

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
ДВОРЕЦ ДЕТСКОГО (ЮНОШЕСКОГО) ТВОРЧЕСТВА ВЫБОРГСКОГО РАЙОНА
САНКТ-ПЕТЕРБУРГА
(ГБУ ДО ДДОТ)

Практические занятия
Модуль «Программирование на языке Pascal»

«Основы компьютерных технологий»

Второй год обучения
Группа № 1
Возраст учащихся 9 — 12 лет

Ухтина Елена Николаевна,
педагог дополнительного образования

Санкт-Петербург
2016 / 2017 учебный год

Цель занятий: познакомить учащихся с матрицами

Задачи:

Ⓢ *Обучающие:*

Ⓢ сформировать у учащихся первичные знания по применению изученного материала.

Ⓢ *Развивающие:*

Ⓢ учить анализировать, обобщать и систематизировать;

Ⓢ обогащать словарный запас учащихся.

Ⓢ *Воспитательные:*

Ⓢ развивать информационную культуру учащихся, способность к самостоятельной и коллективной деятельности, рефлексия.

Тип занятия: изучения и первичного закрепления новых знаний.

Оборудование: мультимедийный проектор, компьютеры с установленной средой программирования Pascal, настроенная локальная сеть, конспект для учащегося раздаточные карточки-задания.

Занятие 1. Матрицы. Количество двузначных чисел в матрице

Задача

Сформировать матрицу из чисел от 0 до 999, вывести ее на экран. Посчитать количество двузначных чисел в ней.

Пояснение к задаче и алгоритм решения

Следует до циклов, в которых заполняется матрица, ввести счетчик двузначных чисел. При заполнении каждой ячейки матрицы, следует проверять число. Если оно лежит в диапазоне от 10 до 99, значит оно двузначное, и следует увеличить счетчик на единицу.

Исходный код на языке программирования Pascal

```
const
M = 7;
N = 5;
var
a: array[1..N,1..M] of integer;
i, j, count: byte;
begin
randomize;
count := 0;
for i:=1 to N do begin
for j:=1 to M do begin
a[i,j] := random(1000);
write(a[i,j]:6);
if (a[i,j] > 9) and (a[i,j] < 100) then
count := count + 1
end;
writeln;
end;
writeln('Количество двузначных чисел: ', count)
end.
```

Особенности решения на языке программирования Pascal

Пример(ы) выполнения программы на языке Pascal:

```
481 356 456 213 842 82 959
59 661 52 729 932 915 663
178 224 751 362 36 121 977
473 466 470 31 288 356 147
941 138 436 971 891 616 465
```

Количество двузначных чисел: 5

Занятие 2. Матрица. Заполнение третьей матрицы по результатам сравнения

элементов первых двух

Задача

Две равноразмерные матрицы (например, 4x3) заполняются вводом с клавиатуры. В ячейки третьей матрицы такой же размерности записываются большие элементы из соответствующих ячеек первых двух матриц. Например, если во второй ячейке третьей строки первой матрицы находится число 89, а в ячейке с таким же индексом второй матрицы находится число 10, то в такую же ячейку третьей матрицы следует записать число 89.

Пояснение к задаче и алгоритм решения

Каждую из двух матриц следует заполнять в отдельном цикле. При заполнении третьей матрицы сравнивать ячейки с такими же адресами первых двух матриц и присваивать ячейке третьей матрицы наибольшее значение. В этом же цикле можно выводить третью матрицу на экран.

Исходный код на языке программирования Pascal

```
const
M = 4;
N = 3;
var
a, b, c: array[1..N,1..M] of integer;
i,j: byte;
begin
  writeln('Первая матрица: ');
  for i:=1 to N do
    for j:=1 to M do read(a[i,j]);
  writeln('Вторая матрица: ');
  for i:=1 to N do
    for j:=1 to M do read(b[i,j]);
  writeln('Результирующая матрица: ');
  for i:=1 to N do begin
    for j:=1 to M do begin
      if a[i,j] > b[i,j] then
        c[i,j] := a[i,j]
      else
        c[i,j] := b[i,j];
      write(c[i,j], ' ');
    end;
    writeln;
  end;
end.
```

Пример(ы) выполнения программы на языке Pascal

Первая матрица:

4 5 3 3

10 12 29 -1

-5 9 2 -5

Вторая матрица:

4 9 -2 2

3 7 4 -8

12 16 -1 3

Резльтирующая матрица:

4 9 3 3

10 12 29 -1

12 16 2 3

Занятие 3. Матрица. Вычислить элементы матрицы по формуле

Задача

Элементы матрицы $N \times M$ вычисляются по формуле $A[i,j] = \sin(N*i+M*j)$ при

индексации с единицы или по формуле $A[i,j] = \sin(N*(i+1)+M*(j+1))$ при индексации с нуля. Если полученный таким образом элемент матрицы отрицателен, то заменить его на 0. Вывести конечную матрицу на экран.

Пояснение к задаче и алгоритм решения

При заполнении матрицы каждый элемент вычисляется по заданной формуле. Если полученный таким образом элемент меньше нуля, то в соответствующую ему ячейку матрицы записывается 0.

В Python и Си требуется импортировать библиотеку с математическими функциями.

Исходный код на языке программирования Pascal

```
const N = 3; M = 4;
var
mtx: array[1..N,1..M] of real;
i,j: byte;
begin
for i:=1 to N do begin
for j:=1 to M do begin
mtx[i,j] := sin(N*i+M*j);
if mtx[i,j] < 0 then begin
mtx[i,j] := 0;
write(mtx[i,j]:7:0);
end
else
write(mtx[i,j]:7:2);
end;
writeln;
end;
end.
```

Пример(ы) выполнения программы на языке Pascal

```
0.66 0 0.65 0.15
0 0.99 0 0
0.42 0 0.84 0
```

Занятие 4. Матрица. Записать матрицу в одномерный массив

Задача

Записать элементов прямоугольной матрицы в одномерный массив в порядке

следования столбцов.

Пояснение к задаче и алгоритм решения

Размерность одномерного массива определяется произведением количества столбцов и строк матрицы. Так, если строк 3, а столбцов 4, то всего элементов в матрице 12. Следовательно, элементов в одномерном массиве должно быть также 12.

В программах ниже номера элементов одномерного массива (k) при его заполнении независимы от номеров строки (i) и столбца (j). Т.е. счетчик k увеличивается "вручную" при каждой записи элемента матрицы в одномерный массив. Возможно есть определенная зависимость между i и j с одной стороны и k - с другой.

Поскольку требуется переписать элементы матрицы в порядке следования столбцов, это значит, что сначала в массив записываются элементы первого столбца, затем второго, после третьего и т.д. Для этого матрица перебирается по столбцам. В таком случае внешний цикл - это проход по столбцам, а вложенный - по элементам столбца (т.е. по номерам строк).

Исходный код на языке программирования Pascal

```
const N = 3; M = 4;
var
  mtx: array[1..N,1..M] of byte;
  arr: array[1..N*M] of byte;
  i,j,k: byte;
begin
  randomize;
  for i:=1 to N do begin
    for j:=1 to M do begin
      mtx[i,j] := random(100);
      write(mtx[i,j]:3);
    end;
    writeln;
  end;
  writeln;
  k := 0;
  for j:=1 to M do
    for i:=1 to N do begin
      k := k + 1;
      arr[k] := mtx[i,j];
      write(arr[k]:3);
    end;
    writeln;
  end.
```

Пример(ы) выполнения программы на языке Pascal

```
23 88 30 8
99 55 40 89
58 3 27 22
```

```
23 99 58 88 55 3 30 40 27 8 89 22
```

Занятие 5. Матрица. Индексы минимальных элементов матрицы

Задача

Вывести на экран индексы всех минимальных элементов матрицы.

Пояснение к задаче и алгоритм решения

Эта задача отличается от поиска минимума тем, что нужно найти и вывести на экран не само минимальное значение, а его индексы (позицию, положение в матрице). Кроме того, минимальных (но равных между собой) значений в массиве может быть несколько. Следовательно, разумно вывести индексы всех минимальных элементов.

Задача складывается из двух подзадач, которые должны быть решены последовательно:

1. Поиск минимума в массиве (в данном случае двумерном).
2. Поиск элементов, равных ранее найденному минимуму.

Найденное минимальное значение должно быть сохранено в переменной (например, `minimum`). Однако можно было бы сохранять не само значение, а индекс элемента массива. Но поскольку мы имеем дело с матрицей, то пришлось бы сохранять два числа (номер строки и номер столбца).

Алгоритм поиска минимального значения:

1. Присвоить `minimum` максимально возможное (или больше) значение для исследуемого массива.
2. Перебрать элементы матрицы (используя конструкцию вложенного цикла). Каждый элемент сравнивать со значением `minimum`. Если очередной элемент меньше значения `minimum`, то следует присвоить значение текущего элемента переменной `minimum`.

Алгоритм определения позиций всех минимальных элементов матрицы:

1. Снова перебираем элементы матрицы.
2. Сравниваем каждый элемент со значением `minimum`.
3. Если они равны между собой, то выводим индексы текущего элемента на экран. (Индексы текущего элемента - это значения счетчиков первого и второго циклов.)

Примечания:

- Ⓢ Ниже в решениях задачи поиска индексов минимальных элементов на языках программирования поиск минимального значения выполняется в том же цикле, что и заполнение матрицы.
- Ⓢ Если индексация массива начинается с нуля, то лучше при выводе индексов увеличивать их на единицу. Так будет более ясно, где находится элемент.

Исходный код на языке программирования Pascal

```
const N = 5; M = 7;
var
  mx: array[1..N,1..M] of integer;
  min: integer;
  i, j: byte;
begin
  min := MAXINT;
  randomize;
  for i:=1 to N do begin
    for j:=1 to M do begin
```

```

mx[i,j] := random(50) - 25;
write(mx[i,j]:4);
if mx[i,j] < min then min:=mx[i,j];
end;
writeln;
end;
writeln('Минимальное значение: ', min);
for i:=1 to N do
for j:=1 to M do
if mx[i,j] = min then
writeln('строка: ', i, '; столбец: ', j);
end.

```

Пример(ы) выполнения программы на языке Pascal

```

-19 6 3 18 -12 -3 24
-4 15 -6 19 -15 -1 4
6 -9 -12 23 -3 3 -11
5 0 -11 -4 -19 -6 1
17 20 -1 6 17 -1 15

```

Минимальное значение: -19

строка: 1; столбец: 1

строка: 4; столбец: 5

Занятие 6. Матрица. Положительные элементы главной диагонали

Задача

Найти положительные элементы главной диагонали квадратной матрицы.

Пояснение к задаче и алгоритм решения

Диагонали выделяют только в том случае, если матрицы квадратные, то есть количество строк равно количеству столбцов. Элементы главной диагонали имеют индексы [1,1], [2,2], ... [N,N] (при условии индексации с единицы).

Поскольку первый индекс равен второму, то, перебирая матрицу в цикле, при обращении к элементам главной диагонали достаточно использовать одну переменную счетчик (например, *i*): имя_массива[*i*,*i*].

Если очередной элемент диагонали больше нуля, то выводим его на экран.

Исходный код на языке программирования Pascal

```
const
N = 10;
var
arr: array[1..N,1..N] of integer;
i,j: byte;
begin
randomize;
for i:=1 to N do begin // просто заполнение матрицы
for j:=1 to N do begin
arr[i,j] := random(100) - 50;
write(arr[i,j]:4);
end;
writeln;
end;
writeln;
for i:=1 to N do // решение задачи
if arr[i,i] > 0 then
write(arr[i,i]:4);
writeln;
end.
```

Пример(ы) выполнения программы на языке Pascal

```
-27 -12 37 -33 4 -44 44 5 21 -11
-46 -18 2 2 31 -36 47 -33 -11 41
-12 47 41 14 25 26 6 22 49 -48
-22 -7 -25 49 -13 -26 13 31 -15 -11
-5 30 31 -26 -48 26 44 1 -49 -4
22 -49 18 -20 -5 -25 0 -8 16 -19
-21 -5 30 -14 27 -35 -34 42 39 21
-2 -24 19 -48 -13 2 -35 44 -1 14
-47 45 -21 29 17 -9 27 -38 25 7
47 -20 -45 -22 -36 -43 12 5 40 -22

41 49 44 25
```