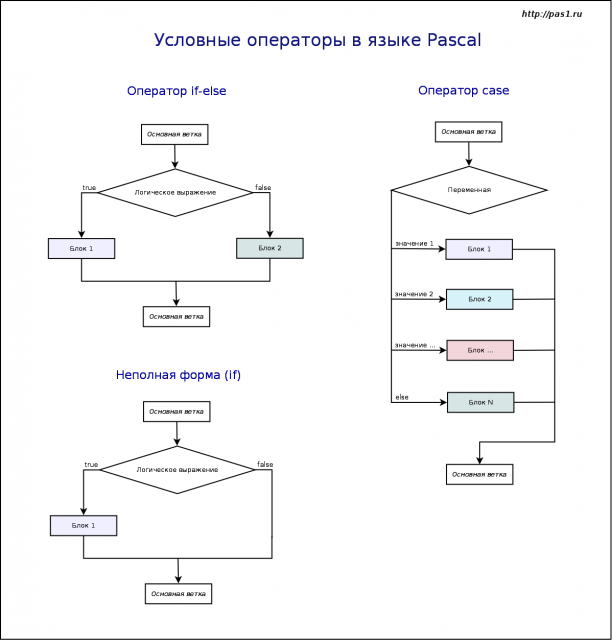
Условные операторы

Бывает, что в процессе выполнения программы требуется реализовать разный набор команд в зависимости от произошедших до этого событий. В языках программирования это достигается с помощью специальных конструкций –*условных операторов*.

Чаще всего в качестве условного оператора в языках программирования используется конструкция **if-else** или ее сокращенный вариант **if**. Также существует оператор выбора **case**, который имеет более специфичное применение.



Оператор if-else

Когда выполнение основной ветки программы доходит до условного оператора if-else, то в зависимости от результата логического выражения в его заголовке выполняются разные блоки кода. Если логическое выражение вернуло true, то выполняется один блок (в Паскале начинается со слова then), если false – то другой (начинается со слова else). После выполнения одного из вложенных блоков кода, ход программы возвращается в основную ветку. Другой вложенный блок не выполняется.

Например, программа должна определять, ввел пользователь четное или нечетное число, и выводить на экран сообщение. Тогда программный код на языке Pascal может быть таким:

**var** n: **integer**;

**begin**

write ('Введите целое число: ');

readln (n);

**if** n **mod** 2 = 0 **then**

write ('Оно четное.')

**else**

write ('Оно нечетное.');

readln

**end**.

Бывают неполные формы условных операторов. В таком случае вложенный в if блок кода выполняется только в случая true логическом выражении заголовка. В случае false выполнение программы сразу передается в основной блок. Понятно, что ветка else в таком случае отсутствует.

В примере ниже, если переменная имеет значение меньше нуля, то ее значение изменяется (находится модуль числа). Если же значение переменной изначально больше нуля, то блок кода при операторе if вообще не выполняется, т.к. не соблюдено условие (n<0).

**var** n: **integer**;

**begin**

write ('Введите целое число: ');

readln (n);

**if** n < 0 **then**

n := abs (n);

write (n);

readln

**end**.

[₽](https://direct.yandex.ru/?partner)Дистанционные курсы по медицине

В качестве условия может стоять любое выражение, результатом вычисления которого является одно из булевых значений — true или false.

Непосредственно после then может стоять только один оператор. При необходимости выполнения нескольких операторов они должны быть заключены в операторные скобки begin-end. Пример программы, которая меняет значения переменных местами, только если эти значения различны. Блок if содержит четыре выражения, поэтому заключен в begin-end.

**var**

a,b,c: **integer**;

**begin**

write('a=');

readln(a);

write('b=');

readln(b);

**if** a <> b **then** **begin**

c := a;

a := b;

b := c;

writeln('a=',a,'; b=',b);

**end**

**else**

writeln('Введены одинаковые числа');

readln;

**end**.

Допустимо вложение одного оператора if (или if-else) в другой. При этом следует соблюдать осторожность, т.е. бывает трудно определить какому if (внешнему или внутреннему) принадлежит ветка else. Рекомендуют использовать вложенную конструкцию if, только в ветке else. К тому же в языке Паскаль действует следующее правило: каждому then соответствует ближайшее else, не задействованное при установлении соответствия с другим then. Глубина вложенности операторов if может быть сколь угодно большой, но разобраться в таком коде будет очень сложно.

Оператор case (оператор выбора)

Кроме оператора if в языке программирования Паскаль предусмотрен так называемый переключатель case. Его можно трактовать как некий вопрос, имеющий большое число ответов (а не только два, как это имеет место в операторе if-else). Однако в отличие от if, case имеет ряд принципиальных ограничений. Его формат следующий:

**case** селектор **of**

значение1: оператор1;

значение2: оператор2;

значение3: оператор3;

...

**else** операторN

**end**;

В заголовке оператора case вместо логического выражения фигурирует переменная, которую называют селектором. До этого в программе ей присваивается какое-либо значение. Эта переменная может иметь только перечисляемый тип (например, она не может быть вещественного типа). По ходу выполнения оператора case, значение переменной-селектора сравнивается с различными, описанными в нем альтернативами (метками-значениями). Как только совпадение будет найдено, то выполняется блок кода при данной метке и происходит выход в основную ветку программы. Значения-метки являются константами, которые может принимать селектор. Их тип и тип селектора должны быть совместимы по присваиванию.

Если совпадений не будет, то выполняется блок else. Если блок else отсутствует (он является не обязательным), то никакой блок кода в операторе case не выполняется.

**var** n: **integer**;

**begin**

write ('Введите класс школы: ');

readln (n);

**case** n **of**

1..4: writeln ('Младшие классы.');

5..8: writeln ('Средняя школа.');

9,11: writeln ('Старшие классы. Выпускной.');

10: writeln ('Старшие классы.');

**else** writeln ('Error')

**end**;

readln

**end**.

На использование оператора выбора накладываются следующие ограничения:

* селектор должен иметь какой-либо порядковый тип;
* каждая альтернатива должна быть константой, диапазоном, списком диапазонов, но не переменной или выражением.