

Лекция №7
по дисциплине
«ТЕОРИЯ АЛГОРИТМОВ»

ЦИКЛЫ

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.

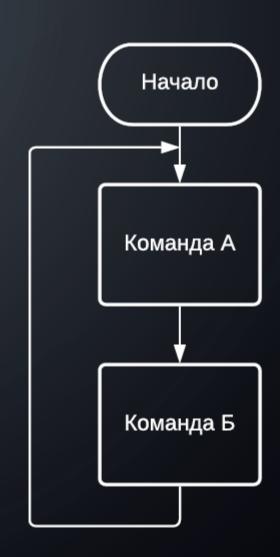
Самый простой пример реализации разветвляющегося алгоритма — если на улице идет дождь, то необходимо взять зонт, иначе не брать зонт с собой.



ВИДЫ АЛГОРИТМОВ: ЦИКЛИЧЕСКИЙ

Циклический алгоритм — это алгоритм, команды которого повторяются некое количество раз подряд.

Самый простой пример реализации циклического алгоритма — при чтении книги будут повторяться одни и те же действия: прочитать страницу, перелистнуть и т.д.



ЦИКЛ WHILE

- Цикл while будет повторятся непрерывно и бесконечно, пока выражение внутри скобок является истиной.
- Условием должно приравниваться к true или false.
- Бывает с предусловием и постусловием.

```
#include <iostream>
int main()
    int count = 0;
    while (count < 10)
        printf("%d\n", count);
        count++;
    printf("Done!");
```

ПРЕД- И ПОСТУСЛОВИЕ

Бывают конструкции циклов while как с пост-, так и с предусловием.

Не все языки программирования поддерживают постусловие (c++ поддерживает).

Все языки программирования поддерживают предусловие.



постусловие



ПРЕДУСВЛОВИЕ WHILE

Конструкция while с предусловием выполняет тело после проверки условия

Синтаксис:

while (условие) {

тело цикла;

```
#include <iostream>
int main()
    int count = 0;
    while (count < 10)
        printf("%d\n", count);
        count++;
    printf("Done!");
```

ПОСТУСВЛОВИЕ WHILE

Конструкция while с постусловием выполняет тело до проверки условия.

Может быть полезно, если необходимо выполнить цикл хотя бы один раз.

Позволяет обработать информацию до проверки условия.

Синтаксис:

```
do {
    тело цикла;
} while (условие) ;
```

```
#include <iostream>
int main()
    int count = 25;
        do {
        printf("%d\n", count);
         count++;
        } while (count < 10);</pre>
    printf("Done!");
```

КОНСТРУКЦИЯ FOR

Если мы, знаем точное количество итераций то можем использовать цикл for. Такая конструкция уже имеет встроенный счетчик.

```
Синтаксис:
for (
    инициализация счетчика;
    условие счетчика;
    инкремент счетчика
) {
    тело цикла;
}
```

```
#include <iostream>
int main()
    for (
        int i = 0;
        i < 10;
        i++
        printf("%d\n", i);
    printf("Done!");
```

ВЛОЖЕННЫЕ ЦИКЛЫ

Подобно другим конструкциям, можно разместить цикл внутри цикла.

С одной итерацией внешнего цикла выполняется десять итерации внутреннего цикла.

Вложенные циклы могут быть сложны для выполнения, так как требуют большое количество операций. Данный пример требует 100.

```
#include <iostream>
int main()
    for (int i = 0; i < 10; i++) {
        printf("%d ==> ", i);
        for (int j = 0; j < 10; j++) {
            printf("%d", j);
        printf("\n");
    printf("Done!");
```

СПАСИБО ЗА ВНИМАНИЕ!

