

Программирование на языке C++

Алгоритм и его свойства

Простейшие программы

Вычисления

Программирование на языке C++

§ 54. Алгоритм и его свойства

Что такое алгоритм?

Алгоритм — это точное описание порядка действий, которые должен выполнить исполнитель для решения задачи за конечное время.

Исполнитель — это устройство или одушевленное существо (человек), способное понять и выполнить команды, составляющие алгоритм.

Формальные исполнители: не понимают (и не могут понять) смысл команд.



Мухаммед ал-Хорезми
(ок. 783—ок. 850 гг.)

Свойства алгоритма

Дискретность — алгоритм состоит из отдельных команд, каждая из которых выполняется за конечное время.

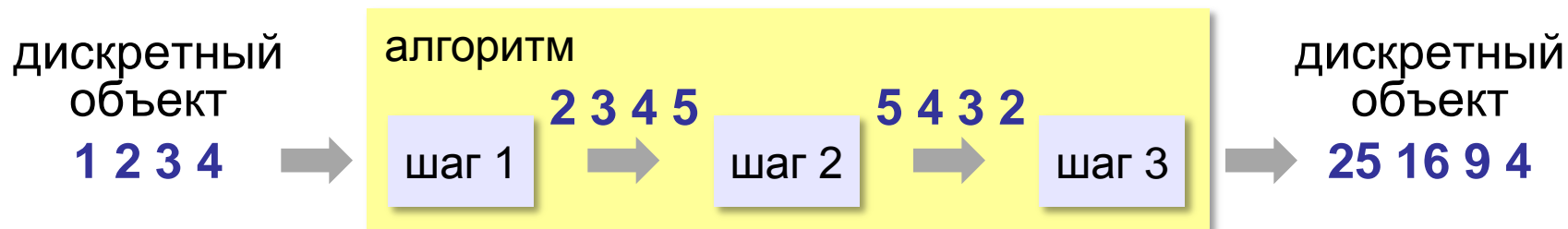
Детерминированность (определённость) — при каждом запуске алгоритма с одними и теми же исходными данными получается один и тот же результат.

Понятность — алгоритм содержит только команды, входящие в **систему команд исполнителя**.

Конечность (результативность) — для корректного набора данных алгоритм должен завершаться через конечное время.

Корректность — для допустимых исходных данных алгоритм должен приводить к правильному результату.

Как работает алгоритм?



- получает на вход дискретный объект
- в результате строит другой дискретный объект (или выдаёт сообщение об ошибке)
- обрабатывает объект по шагам
- на каждом шаге получается новый дискретный объект

Способы записи алгоритмов

- **естественный язык**

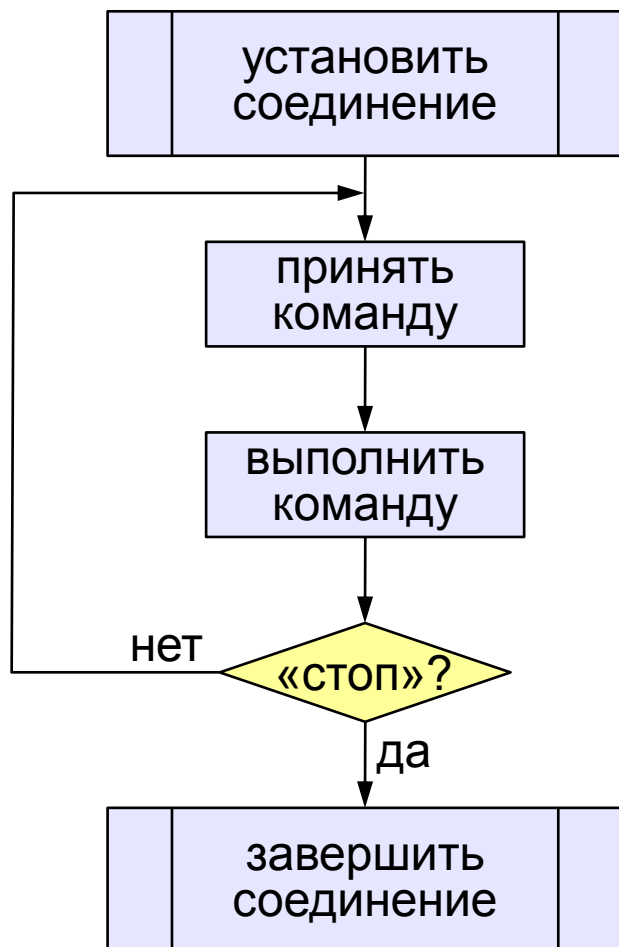
```
установить соединение  
пока не принята команда «стоп»  
    принять команду  
    выполнить команду  
завершить сеанс связи
```

- **псевдокод**

```
установить соединение  
начало цикла  
    принять команду  
    выполнить команду  
конец цикла при команда = 'stop'  
завершить сеанс связи
```

Способы записи алгоритмов

- блок-схема



- программа

```
установитьСоединение
начало цикла
cmd= получитьКоманду
выполнитьКоманду (cmd)
конец при cmd = 'stop'
закретьСоединение
```

Программирование на языке C++

§ 55. Простейшие программы

Простейшая программа

это основная программа

```
main ()
```

```
{
```

```
// это основная программа
```

```
/* здесь записывают
```

```
операторы */
```

```
}
```

комментарии после //
не обрабатываются

это тоже комментарий



Что делает эта программа?

Вывод на экран

```
main ()
```

```
{
```

```
    cout << "2+";
```

```
    cout << "2=?\n";
```

```
    cout << "Ответ: 4";
```

```
}
```

character output – **ВЫХОДНОЙ ПОТОК** [СИМВОЛОВ] на консоль

"\n" – новая строка

Протокол:

2+

Ответ: 4

Вывести на экран текст, при $a=5$, $b=7$ $5+7=?$

```
cout<<a<<"+"<<b<<"=? "
```

Вывести на экран текст, при $a=5$, $b=7$, дающий
ответ вместо «?» $5+7=?$

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

Справка:

- ✓ **cout** (си аут) – команда вывода
- ✓ в кавычках (" ") пишется текст, который будет выведен в консоли "текст",
- ✓ без кавычек будет выведено значение переменной, либо результат выполнения операций над переменными
- ✓ Текст и переменные отделяются <<

Подключение библиотечных функций

```
#include <iostream>
using namespace std;
main ()
{
    cout << "2+";
    cout << "2=?\n";
    cout << "Ответ: 4";
    cin.get ();
}
```

стандартные потоки
ввода и вывода

стандартное
пространство имен

ждать нажатия любой
клавиши

character input – **ВЫХОДНОЙ
ПОТОК** [СИМВОЛОВ] С КОНСОЛИ

Если не подключить пространство имён...

```
#include <iostream>
main()
{
    std::cout << "2+";
    std::cout << "2=?\n";
    std::cout << "Ответ: 4";
    std::cin.get();
}
```

пространство имен std

Вывод в поток

```
cout << "2+" << "2=?" << "\n"  
      << "Ответ: 4";
```

```
cout << "2+" << "2=?" << endl  
      << "Ответ: 4";
```

end of line – конец
строки

Задания

«В»: Вывести на экран текст «лесенкой»

Вася

пошел

гулять

«С»: Вывести на экран рисунок из букв

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

Сложение чисел

Задача. Ввести с клавиатуры два числа и найти их сумму.

Протокол:

Введите два целых числа

компьютер

25 30

пользователь

25+30=55

компьютер считает сам!

?

1. Как ввести числа в память?
2. Где хранить введенные числа?
3. Как вычислить?
4. Как вывести результат?

Сумма: псевдокод

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

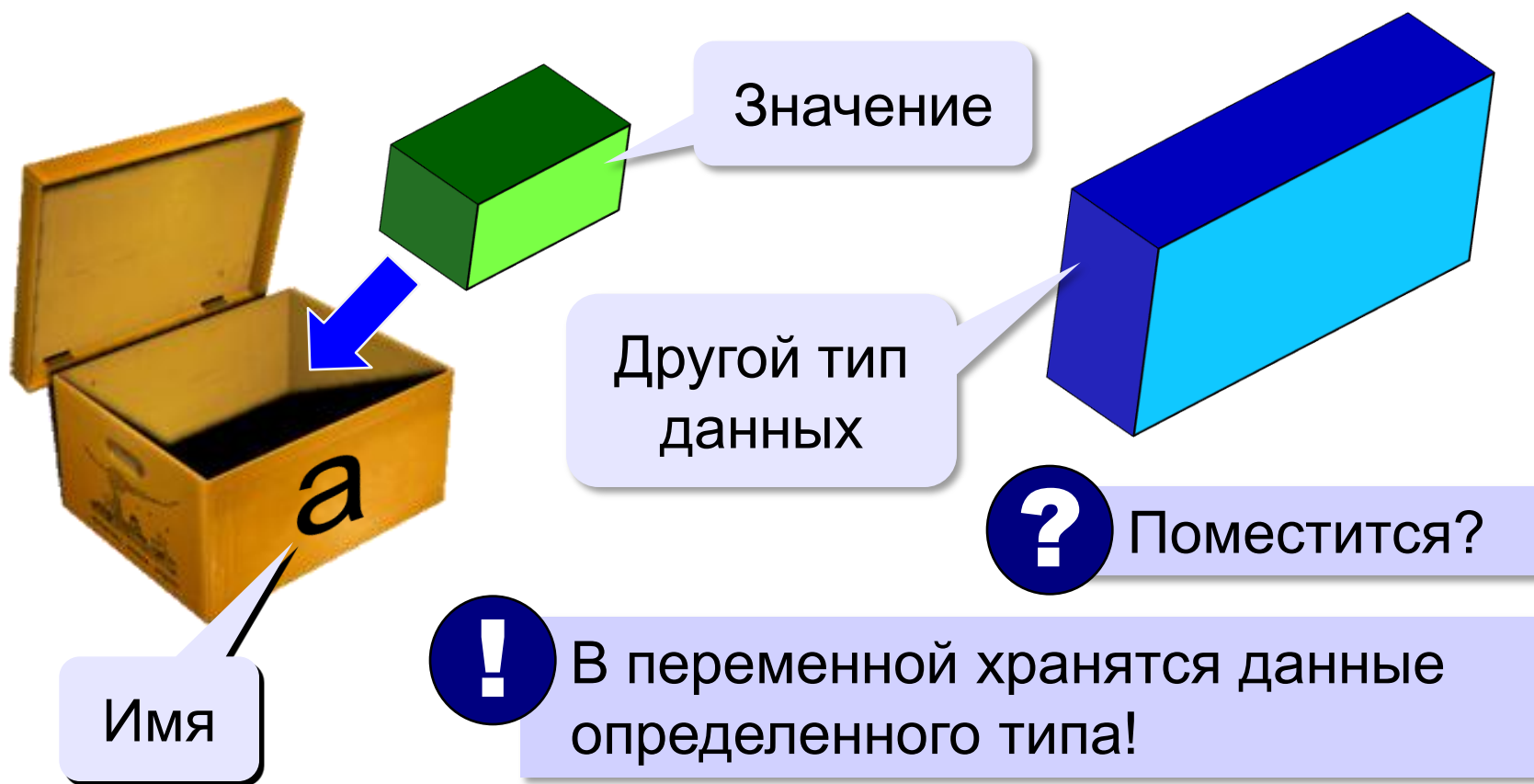
Псевдокод – алгоритм на русском языке с элементами языка программирования.



Компьютер не может исполнить псевдокод!

Переменные

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Имена переменных

МОЖНО использовать

- латинские буквы (A-Z, a-z)

заглавные и строчные буквы **различаются**

- цифры

имя не может начинаться с цифры

- знак подчеркивания _

НЕЛЬЗЯ использовать

- ~~русские буквы~~
- ~~скобки~~
- ~~знаки +, =, !, ? и др.~~

Какие имена правильные?

AXby R&B 4Wheel Вася "PesBarbos"

TU154 [QuQu] _ABBA A+B

Объявление переменных

Типы переменных:

- `int` // целая
- `float` // вещественная
- и другие...

Объявление переменных:

выделение
места в памяти

тип – целые

список имен
переменных

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

Тип переменной

- область допустимых значений
- допустимые операции
- объём памяти
- формат хранения данных
- для предотвращения случайных ошибок

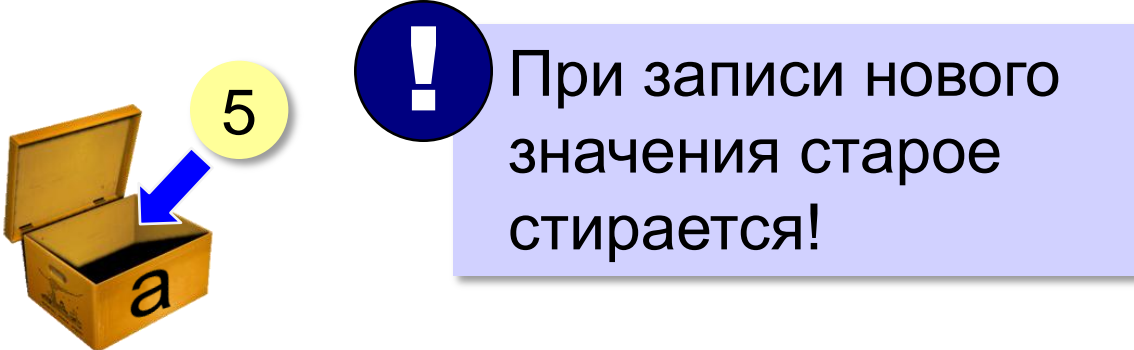
Начальные значения:

```
int a, b = 1, c = 55;
```

Как записать значение в переменную?

оператор присваивания

```
a = 5;
```



При записи нового значения старое стирается!

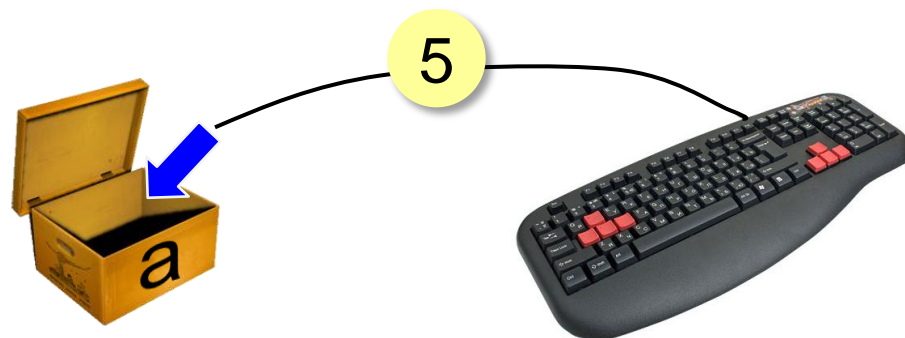
Оператор – это команда языка программирования (инструкция).

Оператор присваивания – это команда для записи нового значения в переменную.

Ввод значения с клавиатуры

ввести значение `a` из
входного потока

```
cin >> a;
```

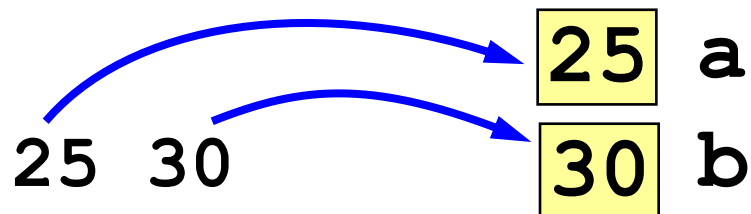


1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную `a`.

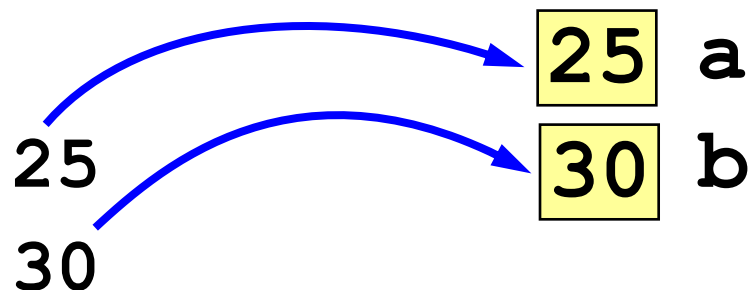
Ввод значений двух переменных

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

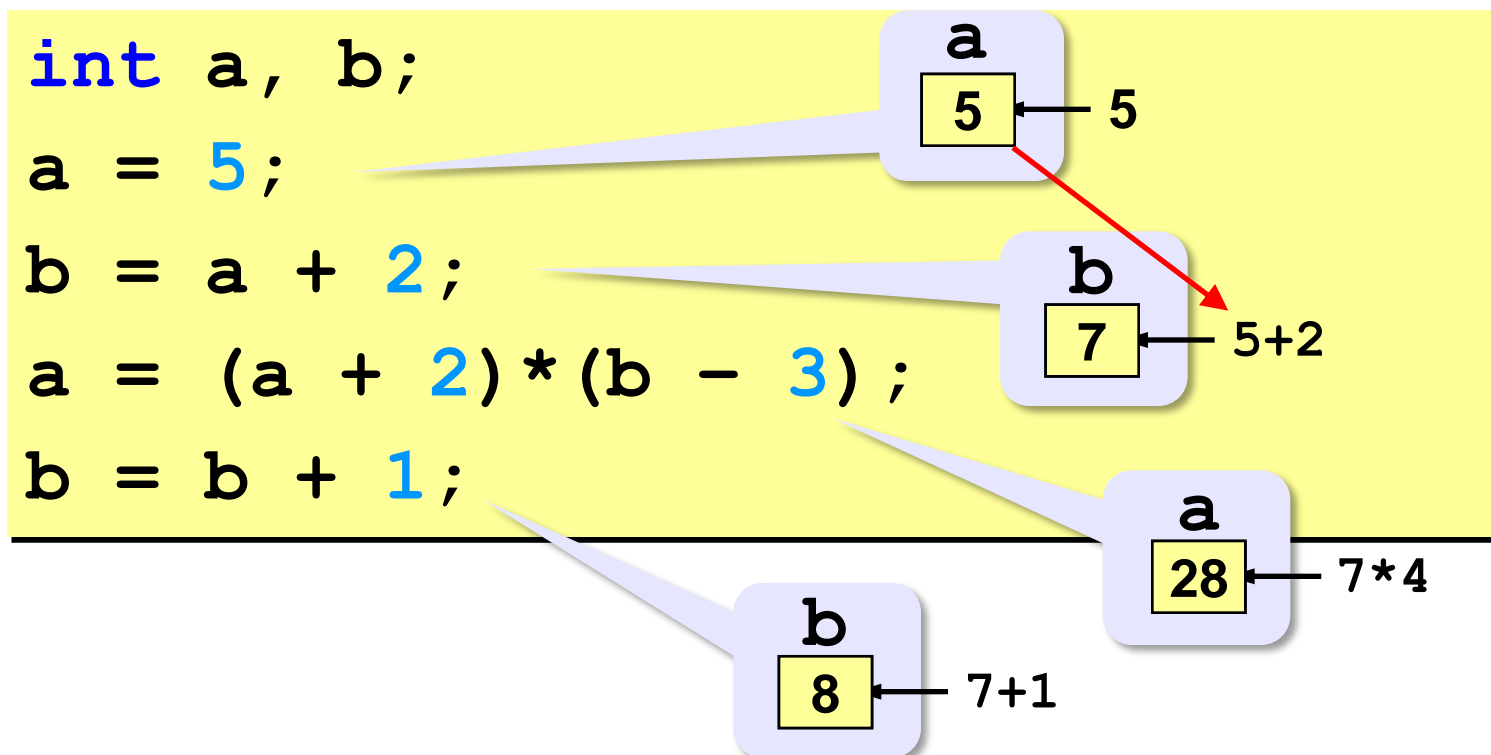
через пробел:



через *Enter*:



Изменение значений переменной



Вывод данных

```
cout << a;
```

// вывод значения
// переменной a

```
cout << a << endl;
```

// ... и переход
// на новую строку

```
cout << "Привет!";
```

// вывод текста

```
cout << "Ответ: " << c;
```

// вывод текста и значения переменной c

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

2+3=5

Сложение чисел: простое решение

```
#include <iostream>
using namespace std;
main()
{
    int a, b, c;
    cin >> a >> b;
    c = a + b;
    cout << c;
    cin.get(); cin.get();
}
```

ждём нажатия
на клавишу

читаем остатки
входного потока
после ввода

?

Что плохо?

Сложение чисел: полное решение

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

подсказка

Протокол:

КОМПЬЮТЕР

Введите два целых числа

25 30

ПОЛЬЗОВАТЕЛЬ

25+30=55

Снова про оператор вывода

Вычисление выражений:

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

Форматный вывод:

```
#include <iomanip>  
...  
a = 123;  
cout << setw(5) << a;
```

манипуляторы для управления потоками

123

5 знаков

set width – установить ширину поля

Программирование на языке C++

§ 56. Вычисления

Типы данных

- `int` // целое
- `long int` // длинное целое
- `float` // вещественное
- `double` // веществ. двойной точности
- `bool` // логические значения
- `char` // СИМВОЛ

Арифметическое выражения

3 1 2 4 5 6
`a = (c + b*5*3 - 1) / 2 * d;`

Приоритет (*старшинство*):

- 1) скобки
- 2) умножение и деление
- 3) сложение и вычитание

$$a = \frac{c + b \cdot 5 \cdot 3 - 1}{2} \cdot d$$

Деление

Результат деления целого на целое – **целое** число (остаток отбрасывается):

```
int a = 3, b = 4;
```

```
float x;
```

```
x = 3 / 4;
```

```
x = 3. / 4;
```

```
x = 3 / 4.;
```

```
x = a / 4;
```

```
x = a / 4.;
```

```
x = a / b;
```

```
x = float(a) / 4;
```

```
x = a / float(b);
```



Что запишется в **x**?

Остаток от деления

`%` – остаток от деления

```
int a, b, d;  
d = 85;  
b = d / 10;  
a = d % 10;  
d = a % b;  
d = b % a;
```

Для отрицательных чисел:

```
int a = -7;  
b = a / 2;  
d = a % 2;
```



В математике не так!

остаток ≥ 0

$$-7 = (-4) * 2 + 1$$

Сокращенная запись операций

```
int a, b;
```

```
...
```

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

```
a --; // a = a - 1;
```

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

```
a -= b; // a = a - b;
```

```
a *= b; // a = a * b;
```

```
a /= b; // a = a / b;
```

```
a %= b; // a = a % b;
```

Вещественные числа



Целая и дробная части числа разделяются точкой!

Форматы вывода:

```
float x = 123.456;  
cout.width(10);  
cout.precision(5);  
cout << x << endl;
```

5 значащих цифр

123.46

```
cout.width(10);  
cout.precision(2);  
cout << x << endl;
```

всего 10 знаков

1.2e+002

$1,2 \cdot 10^2$

Вещественные числа

Формат с фиксированной точкой:

```
#include <iomanip>
```

```
...
```

```
float x = 123.4567890123;
```

```
cout << fixed << setprecision(3)  
      << x;
```

в дробной части

фиксированный

123.456

Вещественные числа

Экспоненциальный (научный) формат:

```
float x;  
x = 1./30000;  
cout << x;  
x = 12345678.;  
cout << x;
```

$3,33333 \cdot 10^{-5}$

3.33333e-005

1.23457e+007

```
float x = 123.456;  
cout.width(10);  
cout.precision(2);  
cout << scientific << x;
```

$1,23457 \cdot 10^7$

в дробной части

1.23e+002

научный

Стандартные функции

```
#include <cmath>
```

подключить
математическую
библиотеку

- `abs (x)` — модуль целого числа
- `fabs (x)` — модуль вещественного числа
- `sqrt (x)` — квадратный корень
- `sin (x)` — синус угла, заданного **в радианах**
- `cos (x)` — косинус угла, заданного **в радианах**
- `exp (x)` — экспонента e^x
- `ln (x)` — натуральный логарифм
- `pow (x, y)` — x^y : возведение числа x в степень y
- `floor (x)` — округление «вниз»
- `ceil (x)` — округление «вверх»

```
float x;  
x = floor(1.6); // 1  
x = ceil(1.6); // 2
```

```
x = floor(-1.6); // -2  
x = ceil(-1.6); // -1
```



Источники иллюстраций

1. old-moneta.ru
2. www.random.org
3. www.allruletka.ru
4. www.lotterypros.com
5. logos.cs.uic.edu
6. ru.wikipedia.org
7. иллюстрации художников издательства «Бином»
8. авторские материалы