Фиг. 2.4 Изпълнение на C++ програма

**изходен**

**код**

**обектен**

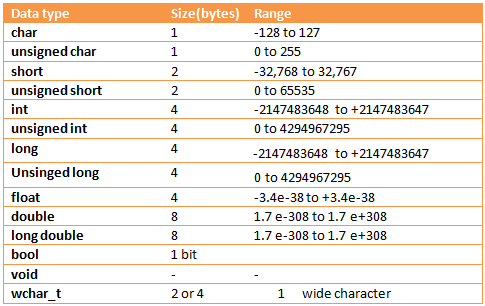
**код**

библиотеки

**изпълнима**

**програма**

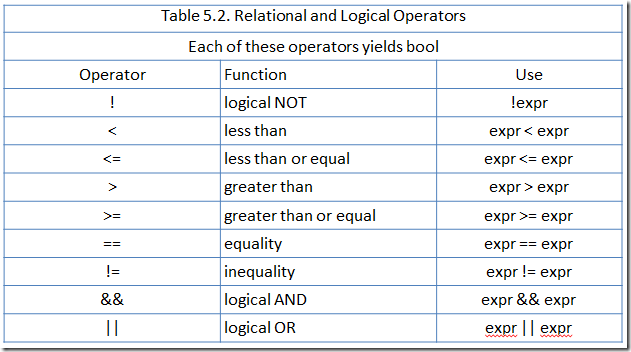
Данни – типове, размер, диапазон от стойности



Вградени функции в библиотеката cmath

|  |  |
| --- | --- |
| **Функция** | **Намира** |
| sin(x) | Синус, sin x, x е в радиани | |
| Cos(x) | косинус, cos x, x е в радиани | |
| tan(x) | тангенс, tg x, x е в радиани | |
| аsin(x) | аркуссинус, arcsin x ∈ [-π/2, π/2], x ∈ [-1, 1] | |
| аcos(x) | аркускосинус, arccos x ∈ [0, π], x ∈ [-1, 1] | |
| аtan(x) | аркустангенс, arctg x ∈ (-π/2, π/2) | |
| exp(x) | експонента, ex | |
| log(x) | натурален логаритъм, ln x, x > 0 | |
| log10(x) | десетичен логаритъм, lg x, x > 0 | |
| sinh(x) | хиперболичен синус, sh x | |
| cosh(x) | хиперболичен косинус, ch x | |
| tanh(x) | хиперболичен тангенс, th x | |
| ceil(x) | най-малкото цяло ≥ x, преобразувано в тип double | |
| Floor(x) | най-голямото цяло ≤ x, преобразувано в тип double | |
| fabs(x) | абсолютна стойност на x, |x| | |
| sqrt(x) | Корен квадратен от x, x ≥ 0 | |
| pow(x, n) | степенуване, xn (x и n са реални от тип double). | |

Логически оператори и оператори за сравнение



Как работят логическото „и“ и „или“

**А B A && B**

false false false

false true false

true false false

true true true

**А B A || B**

false false false

false true true

true false true

true true true

C++ Ключови думи

С ✓ са отбелязани ключовите думи, които ще използваме по УП

|  |  |  |
| --- | --- | --- |
| [alignas](http://en.cppreference.com/w/cpp/keyword/alignas) (since C++11) [alignof](http://en.cppreference.com/w/cpp/keyword/alignof) (since C++11) [and](http://en.cppreference.com/w/cpp/keyword/and)  [and\_eq](http://en.cppreference.com/w/cpp/keyword/and_eq) [asm](http://en.cppreference.com/w/cpp/keyword/asm) [atomic\_cancel](http://en.cppreference.com/w/cpp/language/transactional_memory) (TM TS) [atomic\_commit](http://en.cppreference.com/w/cpp/language/transactional_memory) (TM TS) [atomic\_noexcept](http://en.cppreference.com/w/cpp/language/transactional_memory) (TM TS) [auto](http://en.cppreference.com/w/cpp/keyword/auto)(1) [bitand](http://en.cppreference.com/w/cpp/keyword/bitand) [bitor](http://en.cppreference.com/w/cpp/keyword/bitor) [bool](http://en.cppreference.com/w/cpp/keyword/bool) ✓ [break](http://en.cppreference.com/w/cpp/keyword/break) ✓ [case](http://en.cppreference.com/w/cpp/keyword/case) ✓ [catch](http://en.cppreference.com/w/cpp/keyword/catch)  [char](http://en.cppreference.com/w/cpp/keyword/char) ✓ [char16\_t](http://en.cppreference.com/w/cpp/keyword/char16_t) (since C++11) [char32\_t](http://en.cppreference.com/w/cpp/keyword/char32_t) (since C++11) [class](http://en.cppreference.com/w/cpp/keyword/class)(1)  [compl](http://en.cppreference.com/w/cpp/keyword/compl) [concept](http://en.cppreference.com/w/cpp/keyword/concept) (concepts TS) [const](http://en.cppreference.com/w/cpp/keyword/const) ✓ [constexpr](http://en.cppreference.com/w/cpp/keyword/constexpr) (since C++11) [const\_cast](http://en.cppreference.com/w/cpp/keyword/const_cast) [continue](http://en.cppreference.com/w/cpp/keyword/continue) [decltype](http://en.cppreference.com/w/cpp/keyword/decltype) (since C++11) [default](http://en.cppreference.com/w/cpp/keyword/default)(1 ✓) [delete](http://en.cppreference.com/w/cpp/keyword/delete)(1) ✓ [do](http://en.cppreference.com/w/cpp/keyword/do) ✓ [double](http://en.cppreference.com/w/cpp/keyword/double) ✓ | [dynamic\_cast](http://en.cppreference.com/w/cpp/keyword/dynamic_cast) [else](http://en.cppreference.com/w/cpp/keyword/else) ✓ [enum](http://en.cppreference.com/w/cpp/keyword/enum) ✓ [explicit](http://en.cppreference.com/w/cpp/keyword/explicit) [export](http://en.cppreference.com/w/cpp/keyword/export)(1) [extern](http://en.cppreference.com/w/cpp/keyword/extern)(1) [false](http://en.cppreference.com/w/cpp/keyword/false) ✓ [float](http://en.cppreference.com/w/cpp/keyword/float) ✓ [for](http://en.cppreference.com/w/cpp/keyword/for) ✓ [goto](http://en.cppreference.com/w/cpp/keyword/goto) [if](http://en.cppreference.com/w/cpp/keyword/if) ✓ [inline](http://en.cppreference.com/w/cpp/keyword/inline)(1) [int](http://en.cppreference.com/w/cpp/keyword/int) ✓ [import](http://en.cppreference.com/mwiki/index.php?title=cpp/keyword/import&action=edit&redlink=1) (modules TS) [long](http://en.cppreference.com/w/cpp/keyword/long) ✓ [module](http://en.cppreference.com/mwiki/index.php?title=cpp/keyword/module&action=edit&redlink=1) (modules TS) [mutable](http://en.cppreference.com/w/cpp/keyword/mutable)(1) [namespace](http://en.cppreference.com/w/cpp/keyword/namespace) ✓ [new](http://en.cppreference.com/w/cpp/keyword/new) ✓ [noexcept](http://en.cppreference.com/w/cpp/keyword/noexcept) (since C++11) [not](http://en.cppreference.com/w/cpp/keyword/not) [not\_eq](http://en.cppreference.com/w/cpp/keyword/not_eq) [nullptr](http://en.cppreference.com/w/cpp/keyword/nullptr) (since C++11) [operator](http://en.cppreference.com/w/cpp/keyword/operator) [or](http://en.cppreference.com/w/cpp/keyword/or) [or\_eq](http://en.cppreference.com/w/cpp/keyword/or_eq) [private](http://en.cppreference.com/w/cpp/keyword/private) [protected](http://en.cppreference.com/w/cpp/keyword/protected) [public](http://en.cppreference.com/w/cpp/keyword/public) [register](http://en.cppreference.com/w/cpp/keyword/register)(2) | [reinterpret\_cast](http://en.cppreference.com/w/cpp/keyword/reinterpret_cast) [requires](http://en.cppreference.com/w/cpp/keyword/requires) (concepts TS) [return](http://en.cppreference.com/w/cpp/keyword/return) ✓ [short](http://en.cppreference.com/w/cpp/keyword/short) ✓ [signed](http://en.cppreference.com/w/cpp/keyword/signed) ✓ [sizeof](http://en.cppreference.com/w/cpp/keyword/sizeof)(1) [static](http://en.cppreference.com/w/cpp/keyword/static) [static\_assert](http://en.cppreference.com/w/cpp/keyword/static_assert) (since C++11) [static\_cast](http://en.cppreference.com/w/cpp/keyword/static_cast) [struct](http://en.cppreference.com/w/cpp/keyword/struct)(1) ✓ [switch](http://en.cppreference.com/w/cpp/keyword/switch) ✓ [synchronized](http://en.cppreference.com/w/cpp/language/transactional_memory) (TM TS) [template](http://en.cppreference.com/w/cpp/keyword/template) ✓ [this](http://en.cppreference.com/w/cpp/keyword/this) [thread\_local](http://en.cppreference.com/w/cpp/keyword/thread_local) (since C++11) [throw](http://en.cppreference.com/w/cpp/keyword/throw) [true](http://en.cppreference.com/w/cpp/keyword/true) ✓ [try](http://en.cppreference.com/w/cpp/keyword/try) [typedef](http://en.cppreference.com/w/cpp/keyword/typedef) [typeid](http://en.cppreference.com/w/cpp/keyword/typeid) [typename](http://en.cppreference.com/w/cpp/keyword/typename) ✓ [union](http://en.cppreference.com/w/cpp/keyword/union) [unsigned](http://en.cppreference.com/w/cpp/keyword/unsigned) ✓ [using](http://en.cppreference.com/w/cpp/keyword/using)(1)  [virtual](http://en.cppreference.com/w/cpp/keyword/virtual) [void](http://en.cppreference.com/w/cpp/keyword/void) ✓ [volatile](http://en.cppreference.com/w/cpp/keyword/volatile) [wchar\_t](http://en.cppreference.com/w/cpp/keyword/wchar_t) [while](http://en.cppreference.com/w/cpp/keyword/while) ✓ [xor](http://en.cppreference.com/w/cpp/keyword/xor) [xor\_eq](http://en.cppreference.com/w/cpp/keyword/xor_eq) |

Декларация на променливи

Примери:  
  
1) int var=5; --- > стойността на променливата var е 5

2) double a=5;

double b=a; --- > стойността на променливата b е равна на стойността на a , т.е на 5

3) int a,b,c=3; --- > стойността на променливата c е 3, но стойностите на променливите a и b не са определени

4) int a;b;c; --- > Грешна декларация, правилната е : int a,b,c;

Оператори и вградени функции

Примери:

1)

int a=5,b=3;

int c=a\*b; --- > На c се присвоява стойността, която се получава след умножението на а и b, т.е 3\*5=15

2)

double a=2.4,b=3.4;

cout<<a+b; --- > на екрана се печата 5.8

cout<<”a+b” --- > на екрана се печата „a+b“

3)

int x=3;

pow(x,3) --- > това е равносилно на x\*x\*x=33=27

4)

double x=16;

sqrt(x) --- > това е функция, която пресмята корен квадратен, т.е = = 4

IF-ELSE statement

if(boolean\_expression)

{

// statement(s) will execute if the boolean expression is true

}

else

{

// statement(s) will execute if the boolean expression is false

}

Примери:

1)

if(a>=5)

{

cout<<”Variable a is >=5”;

}

else

{

cout<<”Variable a is <5”;

}

2)

if(a>=5&&a<=10)

{

cout<<”Variable a is in the interval [5;10]”;

}

else

{

cout<<”Variable a is not in the interval [5;10]”;

}

Повече информация за IF-ELSE statement [тук](https://www.tutorialspoint.com/cplusplus/cpp_if_else_statement.htm).