

# Оператор выбора (варианта)

Для реализации разветвляющихся алгоритмов, содержащих более двух ветвей, в некоторых случаях удобно использовать специальный оператор выбора (варианта).

## Формат оператора:

```
Case <выражение-селектор> of
    <СПИСОК МЕТОК 1> : <оператор 1>;
    <СПИСОК МЕТОК 2> : <оператор 2>;
    .....
    <СПИСОК МЕТОК n> : <оператор n>;
[else <оператор n+1>]
end;
```

- Здесь: **case, of, else, end** – служебные слова;



**<выражение-селектор>** – выражение любого скалярного типа, кроме вещественного (часто просто имя переменной);



**<СПИСОК МЕТОК>** - список разделенных запятыми значений, которые может принимать выражение – селектор, или одно его значение.

- При выполнении оператора выбора вычисляется значение выражения - селектора.
- Затем выполняется оператор на той из ветвей, которая содержит метку, совпадающую с полученным значением.
- Если вычисленное значение не совпало ни с одной из меток, то:
  - а) выполняется ветвь *else* в случае полной формы оператора;
  - б) если ветви *else* нет (неполная форма), то управление передается оператору, следующему за *case* (оператор выбора работает 'вхолостую').

- **Пример 1.** Составить программу, определяющую четность или нечетность остатка, полученного при делении целого положительного числа на 7. Если число делится на 7 без остатка, то также выдать соответствующее сообщение.
- Возможны три случая:
- остаток 0 – сообщение «делится нацело»;
- остаток 2,4,6 – сообщение «остаток четный»;
- остаток 1,3,5 – сообщение «остаток нечетный».

- `program ostatok_vibor;`
- `var a : word;`
- `begin`
- `write('введите целое положительное число');`
- `readln(a);`
- `case a mod 7 of`
- `2,4,6 : writeln('остаток четный');`
- `1,3,5 : writeln('остаток нечетный');`
- `0 : writeln('делится без остатка')`
- `end;`
- `end.`

## Пример 2

- Составить программу – простейший калькулятор, которая запрашивает знак арифметической операции (+, -, \*, /), затем числа, над которыми нужно произвести действие, и выводит результат.

- В данной программе возможны 4 ветви, причем в качестве условия, определяющего выбор той или иной ветви, выступает значение символьного типа (*char*) – знак арифметической операции.
- Поэтому будем использовать оператор *case*, в котором выражение-селектор – переменная с типа *char*.



- program calculyator;
- var c:char; x,y:integer; rez:real;
- begin
- write('введите знак операции (+,-,\*,/) ');
- readln(c);
- write('введите числа-операнды');
- readln(x,y);
- case c of
- '+' : rez:=x+y;
- '-' : rez:=x-y;
- '\*' : rez:=x\*y;
- '/' : rez:=x/y;
- end;
- writeln('результат =', rez:5:2);
- end.