



# ОСНОВЫ ПРОГРАММИРОВАНИЯ

---

Учитель информатики и ИКТ  
ГОУ г.Москвы СОШ №310  
«У Чистых прудов»  
Цыбикова Т.Р.



Тема 7.



# АЛГОРИТМЫ ОБРАБОТКИ ТАБЛИЦ



# Двумерный массив (или прямоугольная таблица) $V$ из $n$ строк и $m$ столбцов

- Двумерный массив (или прямоугольная таблица)  $V$  из  $n$  строк и  $m$  столбцов в общем виде выглядит следующим образом:

$$\begin{matrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{matrix}$$

- На Паскале имена элементов массива записываются так же, с двумя номерами (индексами):  $b(1,1)$ ,  $b(1,2)$ , ...,  $b(1,m)$ ,  $b(2,1)$ ,  $b(2,2)$ , ...,  $b(2,m)$ , ...,  $b(n,m)$ .
- В памяти компьютера элементы двумерного массива расположены один за другим: после элементов первой строки следуют элементы второй строки и т.д.
- Если число строк таблицы равно числу столбцов, то такая таблица называется **квадратной**.
- Главная диагональ таблицы проходит из левого верхнего угла в правый нижний (выделена красным цветом).



# Рассмотрим задачи обработки таблиц и алгоритмы их решения.

## Задача 1.

- Для решения задачи надо выполнить следующие шаги:
  - 1) ввести таблицу в память;
  - 2) найти сумму элементов главной диагонали;
  - 3) напечатать результат.
- Описание таблицы, как и **описание** одномерного массива, используется для резервирования памяти.
- В описании указываются диапазоны для **двух номеров**: строк и столбцов.

```
const n=3;
```

```
var b: array [1..n, 1..n] of real; i,j: integer;
```

## Вычисление суммы элементов главной диагонали квадратной таблицы.

- При обработке массивов в разделе переменных программы появляются имена индексов элементов: для одномерного массива – одной, для двумерного – двух целочисленных переменных.
- При вычислении суммы элементов диагонали следует обратить внимание на имена суммируемых элементов: **оба индекса имеют одинаковые значения**, т.е. в общем виде имя элемента диагонали –  $b[i,i]$ . Это означает, что можно рассматривать диагональ, как одномерный массив, и использовать один цикл для вычислений





# Программа имеет вид:

```
program E16;  
const n=7;  
var b: array [1..n,1..n] of real; i,j: integer; S:real;  
begin  
  writeln('введите значения элементов таблицы по строкам');  
  writeln('в конце каждой строки нажимайте <Enter>');  
  for i:=1 to n do  
    begin  
      for j:=1 to n do  
        read(b[i,j]);  
      end;  
    S:=0;  
    for i:=1 to n do  
      S:=S+b[i,i];  
    write('сумма элементов диагонали таблицы S = ',S)  
  end.
```

```
введите значения элементов таблицы по строкам  
в конце каждой строки нажимайте <Enter>  
1 1 1 1 1 1 1  
2 2 2 2 2 2 2  
3 3 3 3 3 3 3  
4 4 4 4 4 4 4  
5 5 5 5 5 5 5  
6 6 6 6 6 6 6  
7 7 7 7 7 7 7  
сумма элементов диагонали таблицы S = 28
```



# Рассмотрим задачи обработки таблиц и алгоритмы их решения.

## Задача 2.

## Нахождение наибольших элементов каждой строки таблицы.

- Каждую строку таблицы можно рассматривать, как **одномерный массив**, и использовать идею нахождения наибольшего значения в программе E13.
- Найденные значения будем помещать в одномерный массив.
- В программе E17 для каждой строки таблицы переменная  $a[i]$  играет такую же роль, как переменная  $\max$  в программе E13.
- Для каждой переменной строки (ее задает переменная  $i$ ) элемент  $a[i]$  получает значение первого элемента строки.

- Затем внутренний цикл переменной  $j$  позволяет просмотреть все элементы данной строки и, если среди них встретится элемент, значение которого больше, чем запомненное в  $a[i]$ , то оно присваивается  $a[i]$ .
- Для распечатки результатов работы программы – массива  $a$  – используется **цикл**.
- Имеющийся в программе комментарий, заключенный в фигурные скобки, позволяет при прочтении программы выделить основную ее часть.



# Пример выполнения программы E17 - нахождения наибольших элементов строк:

## Задача 2.

Нахождение наибольших  
элементов каждой строки  
таблицы.

Массив результатов		Данная таблица		
a[1]	6	5	6	1
a[2]	15	4	12	15
a[3]	2	2	-3	0



```
program E17;
const n=3;
var b:array[1..n,1..n]of integer; i,j:integer;
    a:array[1..n]of integer;
begin
  writeln('введите значения элементов таблицы по строкам');
  writeln('в конце каждой строки нажимайте <Enter>');
  for i:=1 to n do
    begin
      for j:=1 to n do
        read(b[i,j]);
        writeln {в АВСПаскале не обязательный}
      end;
      {построение массива наибольших значений элементов строк таблицы}
      for i:=1 to n do
        begin
          a[i]:=b[i,1];
          for j:=2 to n do
            if a[i]<b[i,j] then a[i]:=b[i,j];
          end;
        end;
      writeln ('наибольшие числа строк таблицы:');
      for i:=1 to n do
        writeln(a[i])
      end.
end.
```

```
введите значения элементов таблицы по строкам
в конце каждой строки нажимайте <Enter>
5 6 1

4 12 15

2 -3 0

наибольшие числа строк таблицы:
6
15
2
```





# Рассмотрим задачи обработки таблиц и алгоритмы их решения.

## Задача 3.

## Нахождение сумм элементов столбцов таблицы.

- При обработке таблиц можно осуществлять операции как над строками, так и над столбцами.
  - Для нахождения сумм элементов столбцов можно использовать алгоритм примера E12.
- Чтобы лучше понимать работу программы,
    - введем переменную  $S$  для вычисления суммы,
    - а затем для каждого столбца запишем полученный результат в массив  $a$ , т.е. присвоим его переменной  $a[j]$ , где  $j$  – текущий номер столбцов таблицы.



# Пример выполнения программы E18 - суммирование по столбцам:

Данная таблица		
5	6	1
4	12	15
2	-3	0
Массив результатов		
$a[1]$	$a[2]$	$a[3]$
11	15	16



```
program E18;
const n=3;
var b:array[1..n,1..n] of integer; S,i,j:integer;
    a:array[1..n] of integer;
begin
  writeln('введите значения элементов таблицы по строкам');
  writeln('в конце каждой строки нажимайте <Enter>');
  for i:=1 to n do
    begin
      for j:=1 to n do
        read(b[i,j]);
      end;
      {построение массива сумм элементов столбцов таблицы}
      for j:=1 to n do
        begin
          S:=0;
          for i:=1 to n do
            S:=S+b[i,j];
            a[j]:=S;
          end;
          writeln ('суммы элементов столбцов таблицы:');
        end;
      for i:=1 to n do
        writeln(a[i])
      end;
    end.
```

```
введите значения элементов таблицы по строкам
в конце каждой строки нажимайте <Enter>
5 6 1
4 12 15
2 -3 0
суммы элементов столбцов таблицы:
11
15
16
```





# Рассмотрим задачи обработки таблиц и алгоритмы их решения.

## Задача 4.

- В прямоугольной таблице  $V$  из  $n$  строк и  $m$  столбцов требуется поменять местами две строки.
- При решении этой задачи можно воспользоваться алгоритмом обмена двух переменных из программы сортировки (пример E14).

## Перестановка строк таблицы.

- Для этого достаточно организовать цикл переменной столбца и, используя промежуточную переменную, менять местами каждую пару элементов, стоящих в одном столбце.
- При заданных номерах **строк  $K$  и  $L$**  решение выглядит так:



```
program E19;
const n=3; m=4;
var b:array[1..n,1..m] of real; c:real; i,j,K,L:integer;
begin
  writeln('введите номера меняемых местами строк');
  readln (K,L);
  {ввод таблицы}
  for i:=1 to n do
    begin
      writeln (i, '-я строка таблицы');
      for j:=1 to m do
        read (b[i,j]);
      end;
    end;
  {перестановка строк}
  for j:=1 to m do
    begin
      c:=b[K,j];
      b[K,j]:=b[L,j];
      b[L,j]:=c;
    end;
  {печать результатов}
  writeln;
  writeln ('таблица с переставленными строками:');
  for i:=1 to n do
    begin
      for j:=1 to m do
        write(b[i,j]);
      writeln
    end
  end.
end.
```

```
введите номера меняемых местами строк таблицы
1 3
1-я строка таблицы
1 1 1 1
2-я строка таблицы
2 2 2 2
3-я строка таблицы
3 3 3 3

таблица с переставленными строками:
3333
2222
1111
```



# Задания

1. В квадратной таблице, не содержащей отрицательных элементов, найдите корень квадратный из произведения диагональных элементов.
2. Найдите наибольший элемент квадратной таблицы.
3. Найдите наименьший элемент квадратной таблицы и замените его нулем.
4. В прямоугольной таблице замените все элементы их квадратами.
5. В целочисленной прямоугольной таблице увеличьте на 0,5 все отрицательные элементы.
6. В квадратной таблице найдите наибольший элемент диагонали.
7. Поменяйте местами первую и последнюю строки прямоугольной таблицы.
8. Найдите произведения элементов строк прямоугольной таблицы.



# Литература

- **А.А.Кузнецов, Н.В.Ипатова**  
«Основы информатики», 8-9 кл.:
  - Раздел 3. ОСНОВЫ ПРОГРАММИРОВАНИЯ,  
С.114-119