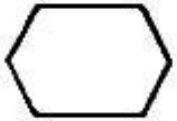
1. Схемы программ

Данные
Символ отображает данные, носитель данных не определен

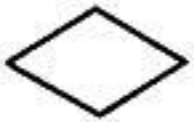
Процесс
Символ отображает функцию обработки данных любого вида

Предопределенный процесс
Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле)

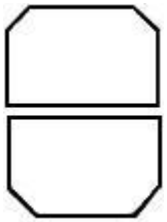


Подготовка
Символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы)

1. Схемы программ



Решение
Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий



Границы цикла
Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле)

Линия
Символ отображает поток данных или управления.
При необходимости или для повышения удобочитаемости могут быть добавлены стрелки-указатели

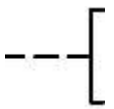
1. Схемы программ

Соединитель

Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение.

Терминатор

Символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).

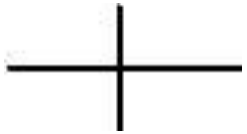


Комментарий

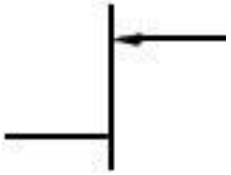
Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний.

1. Схемы программ

В схемах следует избегать пересечения линий. Пересекающиеся линии не имеют логической связи между собой, поэтому изменения направления в точках пересечения не допускаются.



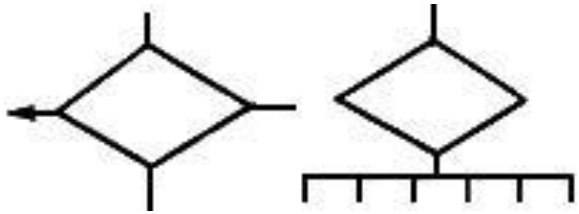
Две или более входящие линии могут объединяться в одну исходящую линию. Если две или более линии объединяются в одну линию, место объединения должно быть смещено.



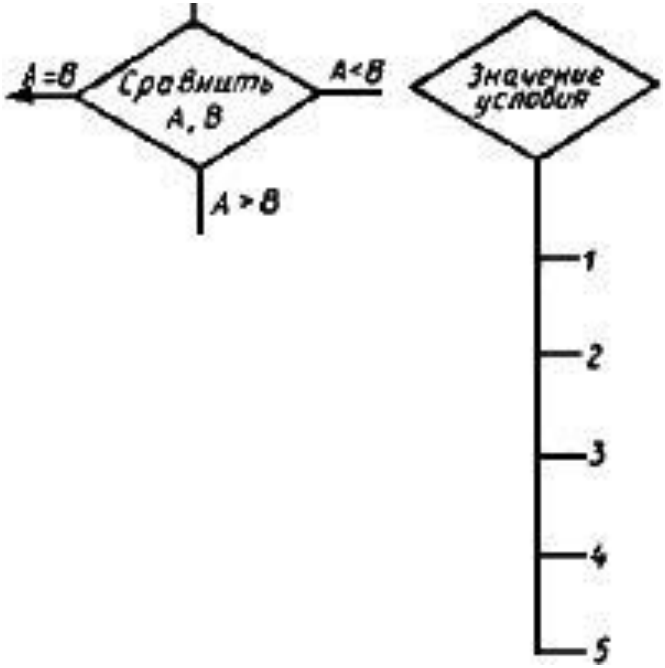
Линии в схемах должны подходить к символу либо слева, либо сверху, а исходить либо справа, либо снизу. Линии должны быть направлены к центру символа

1. Схемы программ

Несколько выходов из символа следует показывать:
1) несколькими линиями от данного символа к другим символам;
2) одной линией от данного символа, которая затем разветвляется в соответствующее число линий.

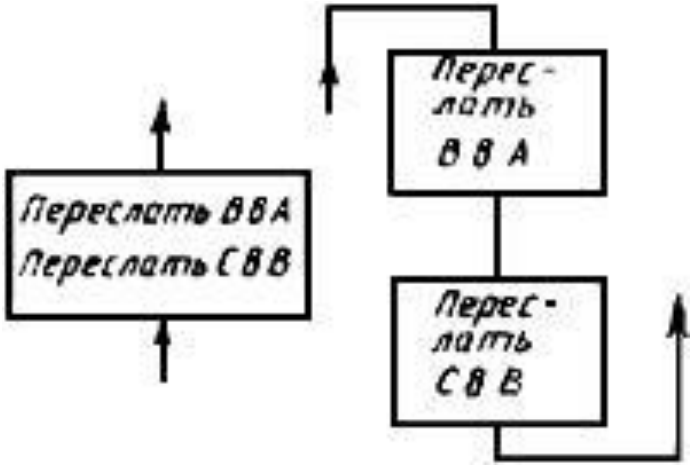


Каждый выход из символа должен сопровождаться соответствующими значениями условий, чтобы показать логический путь, который он представляет, с тем чтобы эти условия и соответствующие ссылки были идентифицированы.



1. Схемы программ

Текст для чтения должен записываться слева направо и сверху вниз независимо от направления потока.

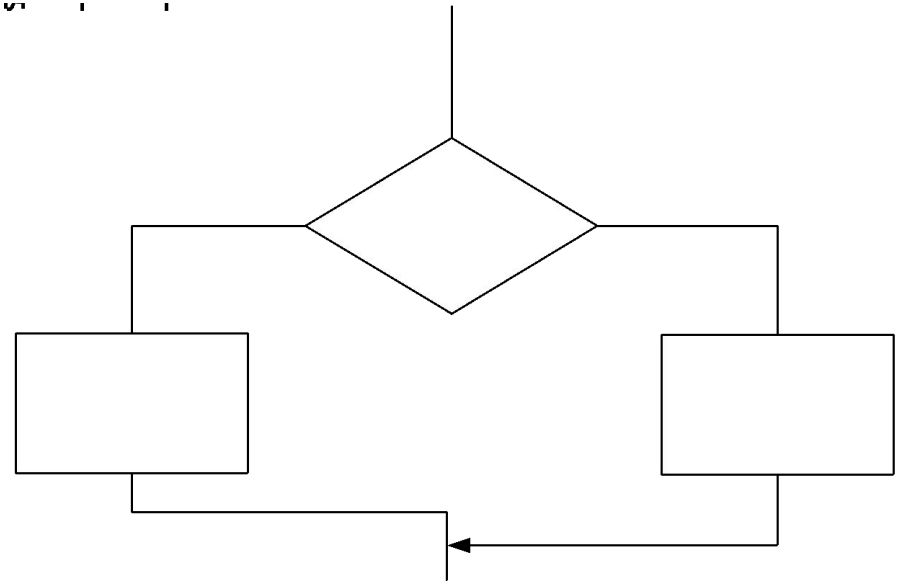


4. Условный оператор

Структура ***полной*** формы условного оператора:

IF <условие> THEN <оператор1> ELSE <оператор2>

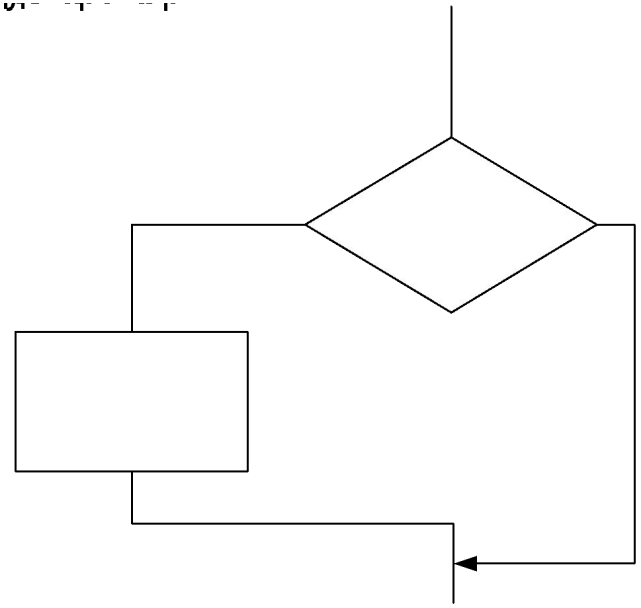
где IF, THEN, ELSE – зарезервированные слова (если, то, иначе);
<условие> – произвольное выражение логического типа;
<оператор1>, <оператор2> – любые операторы



4. Условный оператор

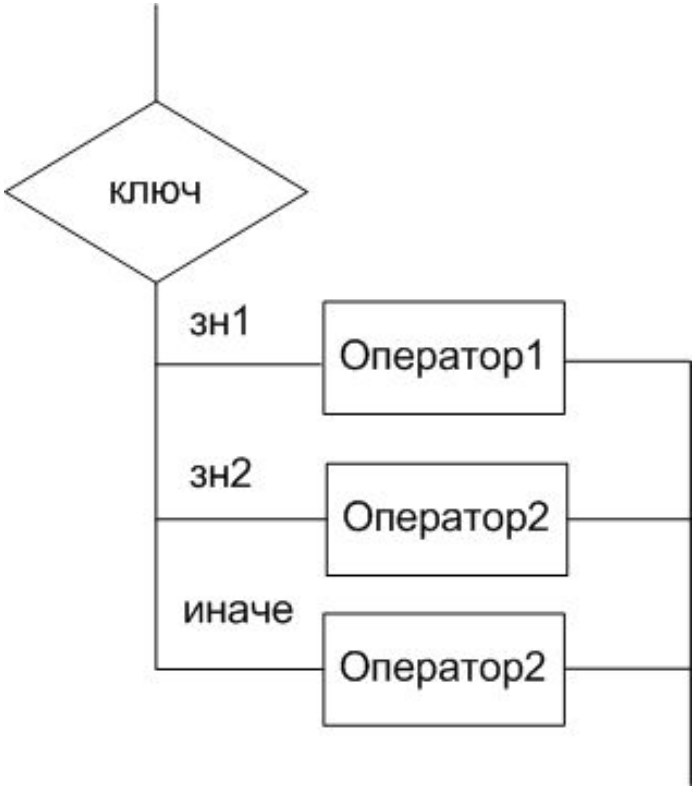
Структура ***краткой*** формы условного оператора:

IF <условие> THEN <оператор1>



5. Оператор выбора CASE

Схема алгоритма оператора выбора CASE



5. Оператор выбора CASE

```
var
  x,y,z: real;
  operation: char;  {знак операции}
  stop: boolean;    {признак ошибочной операции и останов}
begin
stop := false;
repeat
  writeln;          {пустая строка-разделитель}
  write('x, y ='); readln(x,y);
  write('операция:');
  readln(operation);
  case operation of
        '+': z := x+y;
        '-': z := x-y;
        '/': z := x/y;
       '*': z := x*y;
  else
       stop := true;
  end;
  if not stop then
    writeln('результат = ', z);
until stop;
end.
```

7. Оператор цикла for

Структура

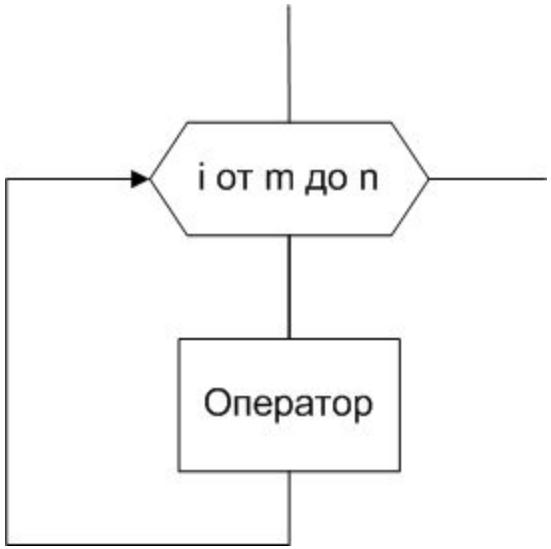
FOR <парам. цикла> := <нач. знач.> TO <кон. знач.> DO <оператор>;
FOR <парам. цикла> := <кон. знач.> DOWNTO <нач. знач.> DO <оператор>;

Здесь FOR, TO, DO – зарезервированные слова (для, до, выполнить)

<парам. цикла> – переменная перечисляемого типа;

<нач. знач.> и <кон. знач.> – выражения переменная перечисляемого типа;

<оператор> – произвольный оператор Паскаля



8. Оператор цикла while

Структура

WHILE <условие> DO <оператор>

Здесь WHILE, DO – зарезервированные слова (пока [выполняется условие], делать)

<условие> – произвольное выражение логического типа;

<оператор> – произвольный оператор.

9. Оператор цикла repeat..until

Структура

REPEAT <тело цикла> UNTIL <условие>

Здесь REPEAT, UNTIL – зарезервированные слова (повторять до тех пор, пока не будет выполнено условие)

<тело цикла> – произвольная последовательность операторов.

<условие> – произвольное выражение логического типа

Угадай число

Программа должна угадать число, которое загадал пользователь.

1. Определить диапазон поиска [a; b]
2. Вычислить c = (a+b)/2
3. Загаданное число совпадает с «c»? Да – останов.
4. Загаданное число ∈ [a;c]? Да – b = a. Нет – a = c
5. Переход в п.2.