

Многоступенчатые циклические вычислительные процессы. Двумерные массивы.

2. Составить блок-схемы и написать программы для выполнения задач по работе с матрицами.

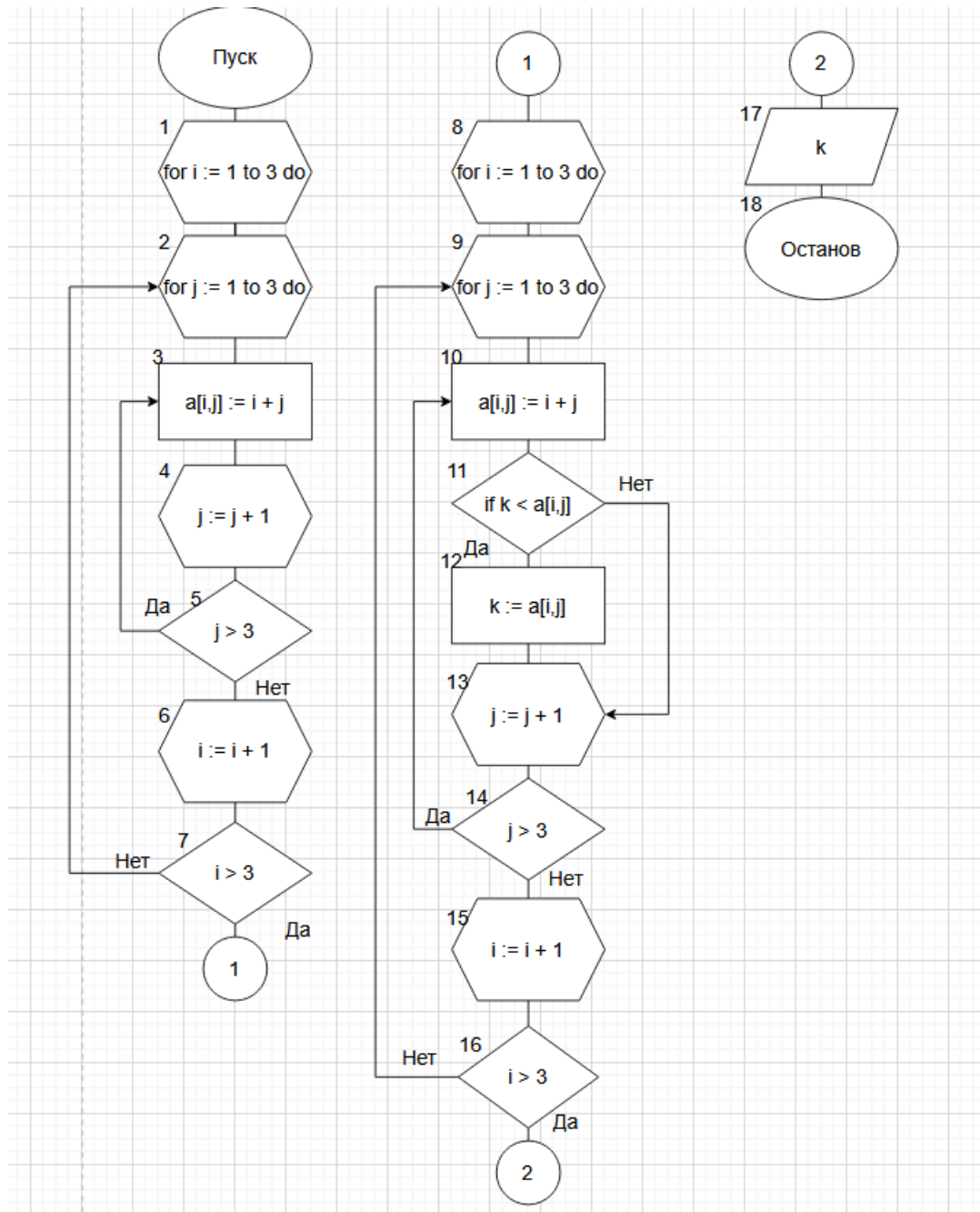
3. Я использовал среду программирования pascalABC для написания кода и сайт <https://app.diagrams.net> для создания блок-схемы

4.1

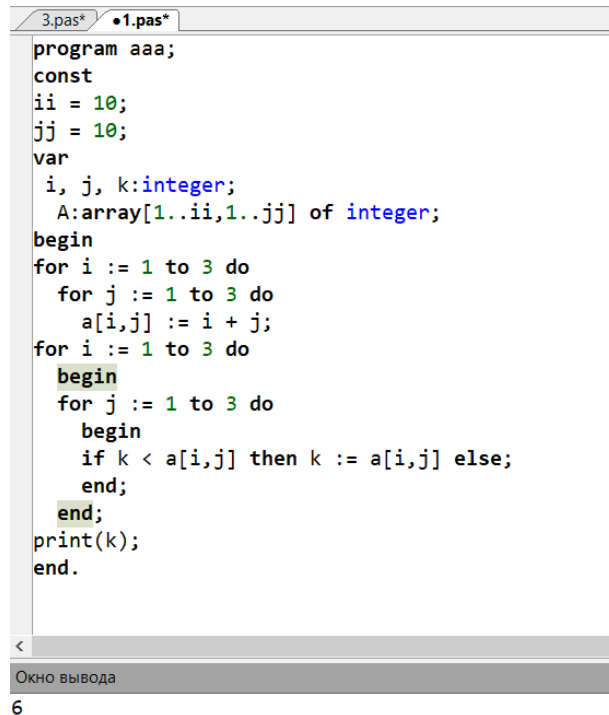
Найти сумму всех элементов массива 3x3. Массив задается явно внутри программы. Найти максимальный элемент.

5.1 $a_{i,j} = i + j$

6.1



7.1



```
3.pas*  •1.pas*
program aaa;
const
  ii = 10;
  jj = 10;
var
  i, j, k:integer;
  A:array[1..ii,1..jj] of integer;
begin
  for i := 1 to 3 do
    for j := 1 to 3 do
      a[i,j] := i + j;
  for i := 1 to 3 do
    begin
      for j := 1 to 3 do
        begin
          if k < a[i,j] then k := a[i,j] else;
        end;
      end;
    end;
  print(k);
end.
```

Окно вывода

6

8.1 —

9.1 В результате выполненной работы сделал блок-схему и написал программу для нахождения наибольшего элемента в матрице 3x3

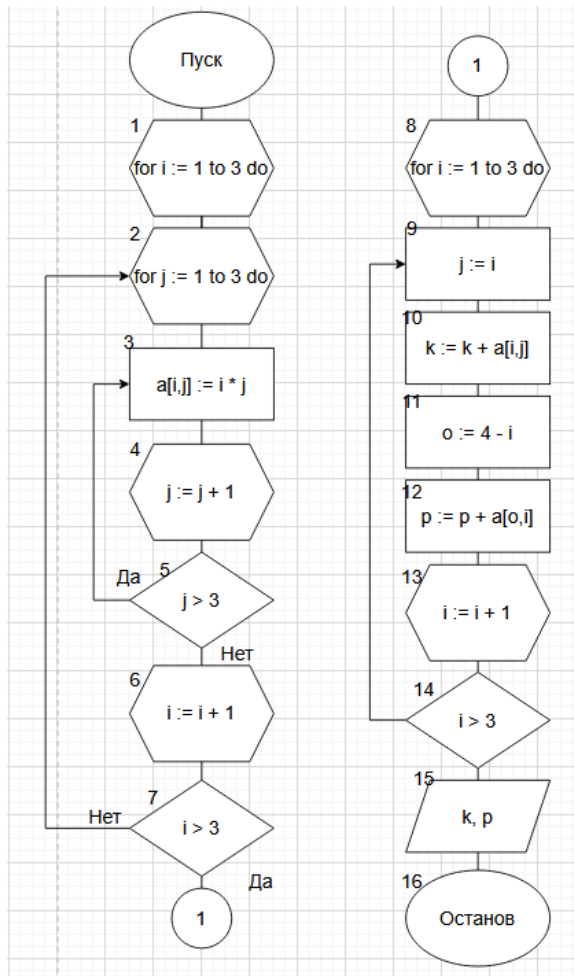
10.1 Мне понадобилось 2 цикла, что-бы перебирать строки и столбцы, и благодаря этому я смог узнать наибольший элемент.

4.2

Дан массив 3x3. Найти сумму элементов на главной диагонали и сумму элементов побочной диагонали.

$$5.2 \ a_{i,j} = i * j$$

6.2



7.2

```

3.pas* 1.pas* 2.pas
program aaa;
const
  ii = 10;
  jj = 10;
var
  i, j, k, p, o:integer;
  A:array[1..ii,1..jj] of integer;
begin
  for i := 1 to 3 do
    for j := 1 to 3 do
      a[i,j] := i * j;
  for i := 1 to 3 do
    begin
      j := i;
      k := k + a[i,j];
      o := 4 - i;
      p := p + a[o,i];
    end;
    print(k);
    print(p);
  end.

```

Окно вывода

14 10

8.2 –

9.2 В результате выполненной работы сделал блок-схему и написал программу для нахождения суммы чисел основной и побочной диагонали матрицы.

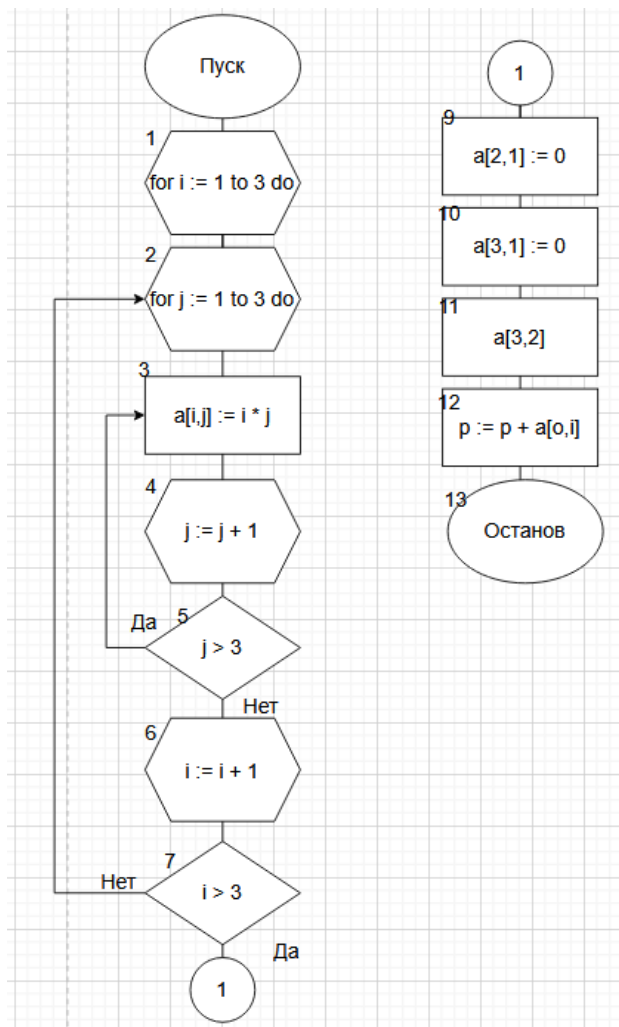
10.2 Для этого задания мне было достаточно 1 цикла и уже с помощью отдельных вычислений я смог получить столбец и строку побочной и основной диагонали матрицы, и же используя эти числа я вычислил сумму объектов матрицы.

4.3

Дан массив 3x3. Заменить элементы, стоящие ниже главной диагонали нулями.

$$5.3 \ a_{i,j} = i * j$$

6.3



7.3

```
•3.pas* 1.pas* 2.pas
program aaa;
const
  ii = 10;
  jj = 10;
var
  i, j:integer;
  A:array[1..ii,1..jj] of integer;
begin
  for i := 1 to 3 do
    for j := 1 to 3 do
      a[i,j] := i * j;
  a[2,1] := 0;
  a[3,1] := 0;
  a[3,2] := 0;

  for i := 1 to 3 do
    begin
      writeln();
      for j := 1 to 3 do
        begin
          write(a[i,j]:2);
        end;
      end;
    end.
end.
```

<

Окно вывода

```
1 2 3
0 4 6
0 0 9
```

8.3 –

9.3 2 В результате выполненной работы сделал блок-схему и написал программу которая делает все числа под главной диагональю матрицы нулями

10.3 В этой программе используются всего два цикла – для создания матрицы, больше циклы можно не использовать, ведь матрица не большая и это будет не целесообразно.

11.3 Из данной работы можно сделать вывод о том что при составлении и работе с матрицами надо использовать много циклов для проверки строк и столбцов