

Двумерные массивы

В Паскале двумерный массив трактуется как одномерный массив, каждый элемент которого также является массивом (массив массивов)

Двумерные массивы представляются в виде прямоугольной таблицы - **матрицы**.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

Матрица,
состоящая из 3
строк и 4
столбцов

Если количество строк равно количеству столбцов, то матрица называется **квадратной**.

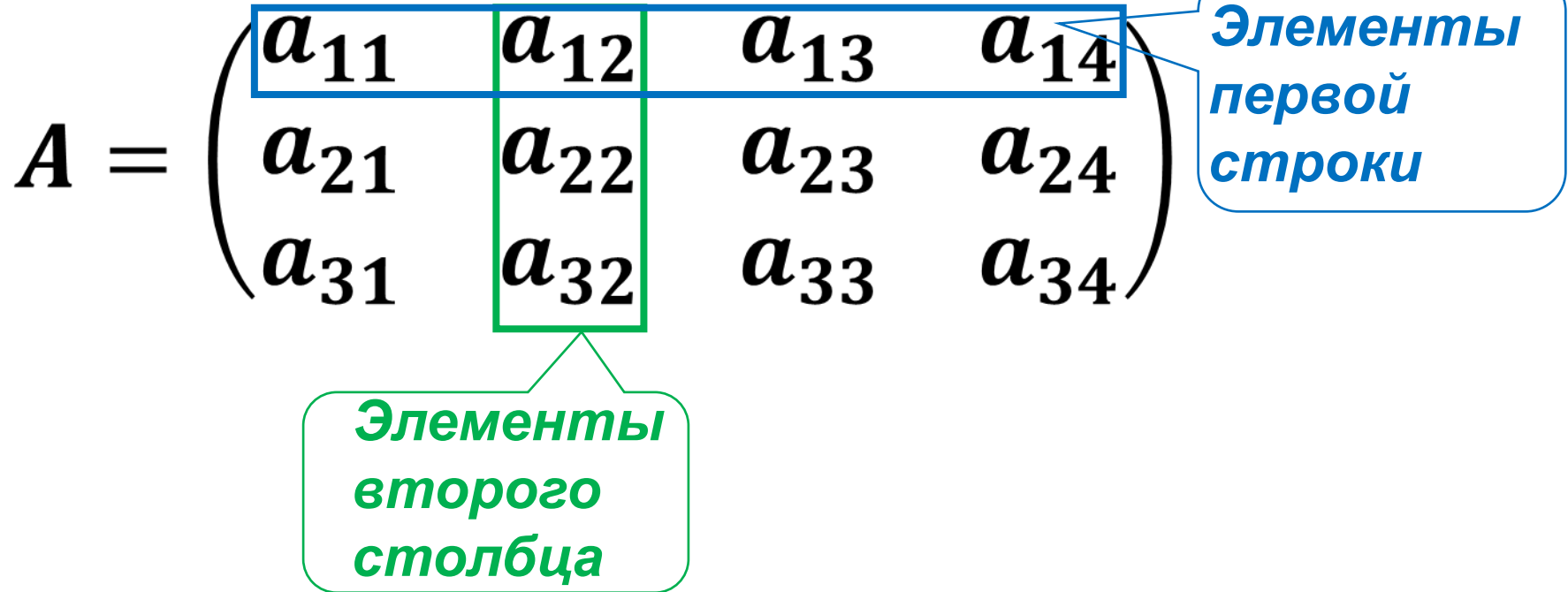
Квадратная матрица размером 3*3

Главная диагональ

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Побочная диагональ

Положение элемента в двумерном массиве определяется двумя индексами - **номером строки** и **номером столбца**, на пересечении которых он находится .



a_{12} – элемент расположен в первой строке второго столбца.

Описание двумерного массива

Описание двумерного массива в разделе **Var** аналогично описанию одномерного массива, только необходимо указывать диапазон изменения двух индексов - **столбцов и строк**.

```
Var b:array [1..15,1..20] of real;
```

Или

```
Const n=15; m=20;
```

```
Var a:array [1..n, 1..m] of integer;
```

Обращение к элементам

Обращение к элементам двумерного массива в Паскале имеет вид: **a [i , j]**,

где **a** - имя массива, описанного в блоке **VAR**,

i - номер строки,

j - номер столбца.

a [1 , 2] – обращение к элементу 1-й строки 2-го столбца массива **a**.

Ввод элементов двумерного массива с клавиатуры

```
writeln ('Введите элементы массива');  
for i:=1 to n do {Цикл по номерам строк - перебирает строки}  
for j:= 1 to m do {Цикл по номерам столбцов}  
begin  
write ('a[', i, ', ', j, ']=');  
  readln (a[i , j]);  
end;
```

Вывод элементов двумерного массива в виде таблицы

```
writeln ('Вывод массива по строкам');  
for i:=1 to n do {Перебор строк}  
begin  
    for j:= 1 to m do write (a[ i, j] :5); {Перебор  
        writeln; {Переход на новую строку} столбцов  
        элемент}  
end;
```

Заполнение двумерного массива случайными числами

```
randomize;  
for i:=1 to n do  
begin  
  for j:=1 to m do  
  begin  
    a[i , j]:=random(11)+2;  
    write(a[i , j], ' ');  
  end;  
  writeln;  
end;  
end;
```

Массив
заполняется
случайными
целыми числами из
интервала [2,12]

Суммирование элементов двумерного массива

```
Program сумма_1;  
Const n1=100; m1=100;  
Var a:array [1..n1,1..m1] of integer;  
    i, j, sum, n, m: integer;  
Begin  
randomize;  
write('Количество строк > '); readln(n);  
write('Количество столбцов > '); readln(m);  
for i:=1 to n do  
    for j:=1 to m do  
        a[i , j]:=random(11)+2;  
writeln ('Исходный массив');  
for i:=1 to n do  
begin  
    for j:=1 to m do Write (a[ i, j] :5);  
    writeln;  
end;  
sum:=0;  
for i:=1 to n do  
    for j:=1 to m do sum:=sum+a[i,j];  
writeln('Сумма=',sum);  
end.
```

Найти сумму элементов каждой строки массива

```
Program сумма_2;  
Const n1=100; m1=100;  
Var a:array [1..n1,1..m1] of integer;  
    i, j, sum, n, m: integer;  
Begin  
randomize;  
write('Введите количество строк'); readln(n);  
write('Введите количество столбцов'); readln(m);  
for i:=1 to n do  
    for j:=1 to m do a[i , j]:=random(11)+2;  
writeln ('Исходный массив');  
for i:=1 to n do  
begin  
    for j:=1 to m do Write (a[ i, j] :5);  
    writeln;  
end;  
for i:=1 to n do  
begin  
    sum:=0;  
    for j:=1 to m do sum:=sum+a[i,j];  
    writeln('Сумма=',sum);  
end; end.
```

Сумма элементов главной диагонали (x[1,1], x[2,2] и т.д.).

```
Program sum;  
Const n1=10; m1=12;  
Var   a:array [1..n1,1..m1] of integer;  
      i, j, sum : integer;  
Begin  
... {Вставить - ввод размерности, ввод элементов  
массива и вывод в виде таблицы}  
sum:=0;  
for i:=1 to n do sum:=sum+a[i,i];  
writeln ('Сумма=',sum);  
end.
```

На главной диагонали номер строки равен номеру столбца, т.е. элементами главной диагонали являются a_{11} , a_{22} , a_{33} и т.д.

Сумма по столбцам с записью в НОВЫЙ массив zum[1..m]

```
Program sum;
```

```
Const n=10; m=12;
```

```
Var    a:array [1..n,1..m] of integer;
```

```
    zum:array [1..m] of integer;
```

```
    l, j, sum : integer;
```

```
Begin
```

```
... {Вставить - ввод элементов массива и вывод в  
виде матрицы}
```

```
for j:=1 to m do begin
```

```
    sum:=0;
```

```
    for i:=1 to n do sum := sum +x[i,j];
```

```
    zum[ j ]:=sum
```

```
end;
```

```
for j:=1 to m do write (zum[ j ]:4);
```

```
End.
```