Лекция №5. Операторы цикла.

Цикл — многократно повторяющаяся часть алгоритма. Существуют различные виды циклов: с предусловием, с постусловием, с параметром, с выходом из середины.

Цикл с предусловием: while

```
while (УСЛОВИЕ) {
        ТЕЛО ЦИКЛА
}

/* Проверка условия → Выполнение тела цикла → если условие ≠ 0, то продолжается цикл (в противном случае выход из цикла) и т.д. */

int i = 1, sum = 0;
while (i ≤ 100) sum += i++;

```

Цикл с постпроверкой: do-while

```
do {
        ТЕЛО ЦИКЛА
} while (УСЛОВИЕ);

/* Выполнение тела цикла → Проверка условия → если условие ≠ 0, то продолжается цикл (в противном случае выход из цикла) и т.д. */
```

Оператор цикла for

Реализует наиболее общий способ организации циклов. При этом организуется цикл с предусловием и цикл с параметром.

```
for (START; STOP; STEP) {
        ТЕЛО ЦИКЛА
}

/* START – обычно используется дял установки начального положения переменной, управляющей циклом (инициализация переменной изначалального положения)
   STOP – условие выполнения тела цикла (когда оно остановится)
   STEP – определяет изменение переменных, управляющих циклом после каждого выполнения тела цикла

Вычисляется START → Вычисляется STOP → Если STOP ≠ 0, то выполняется тело цикла  → Вычисляется STEP → Переход к вычилению STOP (В противном случае управление передается на оператор следующий за циклом) */


int sum = 0;
for (int i = 0; i ≤ 100; i++) sum += i;
```

Задачка.
В чашке петри находится колония из B бактерий. В какой-то момент времени туда попадает V вирусов (V < B). Каждый час один вирус съедает одну бактерию, а затем число и тех и тех удваивается. Определить за какое время все бактерии будут съедены.

C *Решение*

```c
do {
        b -= v;
        b *= 2; // или лучше b <<= 1;
        v *= 2; // или лучше v <<= 1;
        t++;
} while (b > 0);

printf("t=%d\n", t);
return 0
```

Цикл с выходом из середины

Реализуется на основе любого из операторов цикла.

```c
while (1) {
        ТЕЛО ЦИКЛА
        if (1) break;
        ТЕЛО ЦИКЛА
}
```

Оператор **continue**, как и оператор **break**, прерывает выполнение тела цикла, но в отличие от **break**, **continue** — управление передается на управляющую часть цикла.