

## **Лабораторная работа №12**

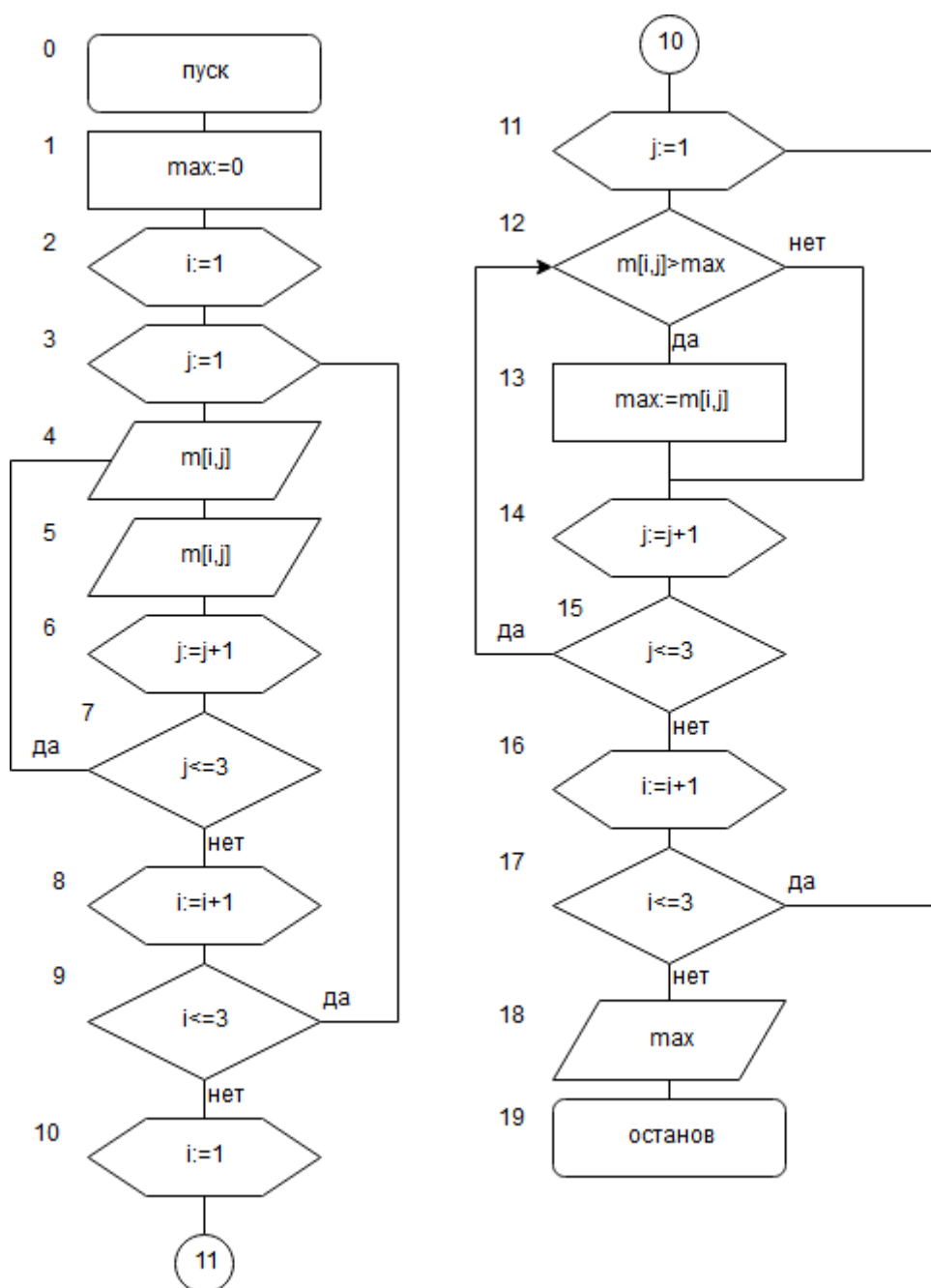
### **Многоступенчатые циклические вычислительные процессы.**

#### **Двумерные массивы.**

2. Цель работы: научиться решать задачи посредством многоступенчатых вычислительных процессов.
3. Используемое оборудование: ПК, Lazarus, PascalABC.

#### **Задание №1**

4. Найти сумму всех элементов массива  $3 \times 3$ . Массив задается явно внутри программы. Найти максимальный элемент.
5. if  $m[i,j] > \max$  then  $\max := m[i,j]$
- 6.



7.

Имя	Смысл	Тип
m	массив	Array of integer
i	Номер строки/параметр цикла	integer
j	Номер столбца/параметр цикла	integer
max	Максимальный элемент массива	integer

8.

```

1  program fdsal;
.  var m: array [1..3,1..3] of integer;
.    i,j,max: integer;
.  begin
5    randomize;
.    max:=0;
.    for i:=1 to 3 do
.      begin
.        for j:=1 to 3 do
10       begin
.         m[i,j]:=random(100);
.         write (m[i,j], ' ');
.       end;
.       writeln;
15     end;
.     for i:=1 to 3 do
.       for j:=1 to 3 do
.         if m[i,j]>max then max:=m[i,j];
.       write ('max=',max);
20     readln;
21   end.

```

9.

```

73 88 72
19 96 26
63 41 79
max=96

```

10. Для нахождения максимального элемента массива был организован многоступенчатый вычислительный процесс. Во внешнем цикле изменялась строка массива, во внутреннем – столбец. Если какой-то элемент при переборе был больше максимума – он становился максимумом.

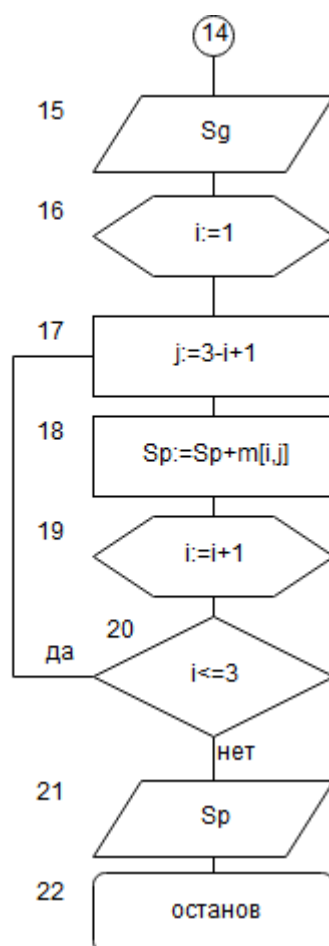
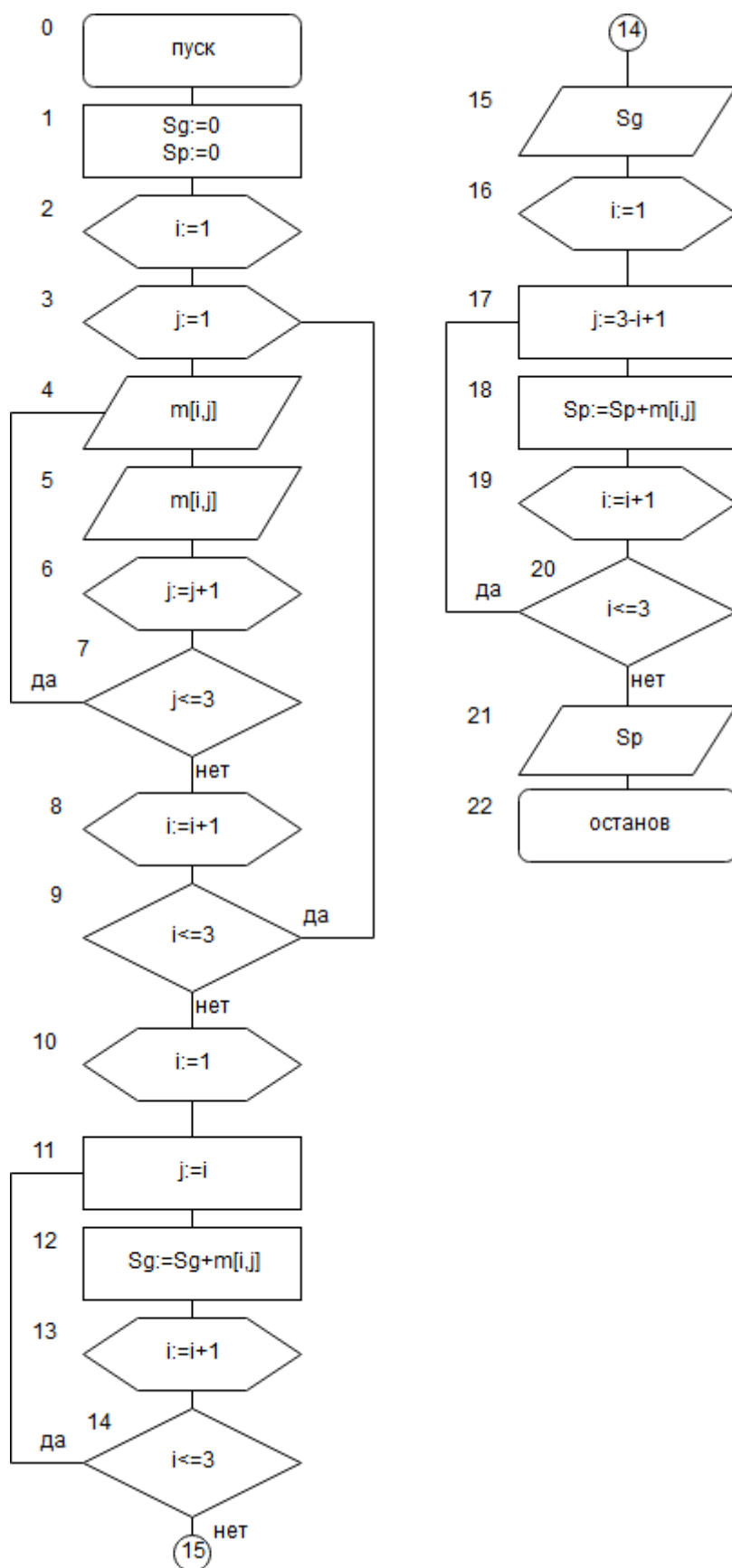
## Задание №2

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

5. Главная диагональ:  $j:=i$

Побочная диагональ:  $j:=n(\text{размерность массива})-i+1$

6.



7.

Имя	Смысл	Тип
m		Array of integer

i	Номер строки/параметр цикла	integer
j	Номер столбца/параметр цикла	integer
Sg	Сумма элементов главной диагонали	integer
Sp	Сумма элементов побочной диагонали	integer

8.

```

1  program sss2;
.  var m: array [1..3,1..3] of integer;
.  i,j,Sg,Sp: integer;
.  begin
5   Sg:=0;
.   Sp:=0;
.   randomize;
.   for i:=1 to 3 do
.   begin
10    for j:=1 to 3 do
.   begin
.       m[i,j]:=random(100);
.       write (m[i,j], ' ');
.       end;
15    writeln;
.    end;
.    for i:=1 to 3 do
.    begin
.        j:=i;
20    Sg:=Sg+m[i,j];
.    end;
.    writeln ('Summa elementov glavnoi diagonali=',Sg);
.    for i:=1 to 3 do
.    begin
25    j:=3-i+1;
.        Sp:=Sp+m[i,j];
.    end;
.    writeln ('Summa elementov pobochnoi diagonali=',Sp);
.    readln;
30 end. |

```

9.

```

7  60  67
70 67  9
94 96 49
Summa elementov glavnoi diagonali=123
Summa elementov pobochnoi diagonali=228

```

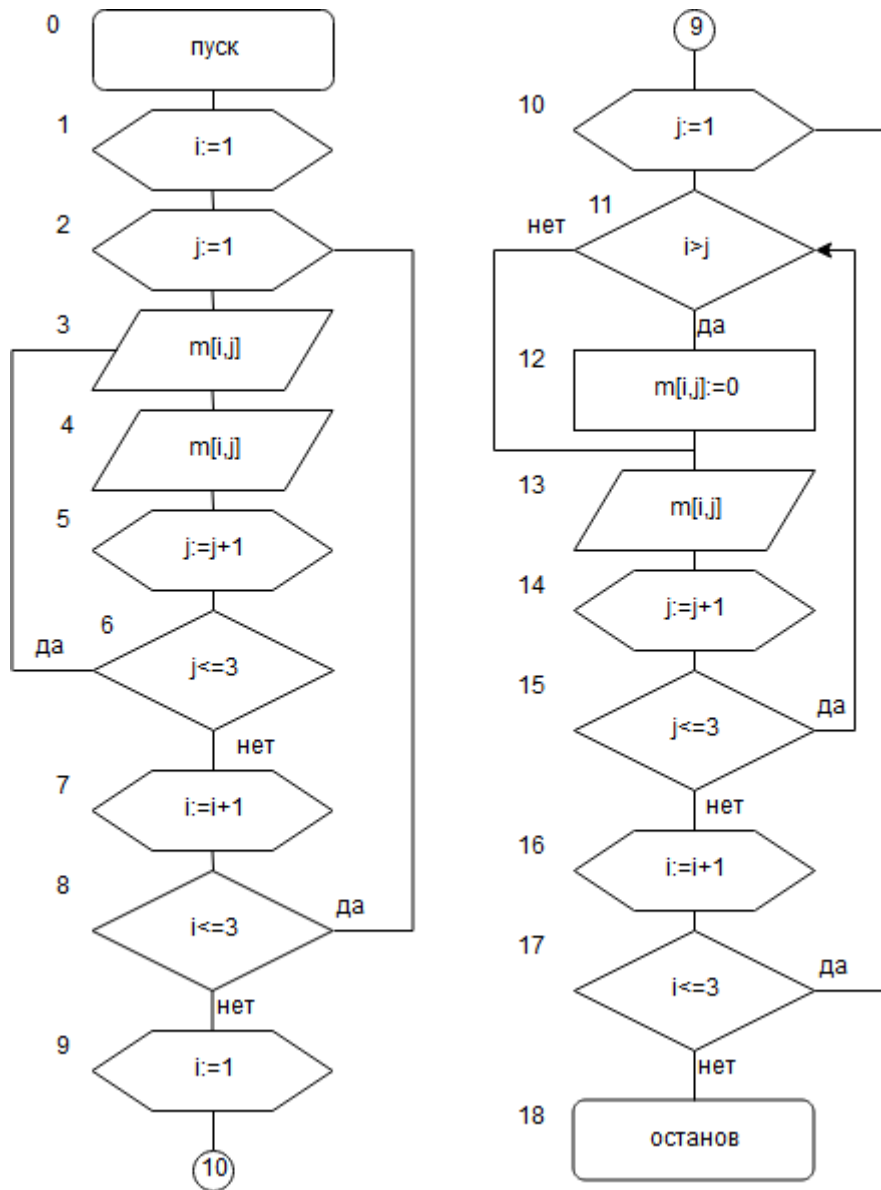
10. Для нахождения суммы главной диагонали в цикле складывались те элементы, у которых строка совпадала со столбцом. Для нахождения суммы побочной диагонали в цикле складывались те элементы, у которых столбцы равнялись  $3-i+1$ .

### Задание №3

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

5. if  $i > j$  then  $m[i,j] := 0$

6.



7.

Имя	Смысл	Тип
m	массив	Array of integer
i	Номер строки/параметр цикла	integer
j	Номер столбца/параметр цикла	integer

8.

```

1  program sss2;
.  var m: array [1..3,1..3] of integer;
.  i,j: integer;
.  begin
5  randomize;
.  for i:=1 to 3 do
.  begin
.  for j:=1 to 3 do
.  begin
10  m[i,j]:=random(100);
.  write (m[i,j], ' ');
.  end;
.  writeln;
.  end;
15  writeln;
.  for i:=1 to 3 do
.  begin
.  for j:=1 to 3 do
.  begin
20  if i>j then m[i,j]:=0;
.  write (m[i,j], ' ');
.  end;
.  writeln;
.  end;
25  readln;
26  end.

```

9.

```

17 43 79
27 73 89
42 16 80

17 43 79
0 73 89
0 0 80

```

10. На ноль заменялись те элементы, значение строки которых было больше, чем значение столбца (т.е. они находились под главной диагональю).

### Задание №4

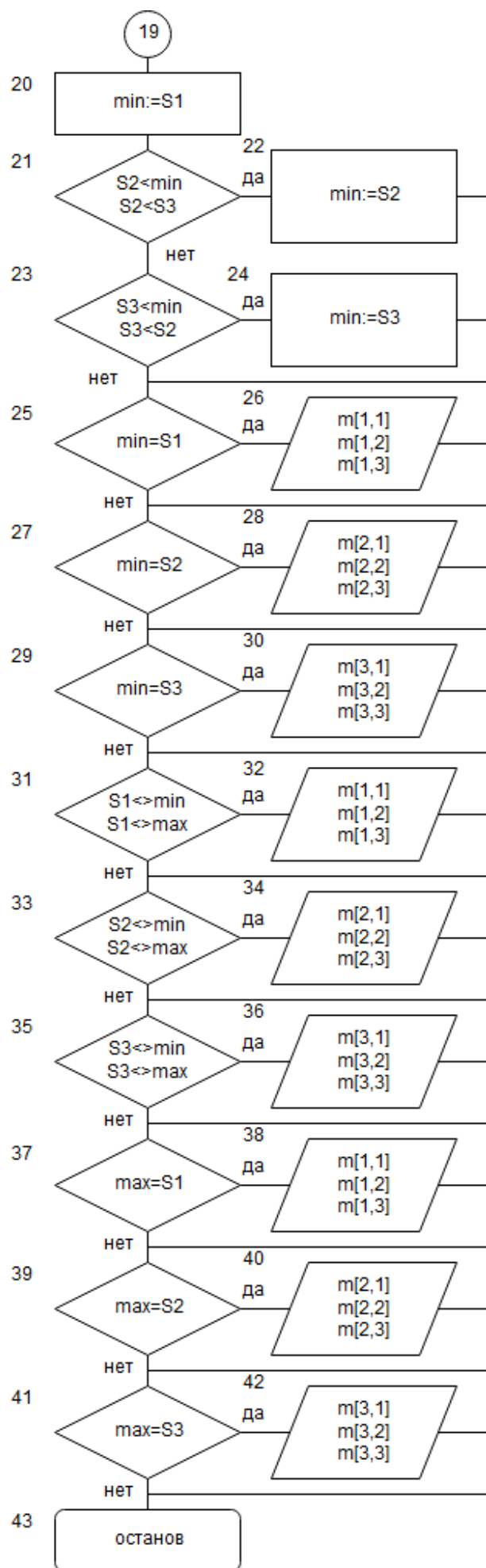
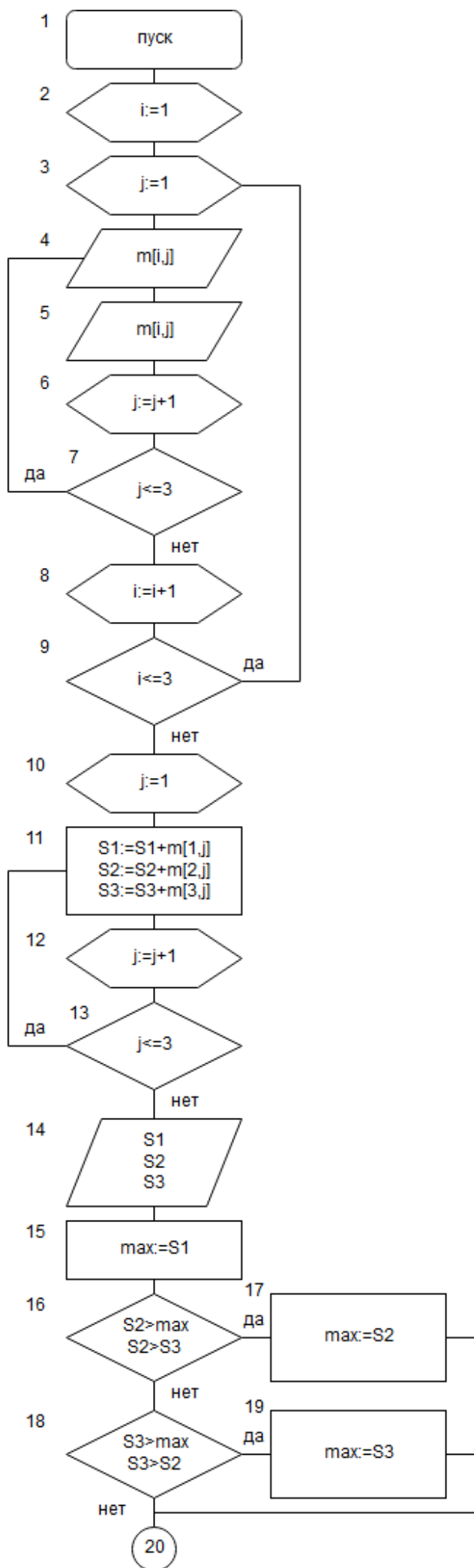
4. Дана матрица 3x3. Найти суммы элементов каждой строки и упорядочить строки по возрастанию согласно их суммам

5. S1:=S1+m[1,j]

S2:=S2+m[2,j]

S3:=S3+m[3,j]

6.





7.

Имя	Смысл	Тип
m	массив	Array of integer
i	Номер строки/параметр цикла	integer
j	Номер столбца/параметр цикла	integer
S1	Сумма элементов первой строки	integer
S2	Сумма элементов второй строки	integer
S3	Сумма элементов третьей строки	integer
max	Максимальная сумма элементов строки	integer
min	Минимальная сумма элементов строки	integer

8.

```

program sss4;
var m: array [1..3,1..3] of integer;
    i,j,S1,S2,S3,max,min: integer;
begin
    randomize;
    for i:=1 to 3 do
        begin
            for j:=1 to 3 do
                begin
                    m[i,j]:=random(100);
                    write (m[i,j], ' ');
                end;
            writeln;
        end;
    for j:=1 to 3 do
        begin
            S1:=S1+m[1,j];
            S2:=S2+m[2,j];
            S3:=S3+m[3,j];
        end;
    writeln;
    writeln ('Сумма элементов первой строки=',S1);
    writeln ('Сумма элементов второй строки=',S2);
    writeln ('Сумма элементов третьей строки=',S3);
    writeln;
    max:=S1;
    if (S2>max) and (S2>S3) then max:=S2;
    if (S3>max) and (S3>S2) then max:=S3;
    min:=S1;
    if (S2<min) and (S2<S3) then min:=S2;
    if (S3<min) and (S3<S2) then min:=S3;
    writeln ('отсортированная матрица');
    if min=S1 then write (m[1,1], ' ',m[1,2], ' ',m[1,3])
    else if min=S2 then write (m[2,1], ' ',m[2,2], ' ',m[2,3])
    else if min=S3 then write (m[3,1], ' ',m[3,2], ' ',m[3,3]);
    writeln;
    if (S1<>min) and (S1<>max) then write (m[1,1], ' ',m[1,2], ' ',m[1,3])
    else if (S2<>min) and (S2<>max) then write (m[2,1], ' ',m[2,2], ' ',m[2,3])
    else if (S3<>min) and (S3<>max) then write (m[3,1], ' ',m[3,2], ' ',m[3,3]);
    writeln;
    if max=S1 then write (m[1,1], ' ',m[1,2], ' ',m[1,3])
    else if max=S2 then write (m[2,1], ' ',m[2,2], ' ',m[2,3])
    else if max=S3 then write (m[3,1], ' ',m[3,2], ' ',m[3,3]);
end.

```

```
Окно вывода
43  4  37
14 15  2
11 53 37

Сумма элементов первой строки=84
Сумма элементов второй строки=31
Сумма элементов третьей строки=101

отсортированная матрица
14 15  2
43  4  37
11 53 37
```

10. Для сортировки массива сначала были найдены суммы элементов первой, второй и третьей строк. Далее переменным `min` и `max` были присвоены значения самой маленькой и самой большой суммы соответственно. Далее если сумма какой-либо строки совпадала с `min`, то она выводилась на экран. Потом если сумма какой-либо строки не совпадала ни с `min` ни с `max`, то она выводилась на экран. И если сумма какой-либо строки совпадала с `max`, то она выводилась на экран.

11. Вывод: Многоступенчатые вычислительные процессы отлично подходят для различных действий с двумерными массивами, потому что удобно параметром внешнего цикла назначать переменную, отвечающую за строки массива, а параметром внутреннего цикла – переменную отвечающую за столбцы массива.