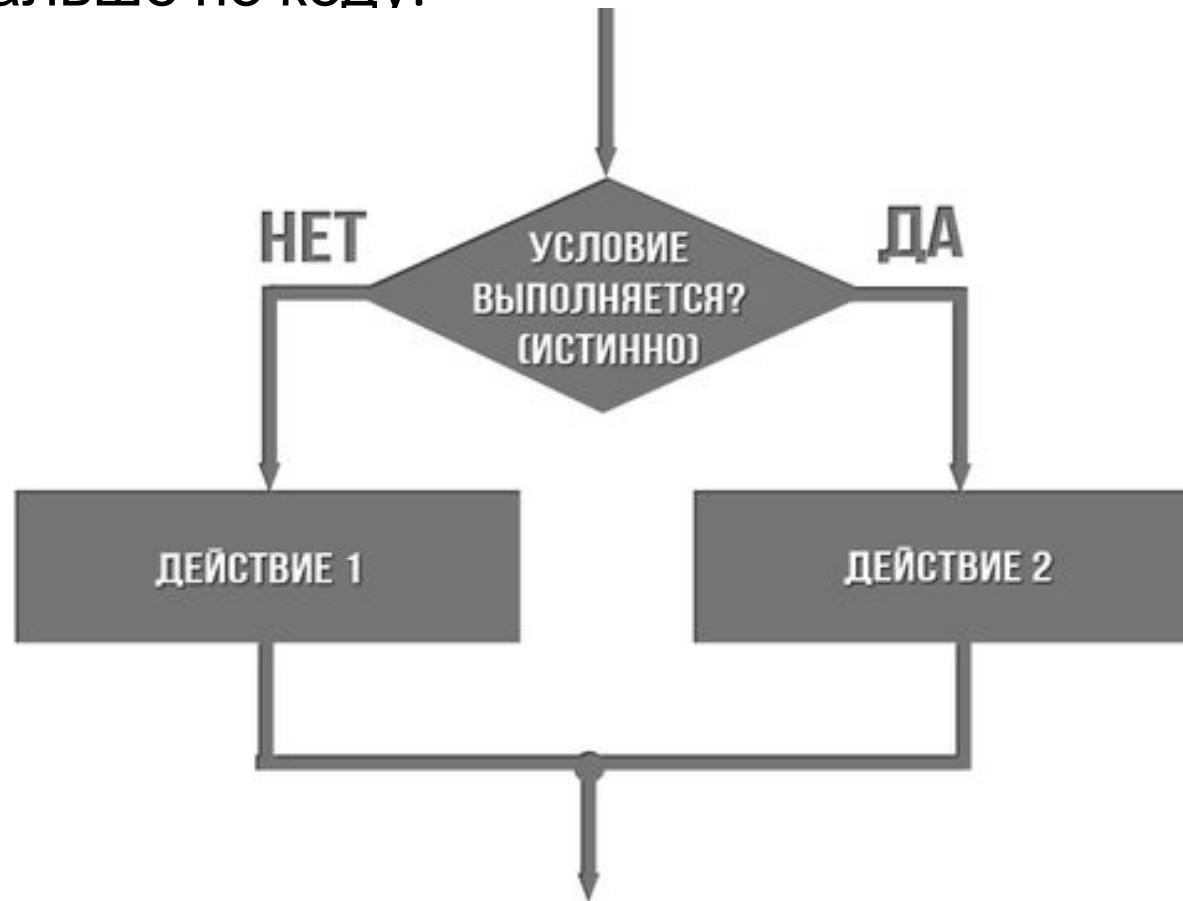


# Условный оператор

Основы программирования C++

- В языке C++ есть такое понятие, как **логическая величина**, которая принимает два значения: правда и ложь, **true** и **false**, 1 и 0.

Если условие верно, то есть истинно, то выполняется специально указанный для этого случая фрагмент кода. Если же условие не верно, то есть ложно, то выполняется либо другая специально указанная часть кода, либо не выполняется ничего и работа микроконтроллера продолжается дальше по коду.



Выражение `if` состоит из ключевого слова `if` (если), затем в круглых скобках пишется условие, а затем в фигурные скобки помещается код, который будет выполнен если условие истинно. Условие всегда булево (логическое) значение: может быть только `true` или `false`.

```
if (условие) {  
    // В этом блоке список команд, выполняющихся, если  
    условие истинно или имеет значение, отличное от 0  
}
```

**else** (иначе, в противном случае). Данный блок выполняется если условие всех предыдущих блоков ложно

```
if (условие) {  
// В этом блоке список команд, выполняющихся,  
если условие истинно или имеет значение,  
отличное от 0  
}  
else {  
// В этом блоке список команд, выполняющихся,  
если условие ложно или имеет значение, равное 0  
}
```

- Условие – это некоторое логическое выражение, возвращающее **истину (TRUE)** или **ложь (FALSE)**. В одном условии можно использовать несколько выражений, объединяя их специальными логическими операторами.

# Операторы сравнения

- $x == y$  (x равно y)
- $x != y$  (x не равно y)
- $x < y$  (x меньше y)
- $x > y$  (x больше y)
- $x <= y$  (x меньше или равно y)
- $x >= y$  (x больше или равно y)

# Логические операторы

- && – условие И
- || – условие ИЛИ
- ! – отрицание условия



- Порядок условий играет очень большую роль при оптимизации кода и попытке сделать его более быстрым в некоторых случаях. Суть очень проста: логические выражения/величины проверяются слева направо, и если хоть одно значение делает всё выражение неверным (ложью), дальнейшая проверка условий прекращается.

# Таблица истинности

<u>a</u>	<u>b</u>	<u>a &amp;&amp; b</u>	<u>a    b</u>	<u>!a</u>	<u>!b</u>
<u>false</u>	<u>false</u>	<u>false</u>	<u>false</u>	<u>true</u>	<u>true</u>
<u>false</u>	<u>true</u>	<u>false</u>	<u>true</u>	<u>true</u>	<u>false</u>
<u>true</u>	<u>false</u>	<u>false</u>	<u>true</u>	<u>false</u>	<u>true</u>
<u>true</u>	<u>true</u>	<u>true</u>	<u>true</u>	<u>false</u>	<u>false</u>

- Если условий много и они дискретны (выполняются для определенного значения, то для нас удобен оператор **switch** (переключатель). Он используется в ситуациях, когда известно какие значения может принимать условие.

Оператор выбора `switch` позволяет создать удобную конструкцию, разветвляющую действия в зависимости от значения одной переменной.

```
switch (значение) {  
  case 0:// выполнить, если значение == 0  
    break;  
  case 1:// выполнить, если значение == 1  
    break;  
  case 2:  
  case 3:  
  case 4:// выполнить, если значение == 2, 3 или 4  
    break;  
  default: // выполнить, если значение не совпадает ни с одним из case  
    break;  
}
```

# Почитать:

- <https://arduino-master.ru/program/arduino-if-else-uslovie/>
- <https://alexgyver.ru/lessons/conditions/>