Содержание

стандартная библиотека C++

C Library

Стандартная библиотека C++ включает в себя спецификации стандарта ISO C90 из стандартной библиотеки C

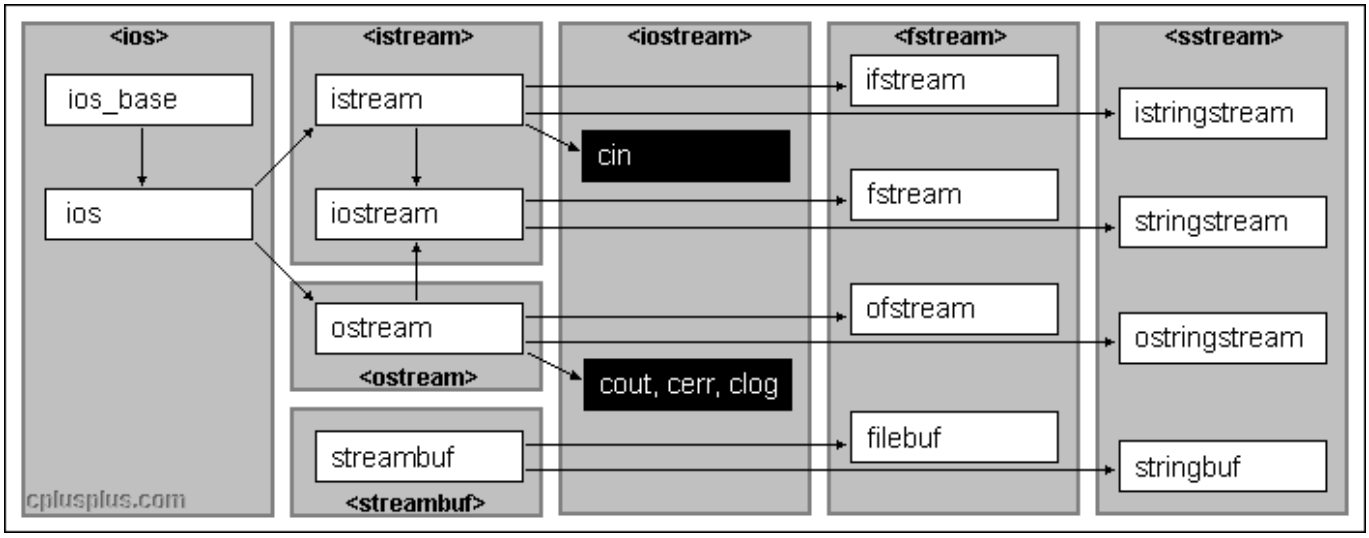
Заголовок	Стандарт	Описание
<assert.h>	C++98	Behaves same as <cassert>
<ctype.h>	C++98	Behaves as if each name from <cctype> is placed in global namespace
<errno.h>	C++98	Behaves same as <cerrno>
<fenv.h>	C++11	Behaves as if each name from <cfenv> is placed in global namespace
<float.h>	C++98	Behaves same as <cfloat>
<inttypes.h>	C++11	Behaves as if each name from <cinttypes> is placed in global namespace
<limits.h>	C++98	Behaves same as <climits>
<locale.h>	C++98	Behaves as if each name from <clocale> is placed in global namespace
<math.h>	C++98	Behaves as if each name from <cmath> is placed in global namespace, except for names of mathematical special functions
<setjmp.h>	C++98	Behaves as if each name from <csetjmp> is placed in global namespace
<signal.h>	C++98	Behaves as if each name from <csignal> is placed in global namespace
<stdarg.h>	C++98	Behaves as if each name from <cstdarg> is placed in global namespace
<stddef.h>	C++98	Behaves as if each name from <cstddef> is placed in global namespace, except for names of std::byte and related functions
<stdint.h>	C++11	Behaves as if each name from <cstdint> is placed in global namespace
<stdio.h>	C++98	Behaves as if each name from <cstdio> is placed in global namespace
<stdlib.h>	C++98	Behaves as if each name from <cstdlib> is placed in global namespace
<string.h>	C++98	Behaves as if each name from <cstring> is placed in global namespace
<time.h>	C++98	Behaves as if each name from <ctime> is placed in global namespace
<uchar.h>	C++11	Behaves as if each name from <cuchar> is placed in global namespace
<wchar.h>	C++98	Behaves as if each name from <cwchar> is placed in global namespace
<wctype.h>	C++98	Behaves as if each name from <cwctype> is placed in global namespace
Специальные заголовки совместимости с C		

Заголовок	Стандарт	Описание
<stdatomic.h>	C++23	Defines _Atomic and provides corresponding components in the C standard library
<stdbit.h>	C++26	Provides corresponding components in the C standard library
<stdckdint.h>	C++26	Provides corresponding components in the C standard library
Пустые заголовки		
<ccomplex>	C++11 устарел в C++17 удален в C++20	Simply includes the header <complex>
<complex.h>	C++11	Simply includes the header <complex>
<ctgmath>	C++11 устарел в C++17 удален в C++20	Simply includes the headers <complex> and <cmath>: the overloads equivalent to the contents of the C header <tgmath.h> are already provided by those headers
<tgmath.h>	C++11	Simply includes the headers <complex> and <cmath>
Бесполезные заголовки		
<ciso646>	C++98 удален в C++20	Empty header. The macros that appear in iso646.h in C are keywords in C++
<cstdalign>	C++11 устарел в C++17 удален в C++20	Defines one compatibility macro constant
<cstdbool>	C++11 устарел в C++17 удален в C++20	Defines one compatibility macro constant
<iso646.h>	C++98	Has no effect
<stdalign.h>	C++11	Defines one compatibility macro constant
<stdbool.h>	C++11	Defines one compatibility macro constant

Заголовок	Стандарт	Описание
<cstdlib>	C++98	General purpose utilities: program control, dynamic memory allocation, random numbers, sort and search

Заголовок	Стандарт	Описание
<execution>	C++17	Predefined execution policies for parallel versions of the algorithms and execution control components (since C26) ^ Заголовок ^ Стандарт ^ Описание ^ | <cfloat> | C++98 | Limits of floating-point types | | <climits> | C++98 | Limits of integral types | | <compare> | C++20 | Three-way comparison operator support | | <contracts> | C++26 | Contracts support library | | <coroutine> | C++20 | Coroutine support library | | <csetjmp> | C++98 | Macro (and function) that saves (and jumps) to an execution context | | <csignal> | C++98 | Functions and macro constants for signal management | | <cstdarg> | C++98 | Handling of variable length argument lists | | <cstddef> | C++98 | Standard macros and typedefs | | <cstdint> | C++11 | Fixed-width integer types and limits of other types | | <exception> | C++98 | Exception handling utilities | | <initializer_list> | C++11 | std::initializer_list class template | | <limits> | C++98 | Query properties of arithmetic types | | <new> | C++98 | Low-level memory management utilities | | <source_location> | C++20 | Supplies means to obtain source code location | | <stdfloat> | C++23 | Fixed-width floating-point types | | <typeindex> | C++11 | std::type_index | | <typeinfo> | C++98 | Runtime type information utilities | | <version> | C++20 | Supplies macros for verifying implementation status of library | ^ Заголовок ^ Стандарт ^ Описание ^ | <concepts> | C++20 | Fundamental library concepts | ^ Заголовок ^ Стандарт ^ Описание ^ | <cassert> | C++98 | Conditionally compiled macro that compares its argument to zero | | <cerrno> | C++98 | Macro containing the last error number | | <debugging> | C++26 | Debugging library | | <stacktrace> | C++23 | Stacktrace library | | <stdexcept> | C++98 | Standard exception types | | <system_error> | C++11 | Defines std::error_code, a platform-dependent error code | ^ Заголовок ^ Стандарт ^ Описание ^ | <memory> | C++98 | High-level memory management utilities | | <memory_resource> | C++17 | Polymorphic allocators and memory resources | | <scoped_allocator> | C++11 | Nested allocator class | ^ Заголовок ^ Стандарт ^ Описание ^ | <ratio> | C++11 | Compile-time rational arithmetic | | <type_traits> | C++11 | Compile-time type information utilities | ^ Заголовок ^ Стандарт ^ Описание ^ | <any> | C++17 | std::any class | | <bit> | C++20 | Bit manipulation functions | | <bitset> | C++98 | std::bitset class template | | <expected> | C++23 | std::expected class template | | <functional> | C++98 | Function objects, Function invocations, Bind operations and Reference wrappers | | <optional> | C++17 | std::optional class template | | <tuple> | C++11 | std::tuple class template | | <utility> | C++98 | Various utility components | | <variant> | C++17 | std::variant class template | ^ Заголовок ^ Стандарт ^ Описание ^ | <array> | C++11 | std::array container | | <deque> | C++98 | std::deque container | | <flat_map> | C++23 | std::flat_map and std::flat_multimap container adaptors | | <flat_set> | C++23 | std::flat_set and std::flat_multiset container adaptors | | <forward_list> | C++11 | std::forward_list container | | <hive> | C++26 | std::hive container | | <inplace_vector> | C++26 | std::inplace_vector container | | <list> | C++98 | std::list container | | <map> | C++98 | std::map and std::multimap associative containers | | <mdspan> | C++23 | std::mdspan view | | <queue> | C++98 | std::queue and std::priority_queue container adaptors | | <set> | C++98 | std::set and std::multiset associative containers | | | C++20 | std::span view | | <stack> | C++98 | std::stack container adaptor | | <unordered_map> | C++11 | std::unordered_map and std::unordered_multimap unordered associative containers | | <unordered_set> | C++11 | std::unordered_set and std::unordered_multiset unordered associative containers | | <vector> | C++98 | std::vector container | ^ Заголовок ^ Стандарт ^ Описание ^ | <iterator> | C++98 | Range iterators | ^ Заголовок ^ Стандарт ^ Описание ^ | <generator> | C++23 | std::generator class template | | <ranges> | C++20 | Range access, primitives, requirements, utilities and adaptors | ^ Заголовок ^ Стандарт ^ Описание ^ | <algorithm> | C++98 | Algorithms that operate on ranges | | <numeric> | C++98 | Numeric operations on values in ranges | ^ Заголовок ^ Стандарт ^ Описание ^ | <cstring> | C++98 | Various narrow character string handling functions | | <string> | C++98 | std::basic_string class template | | <string_view> | C++17 | std::basic_string_view class template | ^ Заголовок ^ Стандарт ^ Описание ^ | <cctype> | C++98 | Functions to determine the category of narrow characters | | <charconv> | C++17 | std::to_chars and std::from_chars | | <clocale> | C++98 | C localization utilities | | <codecvt> | C++11(deprecated in C17)(removed in C26) Unicode conversion facilities | | <cuchar> | C++11 | C-style Unicode character conversion functions | | <cwchar> | C++98 | Various wide and multibyte string handling functions | | <cwctype> | C++98 | Functions to determine the category of wide characters | | <format> | C++20 | Formatting library including std::format | | <locale> | C++98 | Localization utilities | | <regex> | C++11 | Classes, algorithms and iterators to support regular expression processing | | <text_encoding> | C++26 | Text encoding identifications | ^ Заголовок ^ Стандарт ^ Описание ^ | <cfenv> | C++11 | Floating-point environment access functions | | <cmath> | C++98 | Common mathematics functions | | <complex> | C++98 | Complex number type | | <linalg> | C++26 | Basic linear algebra algorithms (BLAS) | | <numbers> | C++20 | Math constants | | <random> | C++11 | Random number generators and distributions | | <simd> | C++26 | Data-parallel types and operations on these types | | <valarray> | C++98 | Class for representing and manipulating arrays of values | ^ Заголовок ^ Стандарт ^ Описание ^ | <chrono> | C++11 | C time utilities
<ctime>	C++98	C-style time/date utilities

связь классов



Заголовок	Стандарт	Описание
<cinttypes>	C++11	Formatting macros, intmax_t and uintmax_t math and conversions
<cstdio>	C++98	C-style input-output functions
<filesystem>	C++17	std::filesystem::path class and supporting functions
<fstream>	C++98	std::basic_fstream, std::basic_ifstream, std::basic_ofstream class templates and typedefs
<iomanip>	C++98	Helper functions to control the format of input and output
<ios>	C++98	std::ios_base class, std::basic_ios class template and typedefs
<iosfwd>	C++98	Forward declarations of all classes in the input/output library
<iostream>	C++98	Several standard stream objects
<istream>	C++98	std::basic_istream class template and typedefs
<ostream>	C++98	std::basic_ostream, std::basic_iostream class templates and typedefs
<print>	C++23	Formatted output library including std::print
<spanstream>	C++23	std::basic_spanstream, std::basic_ispanstream, std::basic_ospanstream class templates and typedefs
<sstream>	C++98	std::basic_stringstream, std::basic_istringstream, std::basic_ostringstream class templates and typedefs
<streambuf>	C++98	std::basic_streambuf class template
<strstream>		