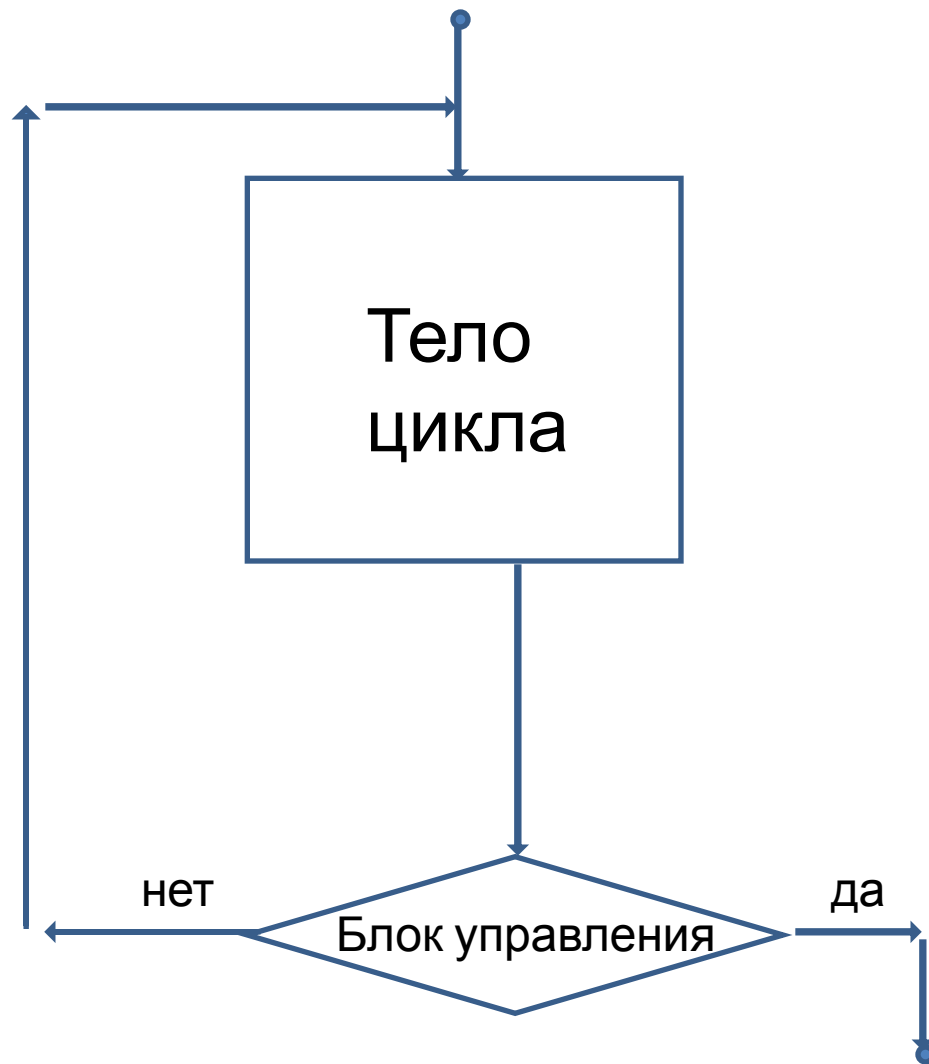


Оператор цикла с постусловием



- Оператор начинается словом **Repeat**.
- Затем следуют операторы, составляющие тело цикла.
- За ними записывается слово **Until** и логическое выражение, определяющее условие завершения цикла.

Repeat

<оператор> ;

<оператор> ;

...

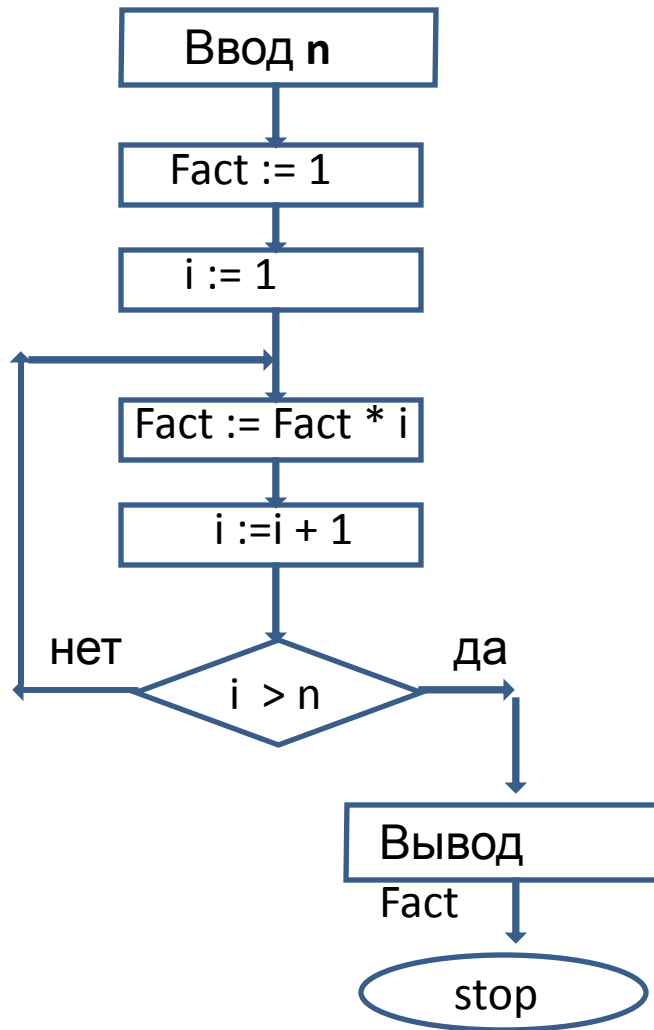
<оператор> ;

— тело цикла

Until <условие выхода из цикла> ;

- При работе такого цикла, сначала выполняются все операторы тела цикла, затем вычисляется логическое выражение, записанное после **Until**.
- Если значение этого выражения **False**, повторяется выполнение тела цикла.
- Если значение логического выражения **True**, цикл заканчивается.

Пример. Подсчёт факториала числа n.



Var i, n, Fact : integer;

Begin

Write('n=');

Readln(n);

Fact := 1;

i := 1;

{цикл с постусловием}

Repeat

Fact := Fact * i;

i := i + 1;

Until i > n;

Writeln('n! = ',Fact)

End.

Производные типы

Перечислимые типы

- Стандартные, определённые в Паскале, например: **INTEGER**, **CHAR** и другие.
- Задаваемые в программе.
Задаваемый в программе перечислимый тип определяется набором идентификаторов, перечисляемых через запятую в круглых скобках.
Каждый идентификатор - допустимое значение данного типа.

Пример задания перечислимого типа:

Type week = (sun, mon, tus, wen, the, fr, sat);

Var a : week;

К переменным перечислимого типа применимы функции **Pred** и **Succ**.

Тип - диапазон

- Каждый такой тип задается путем наложения ограничений на уже заданный перечислимый тип, который называется базовым.
- В качестве базового - может использоваться любой перечислимый тип: стандартный или определенный в программе.

Тип - диапазон задается указанием минимального и максимального значений разделенных двумя точками.

Например:

```
Type num = 5 .. 15;
```

```
Var k : num;
```

```
  i : 1 .. 10;
```

РЕГУЛЯРНЫЙ ТИП

МАССИВ

- Массив это упорядоченная совокупность однотипных компонентов, имеющая имя.
- Для того чтобы выбрать один компонент, необходимо указать имя массива и индекс, т.е. номер компонента в массиве.

A

A_1	A_2						
-------	-------	--	--	--	--	--	--

- Тип компонента называется базовым типом для массива.
- Для задания массива надо задать базовый тип и тип индекса.
- Тип индекса обязательно должен быть перечислимым ограниченным типом, как правило, это диапазон целых чисел.

Регулярный тип задается конструкцией:
array[<тип индекса>] **of** <тип
компонентов>;

Например:

Type gr = 1 .. 100 ;

vek = array[gr] of real;

Var x : vek;

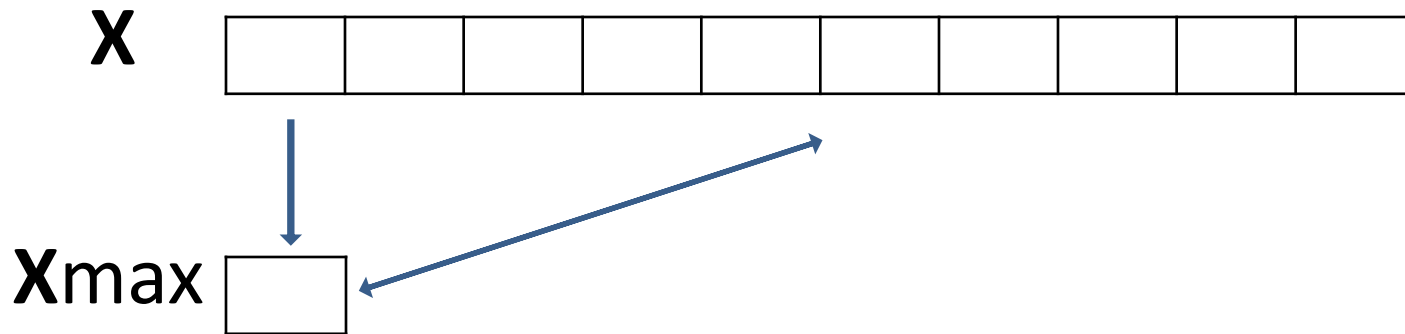
y : array[1 .. 50] of integer;

Обращение к элементам массива

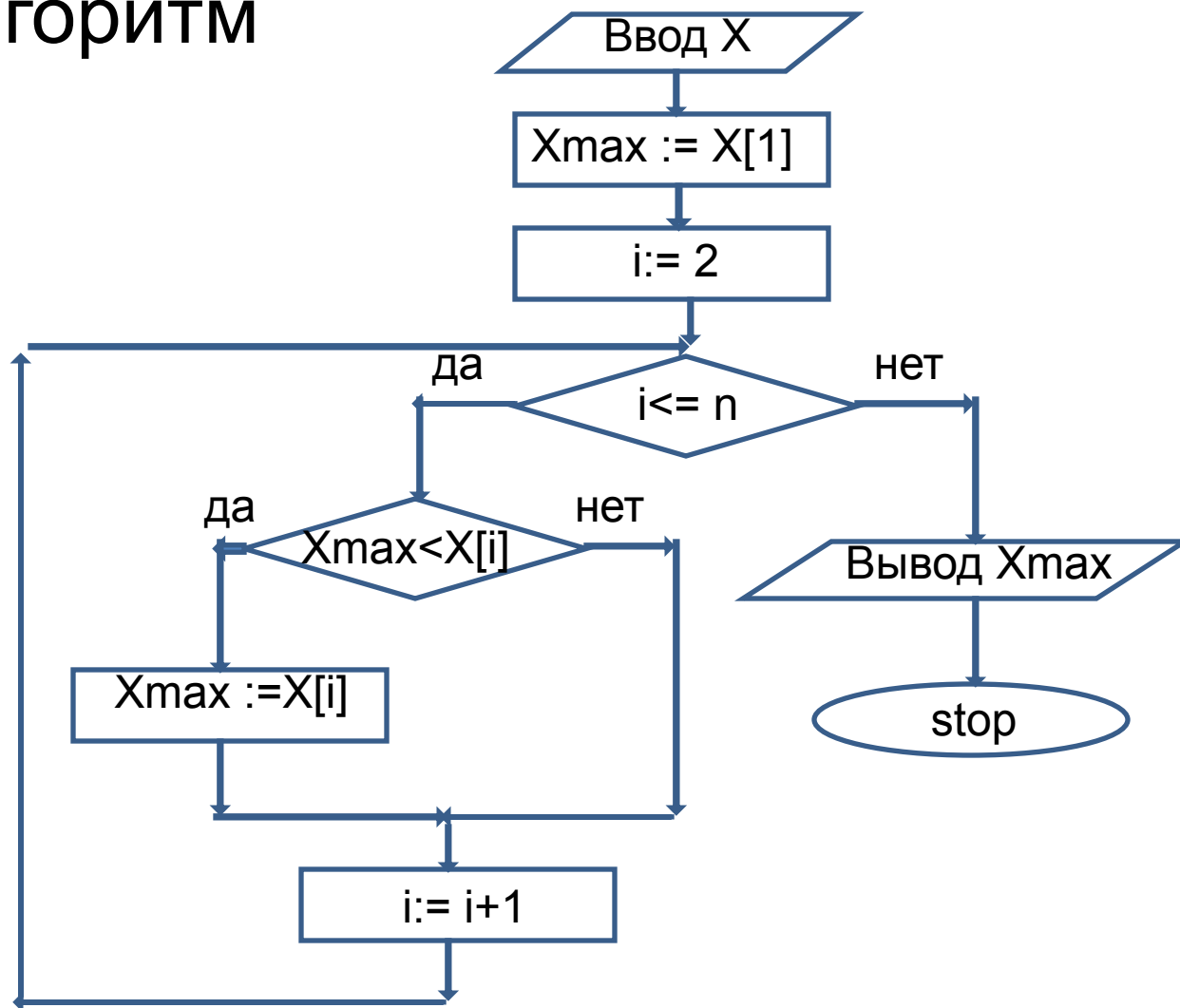
- Чтобы обратиться к элементу массива, надо написать имя массива и за ним в квадратных скобках индекс.
- Например: `x[10]`, `x[i]`, `x[i+k]`.
- Переменная с индексом может стоять в тех местах программы, где допустимо значение соответствующего базового типа.

Пример программы с использованием массива.

- Пусть задан x - массив из n чисел.
- Надо найти максимальное значение в массиве.



- Алгоритм



Program MAX;

Const n = 50;

Type vekt = array [1 .. n] of real;

Var i : 1 .. n;

 x : vekt;

 Xmax : real;

Begin

 {ВВОД МАССИВА x}

 Writeln('ВВЕДИТЕ МАССИВ x');

 For i := 1 To n Do

 Read(x[i]);

 Xmax := x[1];

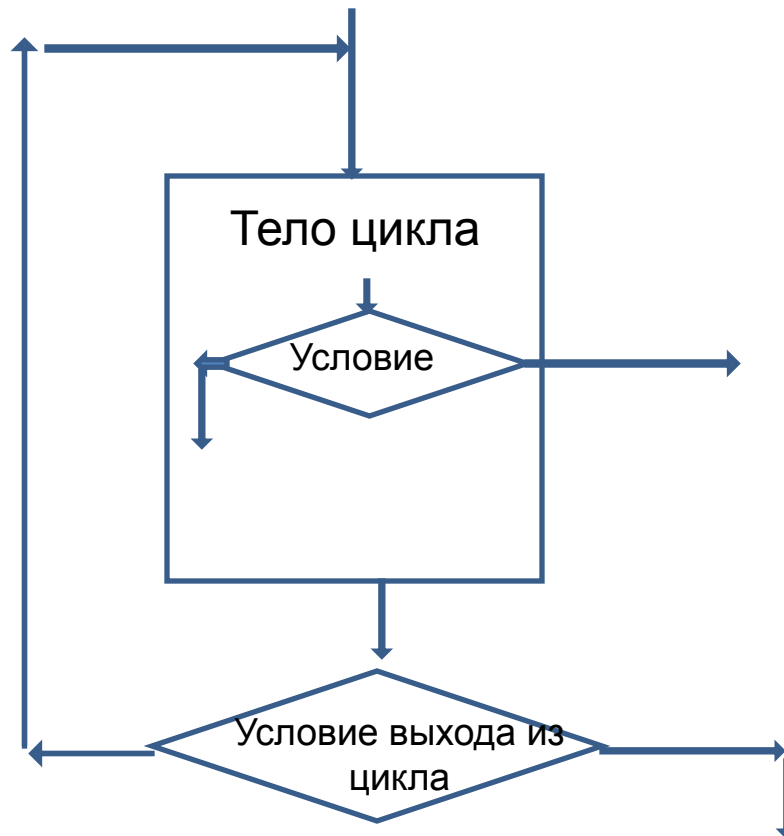
 For i := 2 To n Do

 If x[i] > Xmax Then Xmax := x[i];

 Writeln(' Xmax = ', Xmax)

End.

Досрочный выход из цикла



Не базовая структура

Метод флажка

- Вводится некоторая переменная (флажок), принимающая два значения.
- Одно значение, если выполнено условие досрочного выхода из цикла, другое значение, если надо продолжать вычисления в цикле.
- Проверка значения этой переменной добавляется к критерию выхода из цикла.

Пример.

В одномерном массиве найти под каким номером стоит первый чётный элемент.

Метод решения.

Искомой переменной присвоим значение ноль и организуем просмотр массива.

Если встретится чётный элемент, его номер надо запомнить в переменной и закончить процесс, то есть завершить цикл, используя флажок.

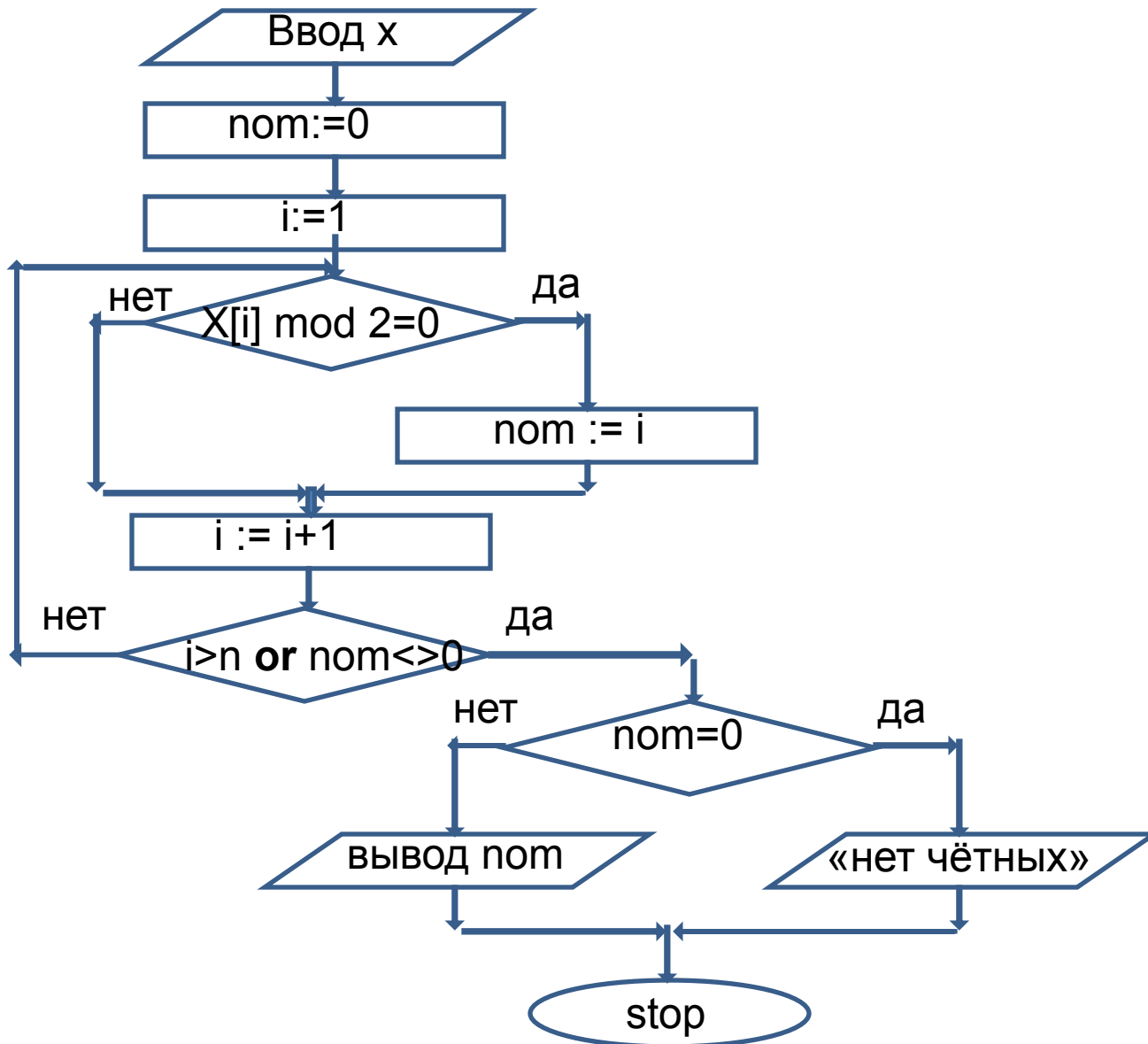
В данной задаче в качестве флажка можно использовать искомую переменную, она может иметь значение ноль, пока чётное число не найдено, или значение номера чётного числа.

x - исходный массив

n - число элементов **x**

nom - искомая переменная

i – номер исследуемого элемента массива
x



```
const n=8;  
var x: array[1..n] of integer;  
    nom,i:integer;  
begin  
    for i:=1 to n do  
        begin  
            write('x[', i, ']=');  
            readln(x[i]);  
        end;  
    nom:=0;  
    i:=1;  
    repeat  
        if x[i] mod 2 = 0 then nom:=i;  
        i:=i+1;  
    until (i > n) or (nom <> 0);  
    if nom = 0 then writeln(' not ')  
        else writeln('nom=', nom);  
end.
```