

Программирование (C++)

§ 17. Введение

§ 18. Линейные программы

§ 19. Ветвления

§ 20. Программирование
циклических алгоритмов

Программирование (C++)

§ 17. Введение

Что такое программирование?

Программирование — это создание программ для компьютеров. Этим занимаются **программисты**.

Чем занимаются **программисты**:

анализ задачи (выделение исходных данных, связей между ними, этапов решения задачи)

системные аналитики

разработка **алгоритмов**

алгоритмисты

написание и отладка **программ**

кодировщики

тестирование программ

тестировщики

написание **документации**

технические писатели

Направления в программировании

системный программист

операционные системы,
утилиты, драйверы

прикладной программист

прикладные программы, в
т.ч. для мобильных
устройств

веб-программист

веб-сайты

программист баз данных

системы управления
базами данных

Простейшая программа на C++

результат – целое
число (integer)

название программы
main – главный

```
int main()  
{  
  // это основная программа  
  /* здесь записывают  
     операторы */  
}
```

комментарии после
// не обрабатываются

комментарии внутри
/* */ не обрабатываются



Что делает эта программа?

Вывод на экран

ПОДКЛЮЧИТЬ
библиотеку
`iostream`

input-output streams –
ПОТОКИ ВВОДА И
ВЫВОДА

```
#include <iostream>
using namespace std;
int main() {
    cout << "Привет!"; // вывод текста
    cin.get(); // ждать Enter
}
```

ИСПОЛЬЗОВАТЬ
пространство имён
`std`

`cout` — ПОТОК ДЛЯ ВЫВОДА СИМВОЛОВ
(*character output stream*)

`cin` — ПОТОК ДЛЯ ВВОДА СИМВОЛОВ
(*character input stream*)

Вывод на экран

оператор
вывода

Оператор — это команда
языка программирования.


```
cout << "Привет! " ;
```


```
cout << "Привет! " ;  
cout << "Вася! " ;
```

или так:

```
cout << "Привет! " << "Вася! " ;
```

```
cout << "Привет, " << "Вася! " ;
```

 Что плохо?

 Символьные строки
записывают в кавычках!

Переход на новую строку

```
cout << "Привет, Вася!";  
cout << "Привет, Петя!";
```

ожидание:

```
Привет, Вася!  
Привет, Петя!
```

реальность:

```
Привет, Вася!Привет, Петя!
```

перейти
на новую
строку

Решение:

```
cout << "Привет, Вася!" << endl;  
cout << "Привет, Петя!" << endl;
```

end of line

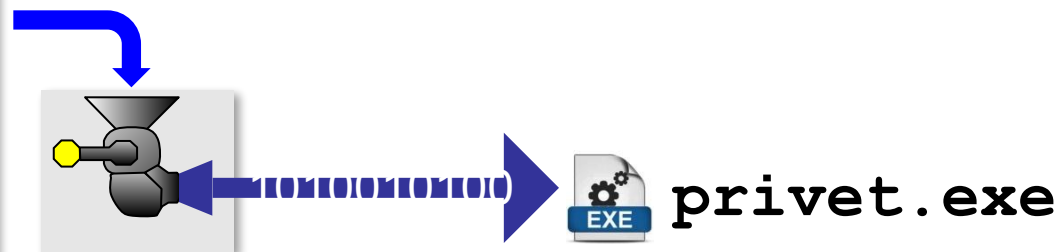
Системы программирования

Системы программирования — это средства для создания новых программ.

Транслятор — это программа, которая переводит тексты программ, написанных программистом, в машинные коды (команды процессора).

- **компилятор** — переводит всю программу в машинные коды, строит исполняемый файл (**.exe**)

```
#include <iostream>
using namespace std;
int main() {
    cout << "Привет!";
    cin.get();
}
```



- **интерпретатор** — сам выполняет программу по частям (по одному оператору).

Системы программирования

Отладчик — это программа для поиска ошибок в других программах.

- **пошаговый режим** — выполнение программы по шагам (по одному оператору)
- **просмотр значений переменных** во время выполнения программы
- **точки останова** — операторы в программе, перед выполнением которых нужно остановиться.

Среда программирования (IDE):

- редактор текста программ
- транслятор
- отладчик

Домашняя работа:

«В»: Напишите программу которая выведет на экран текст «лесенкой»

```
Вася
    пошел
        гулять
```

«С»: Напишите программу которая выведет на экран рисунок из букв

```
Ж
ЖЖЖ
ЖЖЖЖЖ
ЖЖЖЖЖЖЖ
НН НН
ZZZZZ
```

Программирование (C++)

§ 18. Линейные программы

Пример задачи

Задача. Ввести два числа и вычислить их сумму.

```
int main()  
{  
  // ввести два числа  
  // вычислить их сумму  
  // вывести сумму на экран  
}
```

 Выполнится?

Псевдокод – алгоритм на русском языке с элементами языка программирования.

 Компьютер не может исполнить псевдокод!

Зачем нужны переменные?

```
int main()  
{  
  // ввести два числа  
  // вычислить их сумму  
  // вывести сумму на экран  
}
```

Где запомнить?

Переменная — это величина, которая имеет имя, тип и значение. Значение переменной может изменяться во время выполнения программы.

```
int a, b, c;
```

объявление переменных



ячейки памяти

Имена переменных

Идентификатор — это имя программы или переменной.

```
int a, b, c;
```

заглавные и строчные
буквы **различаются**

МОЖНО использовать

- латинские буквы (A-Z, a-z)
- цифры



Имя не может начинаться с цифры!

- знак подчеркивания _

НЕЛЬЗЯ использовать ~~скобки, знаки ", &, |, *, +, =, !, ? и др.~~

Какие имена правильные?

AXby R&B 4Wheel Вася "PesBarbos"
TU154 [QuQu] _ABBA A+B

Типы переменных

- **int** – целые

```
int a = 1, b, c = 0;
```

начальные значения

- **float** – вещественные (могут иметь дробную часть)

```
float x = 1.234, y = 3.0, z = 0.576;
```



Целая и дробная части отделяются точкой!

- **double** – вещественное с двойной точностью
- **char** – один символ (в апострофах)

```
char c = 'ю';
```

- **string** – символьная строка (в кавычках)

```
string s = "молоко", q = "я";
```


Работа с переменными

Присваивание (запись значения)

```
a = 5;
```

оператор
присваивания

$a \leftarrow 5$

```
a = X;  
a = 18;
```

? Что будет храниться в a ?

Вывод на экран

```
cout << a;
```

? В чём разница?

```
c = 14;  
cout << c;
```

14

```
c = 14;  
cout << "c";
```

c

Работа с переменными

Изменение значения

```
i = i + 1;
```

увеличить на 1

$$i \leftarrow i + 1$$

```
a = 4;
```

```
b = 7;
```

```
a = a + 1;
```

```
b = b + 1;
```

```
a = a + b;
```

```
b = b + a;
```

```
a = a + 2;
```

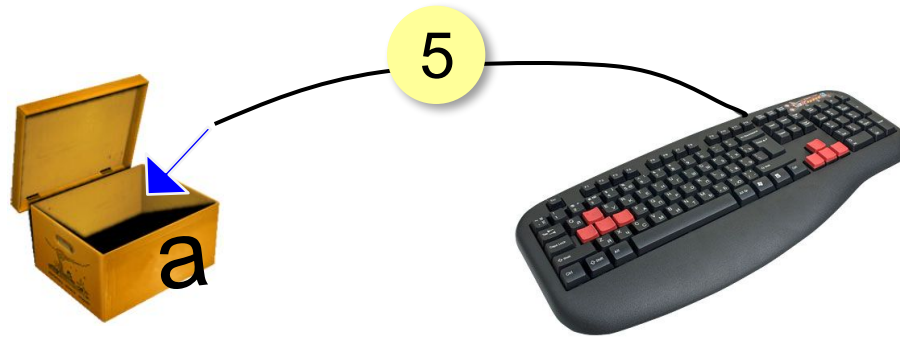
```
b = b + a;
```

a	b
4	
	7
5	
	8
13	
	21
15	
	36

Ввод с клавиатуры

Цель – изменить исходные данные, не меняя программу.

```
cin >> a;
```

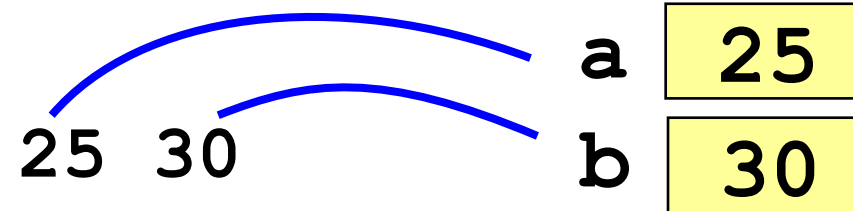


1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a**.

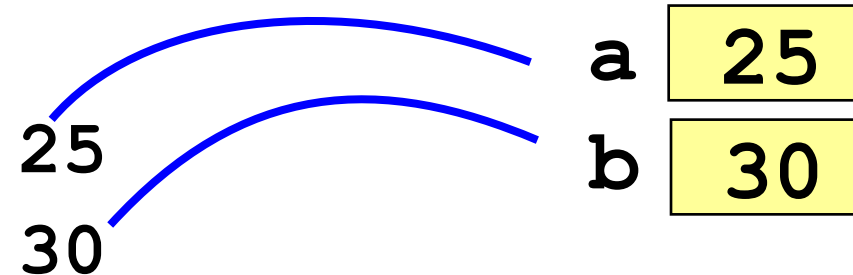
Ввод с клавиатуры

```
cin >> a >> b;
```

через пробел:



через *Enter*:



Программа сложения чисел

```
int main()
{
  int a, b, c;
  cin >> a >> b; // ввести два числа
  c = a + b; // вычислить их сумму
  cout << c; // вывести сумму на экран
}
```



Что плохо?

ожидание:

Введите два числа: 5 7
5+7=12

реальность:

5 7
12



Как улучшить диалог?

cout << данных с текстом

значение *a*

значение *b*

значение *c*

5+7=12

ТЕКСТ

```
cout << a;  
cout << "+" ;  
cout << b;  
cout << "=" ;  
cout << c;
```

```
cout << a << "+"  
    << b << "=" << c;
```

Программа сложения чисел

```
int main()  
{  
    int a, b, c;  
    cout << "Введите два числа: ";  
    cin >> a >> b;  
    c = a + b;  
    cout << a << "+" << b << "=" << c;  
}
```



Как переделать для 3-х чисел?

Задачи

«А»: Ввести три числа, найти их сумму.

Пример:

Введите три числа:

4

5

7

$$4+5+7=16$$

«В»: Ввести три числа, найти их сумму и произведение.

Пример:

Введите три числа:

4

5

7

$$4+5+7=16$$

$$4*5*7=140$$

Задачи

«С»: Ввести три числа, найти их сумму, произведение и среднее арифметическое.

Пример:

Введите три числа:

4

5

7

$$4+5+7=16$$

$$4*5*7=140$$

$$(4+5+7) / 3 = 5.333333$$

Сокращённая запись операций

Полная запись:

```
a = a + b;  
a = a + 1;  
a = a - b;  
a = a - 1;  
a = a * b;  
a = a / b;
```

Сокращённая запись:


```
a += b;  
a += 1;  
a -= b;  
a -= 1;  
a *= b;  
a /= b;
```


```
a++;
```

```
a--;
```

Особенность деления

```
int a = 7, b = 8;  
float x = a / b;
```

 Чему равен **x**?

 Результат деления целого числа на целое – это целое число (остаток отбрасывается)!

```
int a = 3, b = 4;  
float x;  
x = a / b;  
x = 10 / b;  
x = a / 2;  
x = 10. / b;  
x = a / 2. ;  
x = float(a) / b;
```

Частное и остаток при делении целых

/ – деление нацело (остаток отбрасывается)

% – остаток от деления


175 сек = 2 мин 55 сек



Как получить 2 и 55?

```
int t, m, s;  
t = 175;  
m = t / 60;  
s = t % 60;
```

Частное и остаток при делении целых

 Что получится?

```
n = 123  
d = n / 10;  
k = n % 10;
```

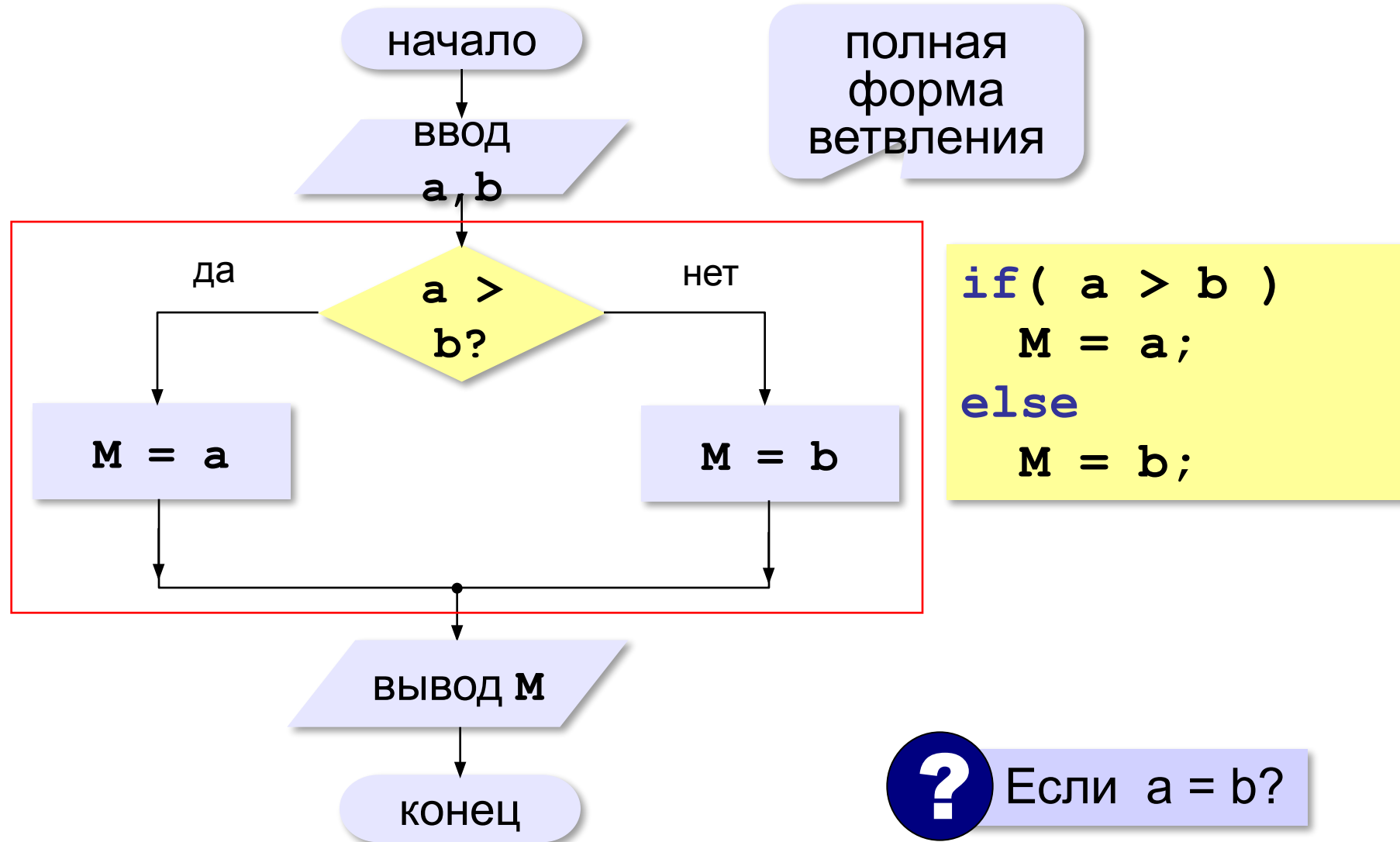
При делении на 10 нацело отбрасывается последняя цифра числа.

Остаток от деления на 10 – это последняя цифра числа.

Программирование (C++)

§ 19. Ветвления

Выбор наибольшего из двух чисел

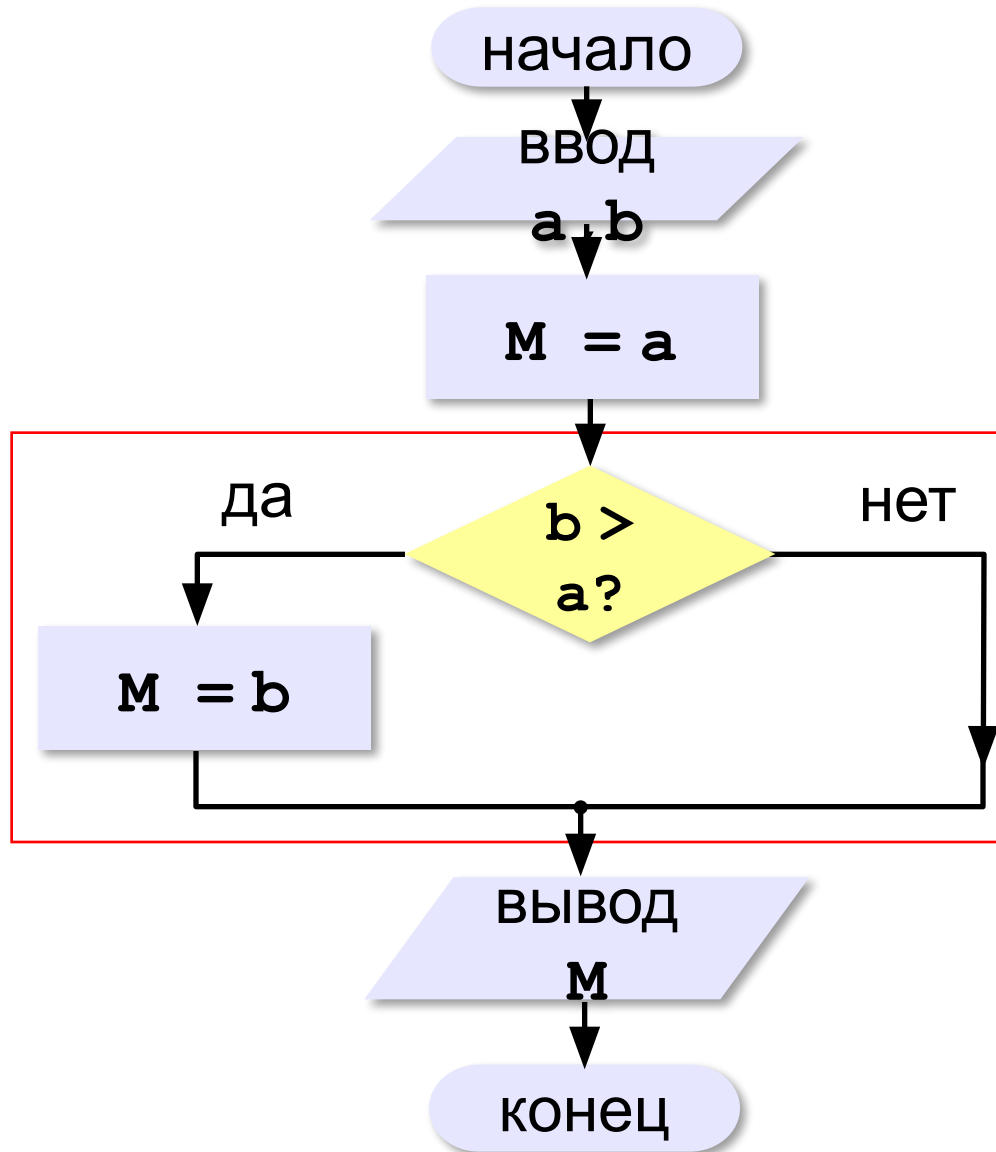


Вариант 1. Программа

```
int main()  
{  
  int a, b, M;  
  cout << "Введите два целых числа:";  
  cin >> a >> b;  
  if( a > b )  
    M = a;  
  else  
    M = b;  
  cout << "Наибольшее число " << M;  
}
```

полная форма
условного
оператора

Выбор наибольшего из двух чисел-2



неполная
форма
ветвления

Вариант 2. Программа

```
int main()  
{  
    int a, b, M;  
    cout << "Введите два целых числа: ";  
    cin >> a << b;  
    M = a;  
    if( b > a )  
        M = b;  
    cout << "Наибольшее число " << M;  
}
```

неполная
форма
условного
оператора

Примеры

Поиск минимального:

```
if( a < b )  
    M = a;  
if( b < a )  
    M = b;
```



Что плохо?



Когда работает неверно?

```
if( a < b )  
{  
    c = a;  
    a = b;  
    b = c;  
}
```



Что делает эта программа?

составной
оператор



Составной оператор (блок) в фигурных скобках!

В других языках программирования

Python:

```
if a < b:  
    c = a  
    a = b  
    b = c
```

Паскаль:

```
if a < b then begin  
    c = a;  
    a = b;  
    b = c;  
end;
```

Вложенные условные операторы

Задача. В переменной **a** записан возраст Антона, а в переменной **b** – возраст Бориса. Определить, кто из них старше.



Сколько вариантов ответа?

```
if( a == b )
    cout << "Одного возраста" ;
else
    if( a > b )
        cout << "Андрей старше" ;
    else
        cout << "Борис старше" ;
```

вложенный
условный
оператор

`else` относится к
ближайшему `if(`

Задачи

«А»: Ввести два целых числа, найти наибольшее и наименьшее из них.

Пример:

Введите два целых числа :

1 5

Наибольшее число 5

Наименьшее число 1

«В»: Ввести четыре целых числа, найти наибольшее из них.

Пример:

Введите четыре целых числа :

1 5 4 3

Наибольшее число 5

Задачи

«С»: Ввести последовательно возраст Антона, Бориса и Виктора. Определить, кто из них старше.

Пример:

Возраст Антона: 15

Возраст Бориса: 17

Возраст Виктора: 16

Ответ: Борис старше всех.

Пример:

Возраст Антона: 17

Возраст Бориса: 17

Возраст Виктора: 16

Ответ: Антон и Борис старше Виктора.

Программирование (C++)

Программирование циклических алгоритмов

Зачем нужен цикл?

Задача. Вывести 5 раз «Привет!».

```
cout << "Привет" ;  
cout << "Привет" ;  
cout << "Привет" ;  
cout << "Привет" ;  
cout << "Привет" ;
```



А если 5000?

Цикл «N раз»:

```
// сделай 5 раз  
cout << "Привет" ;
```



В C++ нет такого цикла! 😞

Как организовать цикл?

! Нужно запоминать, сколько раз цикл уже выполнен!

```
// счётчик = 0  
// пока счётчик < 5  
cout << "Привет";  
// счётчик = счётчик + 1
```

ещё не делали

сделали ещё раз

```
int count = 0;  
while( count < 5 ) {  
    cout << "Привет";  
    count++;  
}
```

составной оператор

Как организовать цикл?

Идея: запоминать, сколько шагов осталось.

```
int count = 5;  
while( count > 0 ) {  
    cout << "Привет";  
    count--;  
}
```

Цикл с предусловием

- условие проверяется при входе в цикл
- как только условие становится ложным, работа цикла заканчивается
- если условие ложно в самом начале, цикл не выполняется **ни разу**

```
while ( условие ) {  
    ...  
}
```

тело цикла

? Если условие никогда не станет ложно?

```
while ( 1 ) {  
    ...  
}
```

бесконечный цикл
(заикливание)

Сумма цифр числа

Задача. Вычислить сумму цифр введенного числа.

$$123 \rightarrow 1 + 2 + 3 = 6$$

Выделить последнюю цифру числа в переменной N :

$$d = N \% 10; \quad 123 \rightarrow 3$$

Отбросить последнюю цифру числа в переменной N :

$$N = N / 10; \quad 123 \rightarrow 12$$

Добавить к переменной sum значение переменной d :

$$sum = sum + d; \quad sum = 6 \rightarrow 6 + 4 = 10$$

или

$$d = 4$$

$$sum += d;$$

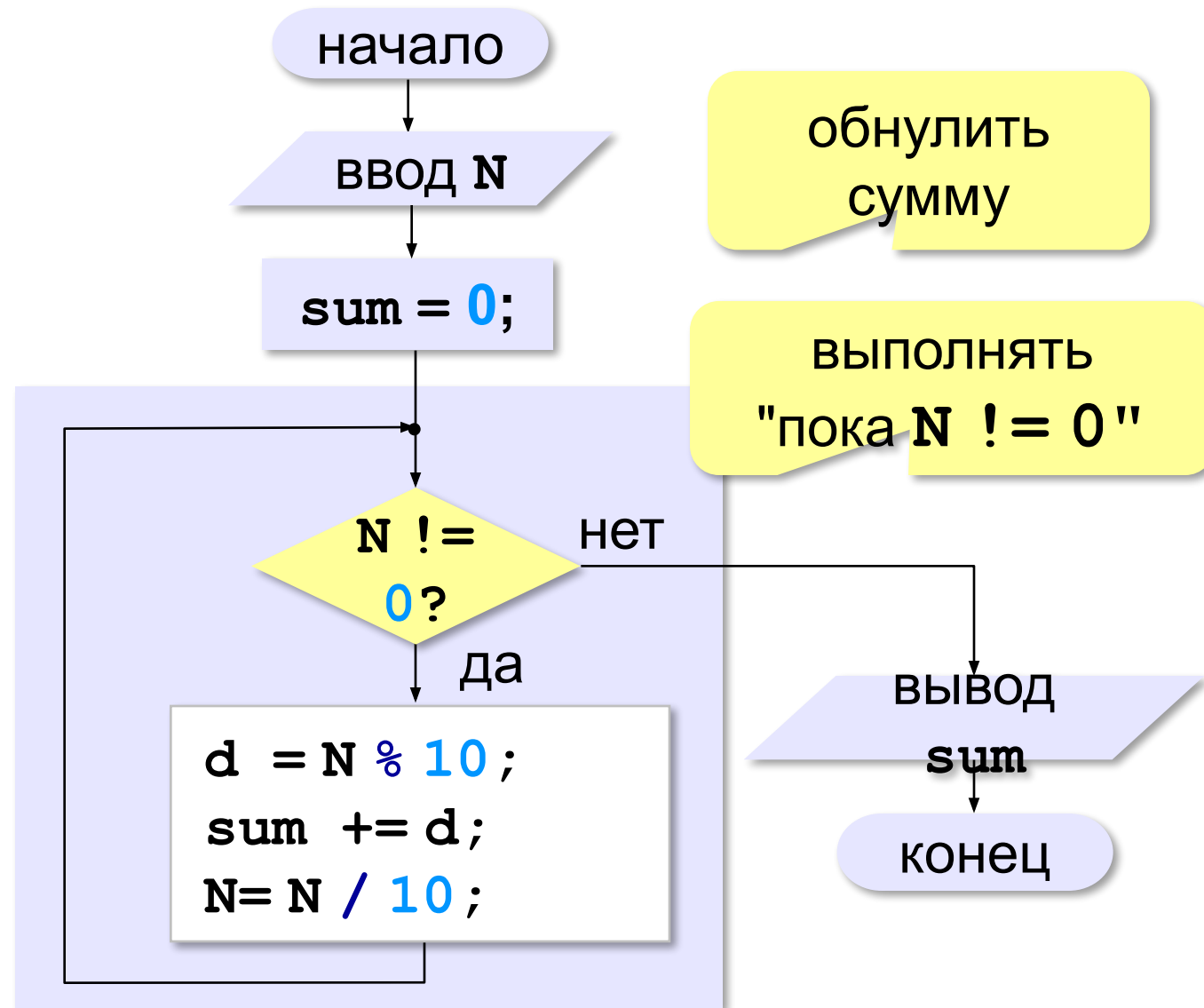
Сумма цифр числа

- выделяем последнюю цифру числа (%)
- увеличиваем сумму на значение цифры (`sum+=d;`)
- отсекаем последнюю цифру числа (/)

N	d	sum
123		0

начальные значения

Сумма цифр числа



Сумма цифр числа

```

int main() {
    int N, d, sum;
    cout << "Введите целое число";
    cin >> N;
    sum = 0;
    while( N != 0 ) {
        d = N % 10;
        sum += d;
        N = N / 10;
    }
    cout << "Сумма цифр числа " <<
        << " равна" << sum;
}

```

, N1

N1 = N;



Что плохо?

N1

Задачи

«А»: Напишите программу, которая получает с клавиатуры количество повторений и выводит столько же раз какое-нибудь сообщение.

Пример:

Сколько раз повторить? **3**

Привет!

Привет!

Привет!

«В»: Напишите программу, которая получает с клавиатуры натуральное число и определяет, сколько раз в его десятичной записи встречается цифра 1.

Пример:

Введите число? **311**

Единиц: **2**

Задачи

«С»: Напишите программу, которая получает с клавиатуры натуральное число и находит наибольшую цифру в его десятичной записи.

Пример:

Введите число: **311**

Наибольшая цифра: **3**

«D»: Напишите программу, которая получает с клавиатуры натуральное число и определяет, есть ли в его десятичной записи одинаковые цифры, стоящие рядом.

Пример:

Введите число: **553**

Ответ: **да.**

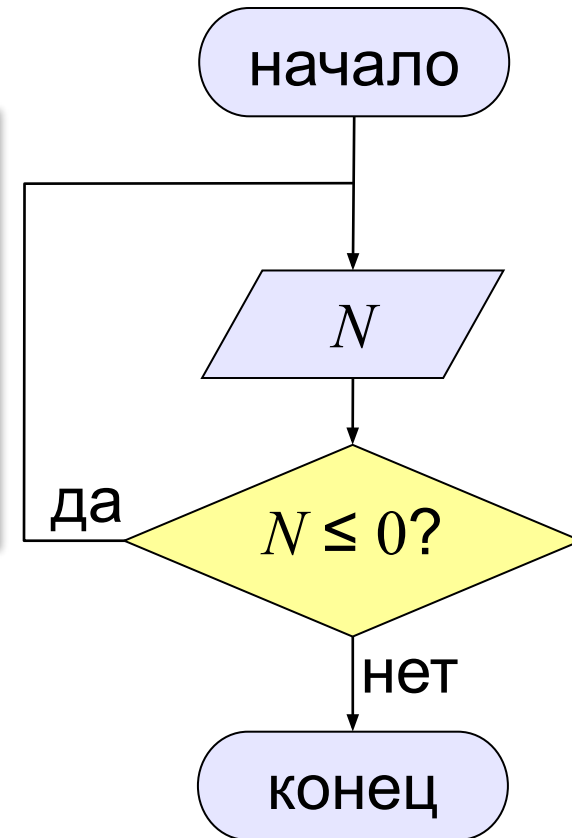
Введите число: **535**

Ответ: **нет.**

Цикл с постусловием

- условие проверяется **после** завершения очередного шага цикла
- цикл всегда выполняется хотя бы один раз
- как только условие становится **ИСТИННЫМ**, работа цикла заканчивается

```
do {  
    cout << "Введите N>0: ";  
    cin >> N;  
}  
while ( N <= 0 );
```



Задачи

- «А»: Напишите программу, которая предлагает ввести пароль и не переходит к выполнению основной части, пока не введён правильный пароль. Основная часть – вывод на экран «секретных сведений».
- «В»: Напишите программу, которая получает с клавиатуры натуральное число, которое больше 1, и определяет, простое оно или нет. Для этого нужно делить число на все натуральные числа, начиная с 2, пока не получится деление без остатка.
- «С»: Напишите программу, которая получает с клавиатуры два целых числа и вычисляет их произведение, используя только операции сложения.

Цикл по переменной

Задача. Вывести на экран степени числа 2 от 2^1 до 2^{10} .

```

k = 1;
N = 2;
while ( k <= 10 )
{
    cout << N << " ";
    N = N*2;
    k++;
}

```



Работа с `k` в трёх местах!

Идея:
собрать всё вместе.

```

N = 2;
for ( k=1; k<=10; k++ )
{
    cout << N << " ";
    N = N*2;
}

```

Цикл по переменной

Задача. Найти сумму чисел от 1 до 1000.

```
int sum = 0;  
for( int i=1; i<=1000; i++ )  
sum = sum +i;
```

переменная
нужна только
в цикле

Задача. Вывести квадраты чисел от 10 до 1 по убыванию.

```
for( int k=10; k>=1; k-- )  
cout << k*k << endl;
```

Цикл по переменной

Задача. Найти сумму чётных чисел от 2 до 1000.

```
int sum = 0;
for( int i=2; i<=1000; i++ )
    if( i % 2 == 0 )
        sum += i;
```

Задачи

«А»: Ипполит задумал трёхзначное число, которое при делении на 15 даёт в остатке 11, а при делении на 11 даёт в остатке 9. Напишите программу, которая находит все такие числа.

«В»: С клавиатуры вводится натуральное число N . Программа должна найти факториал этого числа (обозначается как $N!$) – произведение всех натуральных чисел от 1 до N . Например,

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120.$$

«С»: Натуральное число называется **числом Армстронга**, если сумма цифр числа, возведенных в N -ную степень (где N – количество цифр в числе) равна самому числу. Например, $153 = 1^3 + 5^3 + 3^3$. Найдите все трёхзначные Армстронга.