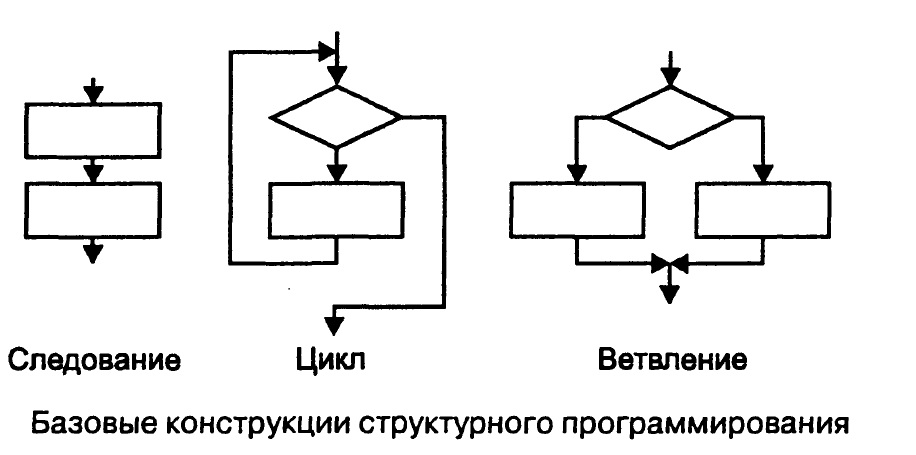
Ветвление и циклы

Согласно теореме структурного программирования, любой исполняемый алгоритм может быть преобразован к структурированному виду, то есть такому виду, когда ход его выполнения определяется только при помощи трёх структур управления: **последовательной**, **ветвлений** и **циклов**.



Последовательная структура была рассмотрена ранее. Рассмотрим оставшиеся структуры.

Ветвление

В случае ветвления происходит проверка условия, в соответствии с которой осуществляется переход для выполнения определенных команд.

В языке С ветвление можно организовать следующим образом:

|  |  |
| --- | --- |
| неполная развилка | if (Условие) {   БлокОпераций1; } |

В данном случае, если **Условие** истинно, то **БлокОпераций1** выполняется, если **Условие** ложно, то **БлокОпераций1** не выполняется.

Также, можно добавить еще один блок операций, который будет выполнен в том случае, если условие не выполнится.

|  |  |
| --- | --- |
| полная развилка | if (Условие) {   БлокОпераций1; } else {   БлокОпераций2; } |

Как и в других языках программирования, ветка else не является обязательной.

Для задания условия используются логические операторы:

* > больше
* < меньше
* >= больше или равно
* <= меньше или равно
* == равно
* != не равно
* ! отрицание выражения

Например,

#include <stdio.h>

int main() {

int num;

scanf("%d", &num);

printf("Вы ввели число %d\n", num);

if (num == 0)

printf("Это ноль\n");

else

printf("Это не ноль\n");

return 0;

}

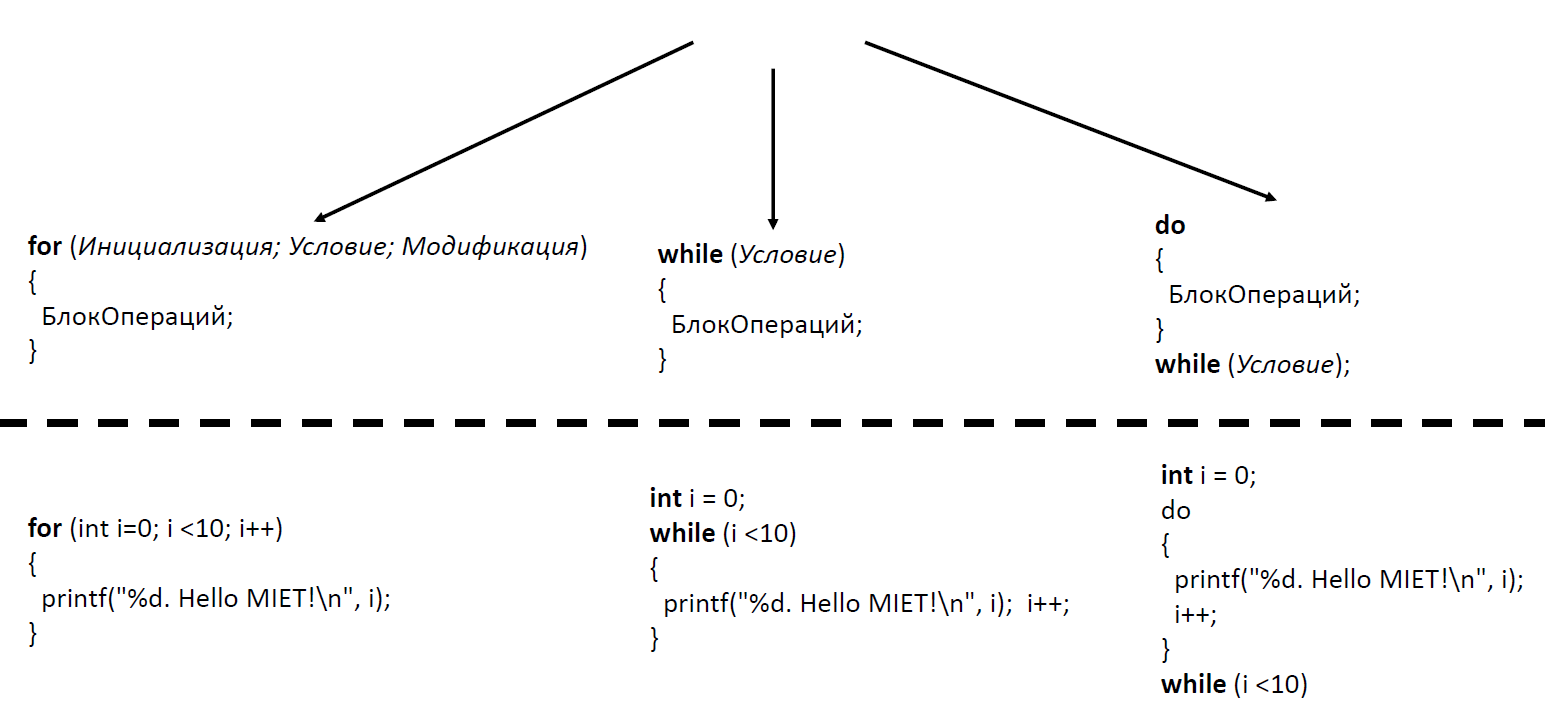
Циклы

Циклом называется блок кода, который для решения задачи требуется повторить несколько раз.

В языке Си следующие виды циклов:

* while — цикл с предусловием;
* do…while — цикл с постусловием;
* for — параметрический цикл (цикл с заданным числом повторений).

В общем виде их можно представить:



Пример:

#include <stdio.h>

int main()

{

for (int i = 0; i < 5; i++) {

printf("FOR: Hello, MIET\n");

}

return 0;

}

В данном примере происходит вывод строчки «FOR: Hello, MIET» 5 раз.

Пример 2:

#include <stdio.h>

const int param = 0;

int main()

{

int i = param;

while (i < 5)

{

printf("WHILE: Hello, MIET\n");

i++;

}

i = param;

do

{

printf("DO WHILE: Hello, MIET\n");

i++;

} while (i < 5);

return 0;

}

В данном примере дважды происходит вывод строчек на экран. Чтобы лучше понять, в чем разница между данными двумя циклами, измените значение *param* на 5.

Примеры с занятия:

Пример 1. В зависимости от введенного числа вывести на экран название дня недели.

#include <stdio.h>

int main()

{

int x;

scanf\_s("%d", &x);

if (x == 1)

printf("Monday\n");

if (x == 2)

printf("Tuesday\n");

if (x == 3)

printf("Wednesday\n");

if (x == 4)

printf("Thursday\n");

if (x == 5)

printf("Friday\n");

if (x == 6)

printf("Saturday\n");

if (x == 7)

printf("Sunday\n");

if (x < 1)

printf("<1\n");

if (x > 7)

printf(">7\n");

return 0;

}

Используем логические операторы

#include <stdio.h>

int main()

{

int x;

scanf\_s("%d", &x);

if (x == 1)

printf("Monday\n");

if (x == 2)

printf("Tuesday\n");

if (x == 3)

printf("Wednesday\n");

if (x == 4)

printf("Thursday\n");

if (x == 5)

printf("Friday\n");

if (x == 6)

printf("Saturday\n");

if (x == 7)

printf("Sunday\n");

if (x < 1 || x > 7)

printf("Error\n");

return 0;

}

Лабораторная работа №3:

1. Самостоятельно изучить главу 3.4 (Переключатель *switch*) книги «Язык программирования С», Б. Керниган, Д. Ритчи. Подумать, где в примере 2 возможно использовать оператор *switch*. Исправить программу, используя его.
2. Написать программу, вычисляющую функцию:
3. Написать программу, вычисляющую функцию