

Базовые типы алгоритмов.

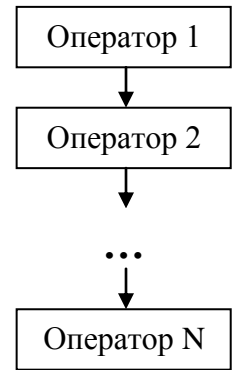
Базовые структуры алгоритмов:

- Линейная (следование).
- Разветвляющаяся (ветвление).
- Циклическая.

Линейный алгоритм – последовательное выполнение команд в порядке их следования.

Разветвляющийся алгоритм – выполнение алгоритма определяется результатами проверки некоторых условий.

Циклический алгоритм – отдельные команды или группы команд повторяются многократно



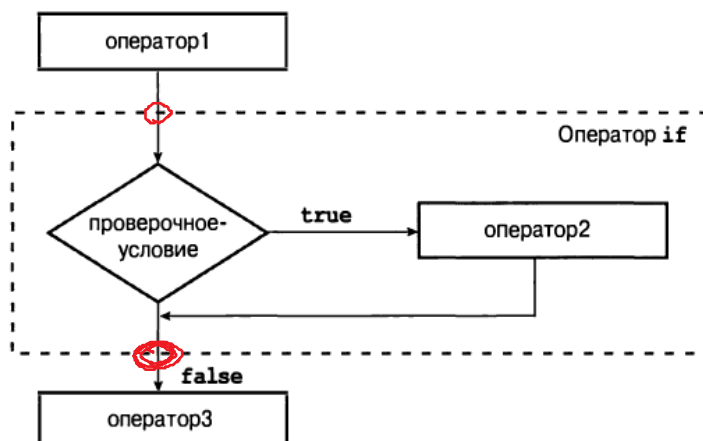
Условный оператор *if*. Сокращённая запись:

```
if (условие)  
    оператор;
```

или

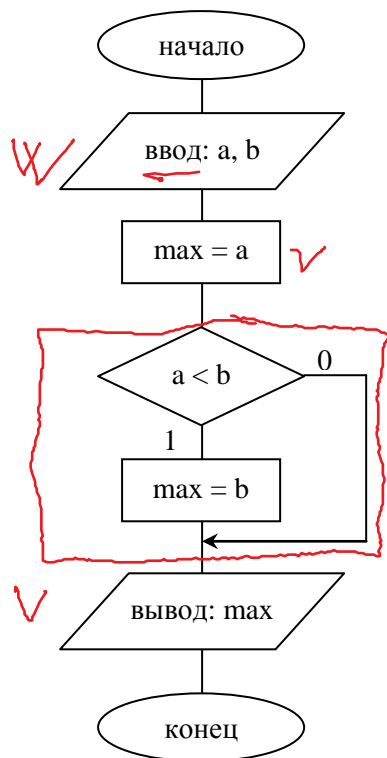
```
if (условие)  
{  
    Оператор1;  
    ...  
    ОператорN;  
}
```

оператор1
if (проверочное-условие)
 оператор2
 оператор3



Пример 1. Найти максимум из двух чисел.

Математическое обоснование: $\max(a, b) = \begin{cases} a, & \text{если } a > b \\ b, & \text{если } b > a \end{cases}$



```
1  #include <iostream>
2  using namespace std;
3
4  int main ()
5  {
6      int a, b, max;
7      cout << "Введите два числа:";
8      cin >> a >> b;
9      max = a;
10     if (a < b)
11         max = b;
12     cout << "Максимум = " << max;
13     cin.get();
14     return 0;
15 }
```

min

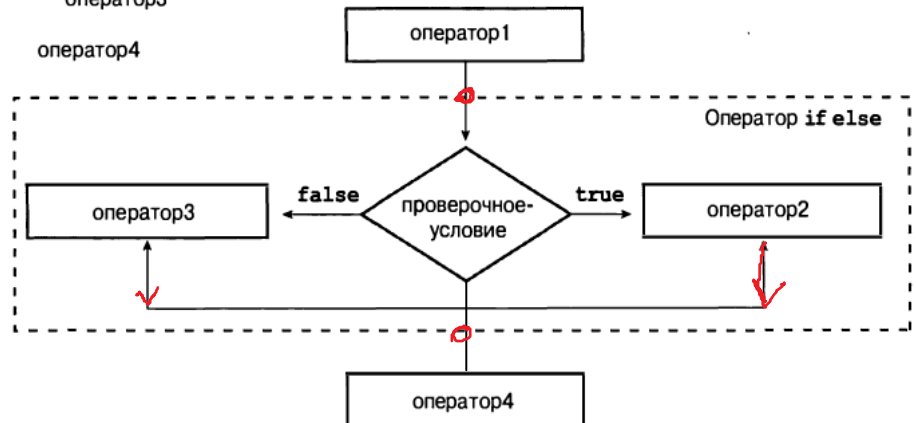
Условный оператор *if*. Полная запись:

```
if (условие)  
    оператор;  
else  
    оператор;
```

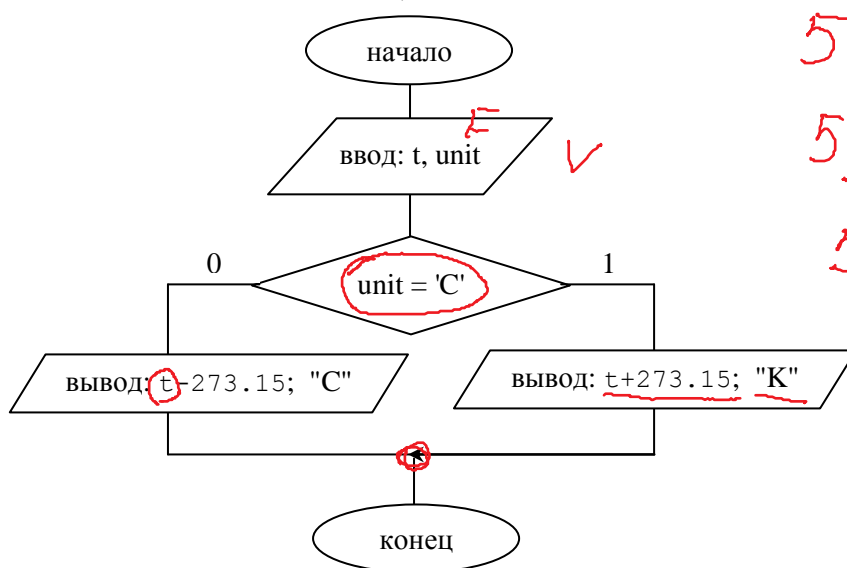
или

```
if (условие)  
{  
    Оператор1;  
    ...  
    ОператорN;  
}  
else  
{  
    Оператор1;  
    ...  
    ОператорN;  
}
```

оператор1
if (проверочное-условие)
 оператор2
else
 оператор3
оператор4



Пример 2. Преобразовать температуру из градусов Кельвина в градусы по Цельсию и наоборот.
 Математическое обоснование: $^{\circ}\text{K} = 273,15 + ^{\circ}\text{C}$.



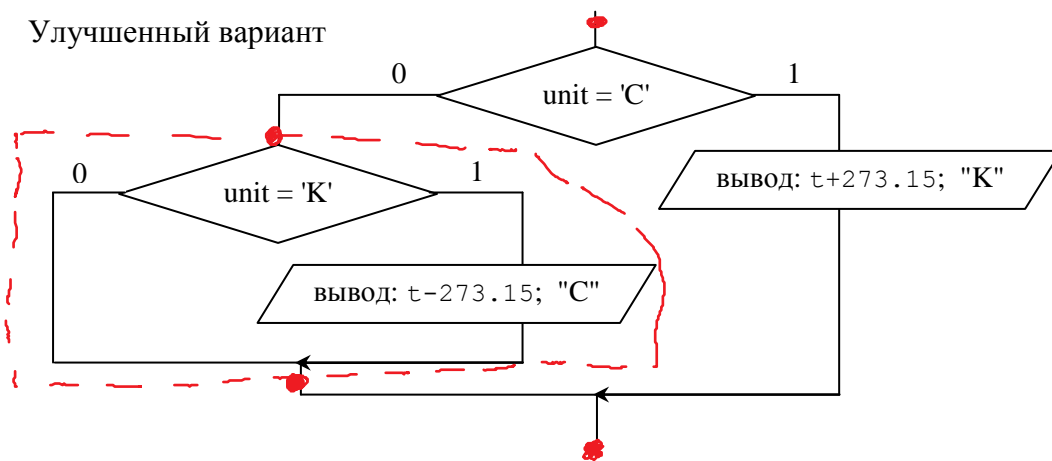
```

float C_K = 273.15;
float t;
char unit;
    
```

```

cout << "Ведите температуру и единицу измерения: " << endl;
cin >> t >> unit;
if (unit == 'C')
    cout << t << "C = " << t+C_K << " K" << endl;
else
    cout << t << "K = " << t-C_K << " C" << endl;
    
```

Улучшенный вариант



```
if (unit == 'C')
    cout << t << "C = " << t+C_K << " K" << endl;
else
    if (unit == 'K')
        cout << t << "K = " << t-C_K << " C" << endl;
    else
        cout << "Неизвестная единица измерения.";
```

Форматирование операторов if-else

1. операторные скобки (фигурные скобки):

```
if (ch == ' Z')
    zorro++; // if заканчивается здесь
    cout << "Another Zorro candidate\n";
V else // неверно
    dull++;
    cout << "Not a Zorro candidate\n";
```

```
if (ch == 'Z')
{ // если условие истинно
}
    zorro++;
    cout << "Another Zorro candidate\n";
else
{ // если условие ложно
}
    dull++;
    cout << "Not a Zorro candidate\n";
}
```

```
if (ch == 'Z') {
    zorro++;
    cout << "Another Zorro candidate\n";
}
else {
    dull++;
    cout << "Not a Zorro candidate\n";
}
```

Форматирование операторов if-else

2.

```
if (ch == 'A')  
    a_grade++; // альтернатива # 1  
else  
    if (ch == 'B') // альтернатива # 2  
        b_grade++; // подальтернатива # 2a  
    else  
        soso++; // подальтернатива # 2b
```

```
if (ch == 'A')  
    a_grade++; // альтернатива # 1  
else if (ch == 'B')  
    b_grade++; // альтернатива # 2  
else  
    soso++; // альтернатива # 3
```


Условные конструкции и предотвращение ошибок.

`unit == 'C'`

переменная == значение
значение == переменная

==
==

F9

```
float C_K = 273.15;  
float t;  
char unit;
```

```
cout << "Ведите температуру и единицу измерения: " << endl;
```

```
cin >> t >> unit; unit = 'C' unit = 'C'
```

```
if (unit == 'C')
```

```
    cout << t << "C = " << t+C_K << " K" << endl;
```

```
else
```

```
    if (unit == 'K')
```

```
        cout << t << "K = " << t-C_K << " C" << endl;
```

```
    else
```

```
        cout << "Неизвестная единица измерения.";
```

5C →

5K

C = 278,15 K

`unit == 'C'`

`'C' == unit`

`'C' = unit`

Логические выражения в условном операторе

- логическое «ИЛИ» – `||`;
- логическое «И» – `&&`;
- логическое «НЕ» – `!`;

Таблицы истинности операций:

A	B	A ИЛИ B	A И B	НЕ A
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

false
true

a	не a
0	1
1	0

Приоритеты:

1. ИЛИ и И ниже, чем у операторов сравнения.

`x > 5 && x < 10` \Leftrightarrow `(x > 5) && (x < 10)`

2. НЕ (!) имеет более высокий приоритет, чем любая арифметическая операция и операция сравнения:

\rightarrow `!(x > 5)` // равно false, если x больше 5

`!x > 5` // равно true, если !x больше 5. Всегда верное неравенство!

3. И имеет более высокий приоритет, чем ИЛИ:

\rightarrow `age > 30 && age < 45 || weight > 300`

\Leftrightarrow

`(age > 30 && age < 45) || weight > 300`

Скобки улучшают читаемость кода!!!

$x = 13 \rightarrow \text{true}$
 $x = 0 \rightarrow \text{false}$

Вычисление логических выражений: слева направо

1. `x != 0 && (1.0 / x) > 100.0`

✓ 2. `i++ < 6 || i == j` ✓

$!x -$
 $\text{false} > 5$
 \downarrow
 $0 > 5$

Альтернативные представления

`&&` – and

`||` – or

`!` – not

$i++$

$i = i + 1$

$x = 0$

false && ? \rightarrow 0 false

Тернарный оператор ?:

Синтаксис

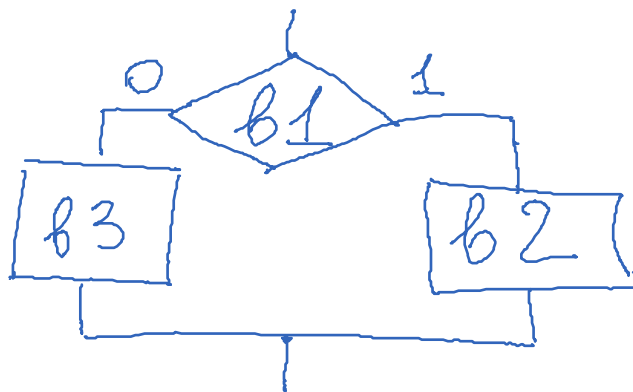
выражение1 ? выражение2 : выражение3

✓

→ $5 > 3 ? 10 : 12$ // $5 > 3$ истинно, поэтому значением всего выражения будет 10
 $3 == 9 ? 25 : 18$ // $3 == 9$ ложно, поэтому значением всего выражения будет 18

Пример. Найти максимум и минимум двух чисел.

```
int a, b;  
cout << "Введите два числа: ";  
cin >> a >> b;  
cout << "max=" << ((a > b) ? a : b) << ", "  
      << "min=" << ((a < b) ? a : b) << endl;
```



if (b1)
 b2; 10;
else
 b3; 12;

Инструкция switch

```
switch (выражение)
{
    case метка1: оператор(ы);
    ...
    case меткаN: оператор(ы);
    [default: оператор (ы);]
}
```

Пример. Температура в °C, °K и °F.

`float t;`
`char unit;`

`cout << "Введите величину и единицу измерения температуры: " << endl;`
`cin >> t >> unit;`

```
switch (unit) {
    case 'C':
        cout << t << "C = " << t+273.15 << " K" << endl;
        cout << t << "C = " << t*9/5+32 << " F" << endl;
        break;
    case 'K':
        cout << t << "K = " << t-273.15 << " C" << endl;
        cout << t << "K = " << t*9/5-459.67 << " F" << endl;
        break;
    case 'F':
        cout << t << "F = " << (t-32)/1.8 << " C" << endl;
        cout << t << "F = " << (t+459.67)/1.8 << " K" << endl;
        break;
    default: cout << "Неизвестная единица измерения.";
```

Пример. Использование множества меток

```
switch (unit) {
    case 'c':
    case 'C':
        cout << t << "C = " << t+273.15 << " K" << endl;
        cout << t << "C = " << t*9/5+32 << " F" << endl;
        break;
```