По одной из классификаций, имеющей значение в программировании, алгоритмы делятся на следующие:

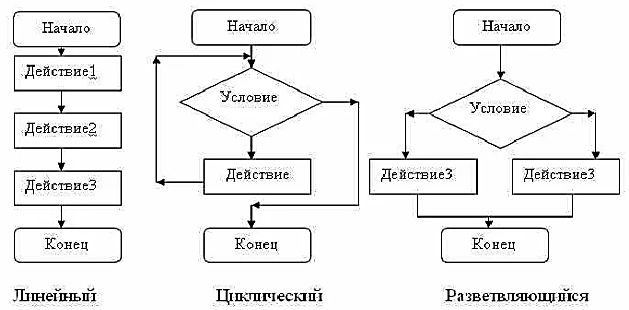
******

Рис 1. Виды алгоритмов

Рассмотрим более подробно циклический алгоритм.

Для реализации циклических алгоритмов в языках программирования реализованы специальные команды (операторы) цикла. Записываются они по-разному, но структура примерно одинакова:

ОператорЦикла (условие)

НачалоЦикла

Оператор

Оператор

…………

КонецЦикла

Команда цикла работает так: операторы цикла повторяются до тех пор, пока условие выполняется или не встретится команда выхода из цикла.

Существует два основных варианта циклов:

1. Количество повторений цикла известно заранее
2. Количество повторений заранее не известно

На самом деле вариантов больше, но все они в конечном итоге сводятся к этим двум.

В каких случаях применяются циклы? Обычно тогда, когда требуется перебор вариантов или данных. Например, при решении отгадывания задуманного числа, можно зациклить процедуру отгадывания. Но в этом случае надо предусмотреть выход из цикла.

В С++ обычно используются два варианта циклов:

**for** – цикл с неизвестным количеством повторений

**while** – цикл с неизвестным количеством повторений.

Рассмотрим **while.** Реализуем задачу отгадывания числа с циклом.

Структура цикла while в С++ следующая:

while (условие)

{

Оператор

Оператор

…………

}

Операторы выполняются до тех пор, пока условие выполняется. Например,

**#include "stdafx.h"**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int x, sum;**

**while (true)**

**{**

**cout << "chislo (1 : 9)?" << endl;**

**system("PAUSE");**

**cout << "+ 10" << endl;**

**system("PAUSE");**

**cout << "\* 2" << endl;**

**system("PAUSE");**

**cout << "- 5" << endl;**

**system("PAUSE");**

**cout << "+ chislo" << endl;**

**system("PAUSE");**

**cout << "read sum - ";**

**cin >> sum;**

**x = (sum - 15) / 3;**

**cout << "chislo = " << x << endl;**

**system("PAUSE");**

**}**

**}**

В данном варианте цикл будет выполнятся «вечно», потому что условие у нас всегда true. Как же сделать, что бы цикл заканчивался тогда, когда нам надо? Самый простой способ – это ввести логическую переменную, например, cont. Пока она равна истине, то цикл продолжается, а если ложь, то заканчивается.

**int main()**

**{**

**int x, sum;**

**bool cont=true;**

**while (cont)**

**{**

**cout << "chislo (1 : 9)?" << endl;**

**system("PAUSE");**

**cout << "+ 10" << endl;**

**system("PAUSE");**

**cout << "\* 2" << endl;**

**system("PAUSE");**

**cout << "- 5" << endl;**

**system("PAUSE");**

**cout << "+ chislo" << endl;**

**system("PAUSE");**

**cout << "read sum - ";**

**cin >> sum;**

**x = (sum - 15) / 3;**

**cout << "chislo = " << x << endl;**

**cout << "for continue press 1, for end press 0" << endl;**

**cin >> cont;**

**}**

**}**

В этом варианте, если на приглашение **"for continue press 1, for end press 0"** введена единица (true), то программа повторится, а если нуль, то закончит свою работу.

Разберем теперь цикл с известным числом повторений for. В С++ он оформляется следующим образом:

**for** (количество повторений)

{

Оператор

Оператор

…………

{

Пример,

for (int i=0; i<10; i++)

{

Оператор

Оператор

…………

}

Запись читается так: цикл повторяется от нуля до 9, с шагом 1. Т.е. переменная цикла меняется от нуля до девяти, с шагом 1. Столько же раз будет повторяться цикл, т.е. 10 раз.

i++ означает запись i=i+1, т.е. к содержимому i прибавляется единица и результат опять помещается в переменную i.

Для демонстрации работы цикла for реализуем классический алгоритм вычисления суммы целых чисел от 1 до 100.

**#include "stdafx.h"**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int sum=0;**

**for (int i=1; i<=100;i++)**

**{**

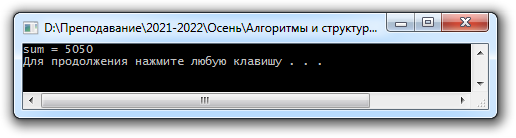
**sum = sum+i;**

**}**

**cout << "sum = " << sum << endl;**

**system("PAUSE");**

**}**



Можно сократить текст программы, т.к. тело цикла состоит из одного оператора и в этом случае скобки не обязательны:

**int main()**

**{**

**int sum=0;**

**for (int i=1; i<=100;i++) sum = sum+i;**

**cout << "sum = " << sum << endl;**

**system("PAUSE");**

**}**

**Рассмотрим еще несколько примеров с циклом while**

1. В цикле вводим любые целые числа. Если введенные числа от 5 до 20, то подсчитать их сумму, а остальные введенные числа не суммируются. Реализуем этот алгоритм.

**int main()**

**{**

**int i, s;**

**bool pr=true;**

**s = 0;**

**while (pr)**

**{**

**cout << "i=";**

**cin >> i;**

**if (i >5 && i<20) s=s+i;**

**cout << "For continue press 1, else 0 ";**

**cin >> pr;**

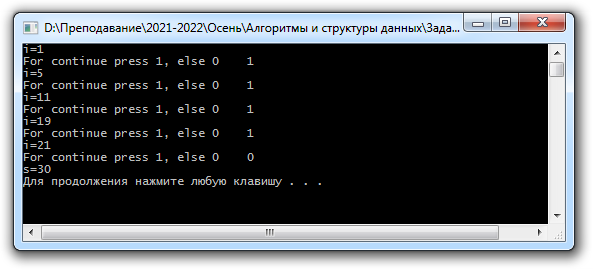
**}**

**cout << "s=" << s << endl;**

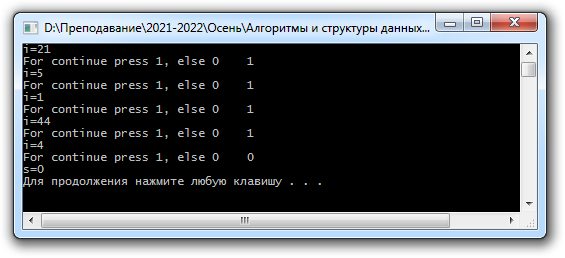
**system("PAUSE");**

**}**

На скриншоте введены числа 1, 5, 11, 19, 21, но в сумму попали лишь числа 11 и 19.



На следующем скриншоте введены числа 21, 5, 1, 44, 4, но в сумму не одно число не попало, поэтому сумма равна нулю.



1. В цикле вводим любые символы, пока не ввели символ #. Если среди введенных символов определился символ @, то на экран выдать сообщение «write @», иначе «not write @». Реализуем этот алгоритм.

**int main()**

**{**

**char sym;**

**bool pr=false;**

**while (!pr)**

**{**

**cout << "sym=";**

**cin >> sym;**

**if (sym == '@') pr = true;**

**else if (sym == '#') break;**

**}**

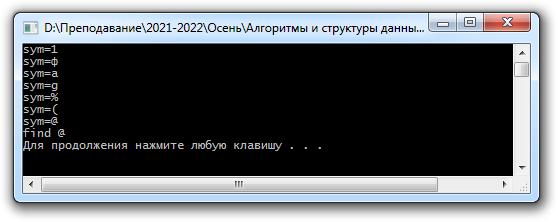
**if (pr) cout << "find @" << endl;**

**else cout << "not find @" << endl;**

**system("PAUSE");**

**}**

В первом варианте входных данных есть символ @



Во втором варианте вводили символы, пока был не введен символ #, после чего ввод прекращен и нужный символ не обнаружен:

