

Основы алгоритмизации и программирования.

Матрицы

Педагогическое образование, 3 семестр

Mayer Svetlana Fyodorovna

Lecture **#8**

Lesson **# 15**



Матрицы

Объявление

- Матрица или двумерный массив - это прямоугольная **таблица** чисел (или другие элементы того же типа). Каждый элемент матрицы имеет два индекса (номер строки и номер столбца).

A

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

колонка 2

строка 1

ячейка **A**[2, 3]

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

Двумерный массив

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

Объявление

```
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.
```

Проход в цикле по строкам

Проход в цикле по колонкам

Вывод

2 ПОЗИЦИИ НА ВЫВОД
элемента

```
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);

Пример

Задача: Инициализируйте двумерный массив с именем **a** с заданными целочисленными значениями: **[[1, 2], [3, 4], [5, 6]]**

1) Сначала выведите все элементы массива. **2)** После этого выведите в окно вывода следующие элементы (элементы для вывода отмечены красным цветом):

[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

Заполнение матрицы

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 позиций на вывод элемента, 1 позиция на десятичную часть

Размерность матрицы

```
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
```

Заполнение случайно сгенерированными числами

```
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
```

Обработка всех элементов матрицы

Традиционный код

Задача. Вычислить сумму элементов.

```
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
```


Строки и столбцы матриц

```
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-я строка матрицы (одномерный массив)

k-й столбец матрицы (одномерный массив)

```
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
```

Операции по строкам и столбцам

```
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
```

Обработка всех элементов матрицы

Традиционный код

Задача. Вычислить сумму в каждой строке.

```
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.
```

одномерный массив для хранения сумм строк

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

Массовые операции по строкам и столбцам

Задача. Вычислите сумму в каждой строке.

```
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.
```

Два способа вычисления суммы



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

Массовые операции по строкам и столбцам(2)

Задача. Вычислите минимальный элемент в каждом столбце.

```
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
```

Массовые операции по строкам и столбцам(3)

Задача. Вычислите количество четных в каждой строке

```
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
```

Массовые операции по строкам и столбцам (4)

Задача. Вычислите минимум среди максимальных элементов строк.

```
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
```

Поиск в матрице

Задача. Содержит ли матрица элемент **x**?

Решение (неэффективное).

```
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
```


Поиск в матрице с функцией

Задача. Содержит ли матрица элемент **x**?

Решение (эффективное).

```
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.
```

Ищем 5

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

Q & A