



# Модуль: Библиотеки C++.

Темы:

Действительные числа. Полезные функции библиотеки `cmath`. Вывод действительных чисел. Функции преобразования типов данных



# Что такое действительные числа в C++?

---

В C++ действительные числа — это числовые значения, которые могут иметь дробную часть.

В C++ **действительные числа** представлены с помощью **вещественных типов данных**:

**float** — одинарная точность (4 байта)

**double** — двойная точность (8 байт)

**long double** — расширенная точность

```
float a = 3.14f;  
double b = 2.71828;  
long double c = 1.6180339887L;
```

# <cmath>

Назначение:

**Математические функции:** корень, синус, степень и т.д.

Почему использовать:

- Стандартные и точные **функции**
  - Часто нужны при работе с **графикой, физикой** и т.д.
- 

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double x = 9.0;

    cout << "sqrt(x) = " << sqrt(x) << endl;    // Корень
    cout << "pow(x, 2) = " << pow(x, 2) << endl; // Степень
    cout << "sin(x) = " << sin(x) << endl;      // Синус
    cout << "log(x) = " << log(x) << endl;      // Натуральный логарифм
    cout << "abs(-x) = " << abs(-x) << endl;    // Модуль
}
```

Интересные факты:

- В `cmath` также есть `fmod(x, y)` — остаток от деления для **вещественных** чисел.
- Некоторые функции, например `sqrt()`, возвращают **NaN** (не число), если передать **отрицательное значение**.

# Полезные функции библиотеки <cmath> - округление

---

```
// ceil(x) – округление вверх
// floor(x) – округление вниз
double x = 3.7;
cout << ceil(x); // 4
cout << floor(x); // 3

// round(x) – округление до ближайшего целого
double x = 2.5;
cout << round(x); // 3
```

Интересные факты:  
**Ошибки округления** — одна из главных причин ошибок в числовых вычислениях.

# Вывод действительных чисел

---

Для форматирования **вывода**  
используют **iomanip**:

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    double pi = 3.1415926535;

    cout << fixed << setprecision(2) << pi << endl; // 3.14
    cout << scientific << pi << endl;                // 3.141593e+00
}
```

Мини-задание:

Попробуйте вывести  
число с 5, 10 и 15  
знаками после запятой!

# Преобразование типов данных

C++ позволяет преобразовывать между типами:

*Преобразование типов данных* (или type casting) — это **процесс преобразования** значения из одного **типа данных** в другой.

```
// Неявное преобразование:  
double d = 5; // int → double  
  
// Явное (cast):  
int a = 7;  
double b = (double)a / 2; // 3.5  
  
// C++-стиль:  
double b = static_cast<double>(a) / 2;
```

Мини-задания:

- Напишите программу, которая запрашивает радиус круга и выводит его площадь с точностью до 3 знаков.
- Используйте pow() и sqrt() для расчёта длины гипотенузы по двум катетам.