

# Матрицы

---

10 класс

- Многие программы работают с данными, организованными в виде таблиц.
- Например, при составлении программы для игры в крестики-нолики нужно запоминать состояние каждой клетки квадратной доски.
- Можно поступить так: пустым клеткам присвоить код -1, клетке, где стоит нолик, — код 0, а клетке с крестиком — код 1.
- Тогда информация о состоянии поля может быть записана в виде таблицы

	○	×
	○	×
○	×	

нет знака

	1	2	3
1	-1	0	1
2	-1	0	1
3	0	1	-1

НОЛИК

крестик

строка 2,  
столбец 3

	○	×
	○	×
○	×	

нет знака

НОЛИК

	1	2	3
1	-1	0	1
2	-1	0	1
3	0	1	-1

крестик

строка 2,  
столбец 3

- Такие таблицы называются **матрицами** или **двумерными массивами**. Каждый элемент матрицы, в отличие от обычного (линейного) массива, имеет два индекса — номер строки и номер столбца. В таблице выше зеленым фоном выделен элемент, находящийся на пересечении второй строки и третьего столбца.

**Матрица — это прямоугольная таблица, составленная из элементов одного типа (чисел, строк и т.д.).**

**Каждый элемент матрицы имеет два индекса — номера строки и столбца.**

При объявлении матриц указывают два диапазона индексов (для строк и столбцов):

---

```
const N = 3; M = 4;  
var A: array[1..N, 1..M] of integer;  
    X: array[-3..0, -8..M] of double;  
    L: array[1..N, 0..1] of boolean;
```

строки

столбцы

Идентификатор	Длина (байт)	Диапазон значений	Операции
<b>Целые типы</b>			
<code>integer</code>	2	-32768..32767	+, -, /, *, Div, Mod, >=, <=, =, <, >
<code>byte</code>	1	0..255	+, -, /, *, Div, Mod, >=, <=, =, <, >
<code>word</code>	2	0..65535	+, -, /, *, Div, Mod, >=, <=, =, <, >
<code>shortint</code>	1	-128..127	+, -, /, *, Div, Mod, >=, <=, =, <, >
<code>longint</code>	4	-2147483648..2147483647	+, -, /, *, Div, Mod, >=, <=, =, <, >
<b>Вещественные типы</b>			
<code>real</code>	6	$2,9 \times 10^{-39} - 1,7 \times 10^{38}$	+, -, /, *, >=, <=, =, <, >
<code>single</code>	4	$1,5 \times 10^{-45} - 3,4 \times 10^{38}$	+, -, /, *, >=, <=, =, <, >
<code>double</code>	8	$5 \times 10^{-324} - 1,7 \times 10^{308}$	+, -, /, *, >=, <=, =, <, >
<code>extended</code>	10	$3,4 \times 10^{-4932} - 1,1 \times 10^{4932}$	+, -, /, *, >=, <=, =, <, >
<b>Логический тип</b>			
<code>boolean</code>	1	true, false	Not, And, Or, Xor, >=, <=, =, <, >
<b>Символьный тип</b>			
<code>char</code>	1	все символы кода ASCII	+, >=, <=, =, <, >

- Каждому элементу матрицы можно присвоить любое значение, допустимое для выбранного типа данных.
- Поскольку индексов два, для заполнения матрицы нужно использовать вложенный цикл.
- Далее в примерах будем считать, что объявлена матрица из  $N$  строк и  $M$  столбцов, а  $i$  и  $j$  — целочисленные переменные, обозначающие индексы строки и столбца.
- В следующем примере матрица заполняется случайными числами и выводится на экран:

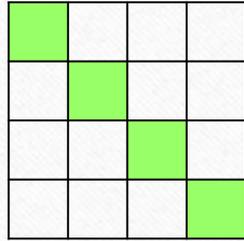
```
for i:=1 to N do begin
  for j:=1 to M do begin
    A[i,j]:=random(61)+20;
    write(A[i,j]:3)
  end;
  writeln
end;
```

Вложенный цикл!

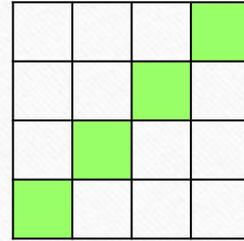
- Такой же двойной цикл нужно использовать для перебора всех элементов матрицы.
- Вот как вычисляется сумма значений всех элементов:

```
s := 0;  
for i:=1 to N do  
  for j:=1 to M do  
    s := s + A[i,j];
```

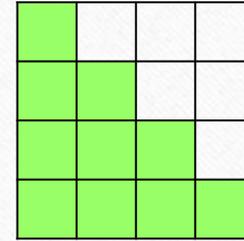
# Обработка элементов матрицы



а



б

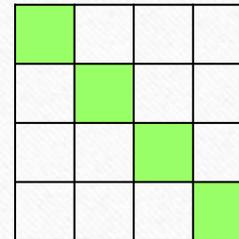


в

- Покажем, как можно обработать (например, сложить) некоторые элементы **квадратной матрицы**  $A$ , содержащей  $N$  строк и  $N$  столбцов.
- На рисунке выделены а – главная диагональ матрицы, б – вторая, побочная диагональ, в – главная диагональ и все элементы под ней.

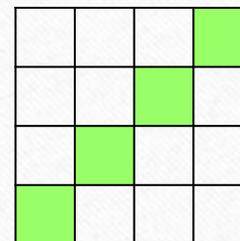
- Главная диагональ — это элементы  $A[1,1]$ ,  $A[2,2]$ , ...,  $A[N,N]$ , т. е. элементы, у которых номер строки равен номеру столбца.
  - Для перебора этих элементов нужен один цикл:
- 

```
for i:=1 to N do begin
  { работаем с A[i,i] }
end;
```



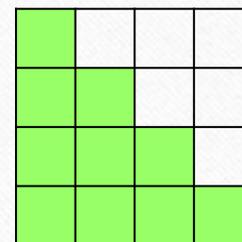
- Элементы побочной диагонали — это  $A[1,N]$ ,  $A[2,N-1]$ , ...,  $A[N,1]$ .
  - Заметим, что сумма номеров строки и столбца для каждого элемента равны  $N + 1$ , поэтому получаем такой цикл перебора:
- 

```
for i:=1 to N do begin
  { работаем с  $A[i,N+1-i]$  }
end;
```



- В случае обработки всех элементов на главной диагонали и под ней нужен вложенный цикл: номер строки будет меняться от 1 до  $N$ , а номер столбца для каждой строки  $i$  — от 1 до  $i$ :
- 

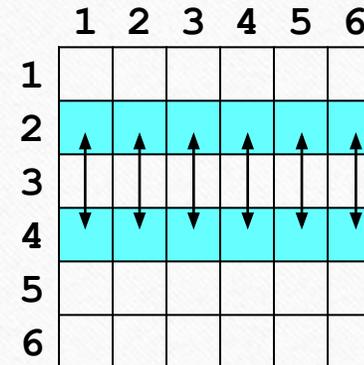
```
for i:=1 to N do
  for j:=1 to i do begin
    { работаем с A[i,j] }
  end;
```



Чтобы переставить строки или столбцы, достаточно одного цикла. Например, переставим строки 2 и 4, используя вспомогательную целую переменную  $c$ :

---

```
for j:=1 to M do
  c:=A[2,j];
  A[2,j]:=A[4,j];
  A[4,j]:=c
end;
```



# Задание 1.

---

## Пример:

Матрица A:

12 14 67 45

32 87 45 63

69 45 14 11

40 12 35 15

Максимальный элемент  $A[2,2]=87$

Минимальный элемент  $A[3,4]=11$

- Напишите программу, которая заполняет квадратную матрицу случайными числами в интервале  $[10,99]$ , и находит максимальный и минимальный элементы в матрице и их индексы.

## Задание 2.

Пример:

**Матрица А:**

12	14	67	45
32	87	45	63
69	45	14	30
40	12	35	65

**Результат:**

12	0	0	0
32	87	0	0
69	45	14	0
40	12	35	65

- Напишите программу, которая заполняет квадратную матрицу случайными числами в интервале  $[10,99]$ , а затем записывает нули во все элементы выше главной диагонали. Алгоритм не должен изменяться при изменении размеров матрицы.

# Вопросы и задания

---

1. Что такое матрицы? Зачем они нужны?
2. Сравните понятия «массив» и «матрица».
3. Как вы думаете, можно ли считать, что первый индекс элемента матрицы — это номер столбца, а второй — номер строки?
4. Могут ли индексы элементов матрицы принимать отрицательные и нулевые значения?
5. Что такое главная и побочная диагонали матрицы?
6. Почему суммирование элементов главной диагонали требует одиночного цикла, а суммирование элементов под главной диагональю — вложенного?

# Домашнее задание

---

- Читать конспект и презентацию
- Ответить на вопросы и выполнить задания