	6	7	7
1	9	2	3

False

Tasks

- Task 8,9

Q & A