

Лекция №6 по дисциплине «ТЕОРИЯ АЛГОРИТМОВ»

УСЛОВИЯ

Преподаватель: Золотоверх Д.О.

# ВИДЫ АЛГОРИТМОВ: ЛИНЕЙНЫЙ

Линейный алгоритм — это алгоритм, в котором действия выполняются однократно и строго последовательно.

Самый простой пример реализации линейного алгоритма — путь домой.

Словесный способ записи данного алгоритма:

- 1) выйти на остановку;
- 2) подождать нужный автобус;
- 3) сесть на нужный автобус;
- 4) оплатить проезд;
- 5) выйти на остановке;
- 6) дойти до дома.



## ВИДЫ АЛГОРИТМОВ: РАЗВЕТВЛЯЮЩИЙСЯ

Разветвляющийся алгоритм — это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Самый простой пример реализации разветвляющегося алгоритма — если на улице идет дождь, то необходимо взять зонт, иначе не брать зонт с собой.



#### КОНСТРУКЦИЯ IF-ELSE

За счёт if-else можно проверить одно или несколько условий и в случае их успешной проверки будет выполнен один, иначе другой.

#### Синтаксис:

```
if (someVariable > 50)
{
    // выполнять действия
```

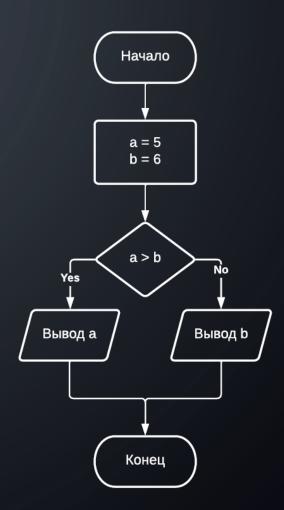
```
int a = 2, b = 10;
if (a == b) { // Если а будет равным b
    // Здесь код выполняется
    // Выполнение до след. '}'
} else if (a <= b) {</pre>
    // Если пред. условие не сработало
    // И а больше b, код выполняется
здесь
} else {
    // Этот код сработает
    // если другие условия
    // не будут выполнены
```

### КОНСТРУКЦИЯ IF-ELSE

С++ поддерживает обычные логические условия:

- меньше чем: а < б
- меньше или равно: a <= b
- больше, чем: а > б
- больше или равно: a >= b
- равно а == б
- не равно: а != 6

```
#include <iostream>
using namespace std;
int main() {
    int a = 5, b = 6;
    if (a > b) {
        printf("%d\n", a);
    } else {
        printf("%d\n", b);
    system("pause");
```

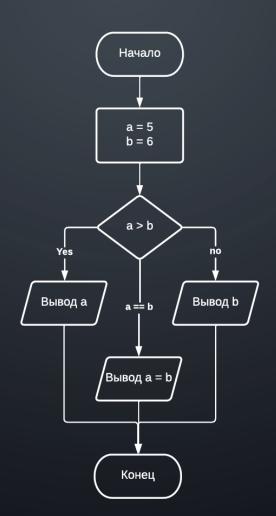


#### КОНСТРУКЦИЯ IF ELSE IF ELSE

Можно не использовать ключевое слово else.

Для конструкции условия его наличие не есть обязательным.

Также дополнительно можно дописать в неё сколькоугодно вложенных условий else if.



```
#include <iostream>
using namespace std;
int main() {
    int a = 5, b = 6;
    if (a > b) {
        printf("%d\n", a);
    } else if (a == b) {
        printf("%d = %d\n", a, b);
    } else {
        printf("%d\n", b);
    system("pause");
```

### КОНСТРУКЦИЯ SWITCH

Конструкция case обладает более удобным форматом для проверки множественных условий на совпадение значения.

В конструкцию записывается переменная, что проверяется, а также значения на которые происходит проверки.

```
switch (x) { // Проверяем переменную х
   case 1:
       // Если переменная будет равна 1, то здесь сработает
        // Может быть множество строк, а не только одна
        printf("Переменная равна 1\n");
        break; // Указываем конец для кода для этой проверки
    case 56: // Если переменная будет равна 56, то здесь
сработает код
        // Может быть множество строк, а не только одна
        printf("Переменная равна 56\n");
        break; // Указываем конец для кода для этой проверки
        // По аналогии таких проверок может быть множество
        // Также можно добавить проверку, которая сработает в
случае
       // если все остальные проверки не сработают
   default:
       printf("Переменная равна %d\n", x);
       break; // Можно и не ставить, так как это последние
условие
```

# СПАСИБО ЗА ВНИМАНИЕ!

