

Обновление для C++ 20



# Экскурсия по C++

Третье издание

Бьярне Страуструп



Углубленное изучение C++

Бьярне Страуструп



# Содержание

Содержание.....	2
Тур по C++ .....	9
Благодарность .....	11
Основы.....	12
1.1 Введение.....	12
1.2 Программы.....	12
1.2.1 Hello, World! .....	13
1.3 Функции.....	15
1.4 Типы, Переменные и Арифметика.....	16
1.4.1 Арифметика.....	17
1.4.2 Инициализация.....	19
1.5 Область видимости и время жизни.....	20
1.6 Константы .....	21
1.7 Указатели, Массивы и Ссылки .....	22
1.7.1 Нулевой указатель .....	24
1.8 Условные операторы .....	25
1.9 Сопоставление с аппаратным обеспечением .....	27
1.9.1 Присваивание .....	27
1.9.2 Инициализация.....	28
1.10 Советы.....	29
Пользовательские типы.....	31
2.1 Введение.....	31
2.2 Структуры.....	32
2.3 Классы.....	33
2.4 Перечисления .....	35
2.5 Объединения.....	36
2.6 Советы .....	38
Модульность.....	39
3.1 Введение.....	39
3.2 Раздельная компиляция .....	40
3.2.1 Заголовочные файлы.....	40
3.2.2 Модули .....	42
3.3 Пространства имён .....	45
3.4 Аргументы функции и возвращаемые значения .....	47
3.4.1 Передача аргументов.....	47
3.4.2 Возвращение значений.....	48
3.4.3 Выведение типа возвращаемого значения .....	50
3.4.4 Суффиксная запись типа возвращаемого значения.....	50
3.4.5 Структурное связывание.....	50
3.5 Советы .....	52
Обработка ошибок.....	53
4.1 Введение.....	53
4.2 Исключения .....	54
4.3 Инварианты.....	55
4.4 Альтернативные способы обработки ошибок .....	57

4.5 Утверждения.....	59
4.5.1 assert().....	60
4.5.2 static_assert().....	60
4.5.3 noexcept.....	61
4.6 Советы.....	61
Классы.....	63
5.1 Введение.....	63
5.1.1 Классы.....	64
5.2 Конкретные типы.....	64
5.2.1 Арифметические типы.....	65
5.2.2 Контейнеры.....	67
5.2.3 Инициализация контейнеров.....	68
5.3 Абстрактные типы.....	70
5.4 Виртуальные функции.....	73
5.5 Иерархии классов.....	73
5.5.1 Преимущества иерархий.....	76
5.5.2 Навигация в иерархии.....	77
5.5.3 Предотвращение утечки ресурсов.....	78
5.6 Советы.....	79
Основные операции.....	81
6.1 Введение.....	81
6.1.1 Основные операции.....	81
6.1.2 Преобразования типов.....	83
6.1.3 Инициализация элементов.....	84
6.2 Копирование и перемещение.....	84
6.2.1 Копирование контейнеров.....	84
6.2.2 Перемещение контейнеров.....	86
6.3 Управление ресурсами.....	88
6.4 Перегрузка операторов.....	90
6.5 Стандартные операции.....	91
6.5.1 Операторы сравнения.....	91
6.5.2 Операции с контейнерами.....	92
6.5.3 Итераторы и “умные указатели”.....	93
6.5.4 Операции ввода-вывода.....	93
6.5.5 swap().....	94
6.5.6 hash<>.....	94
6.6 Пользовательские литералы.....	94
6.7 Советы.....	95
Шаблоны.....	97
7.1 Введение.....	97
7.2 Параметризованные типы.....	97
7.2.1 Ограниченные аргументы шаблона.....	99
7.2.2 Аргументы-значения шаблона.....	100
7.2.3 Выведение типов аргументов шаблонов.....	101
7.3 Параметризованные операции.....	102
7.3.1 Шаблоны функций.....	103
7.3.2 Функциональные объекты.....	103

7.3.3 Лямбда выражения.....	105
7.3.3.1 Лямбды как аргументы функции .....	105
7.3.3.2 Лямбды для инициализации .....	106
7.3.3.3 Напоследок, <code>finally()</code> .....	108
7.4 Механизмы шаблонов .....	108
7.4.1 Шаблоны переменных .....	109
7.4.2 Псевдонимы.....	109
7.4.3 <code>if</code> времени компиляции.....	110
7.5 Советы .....	111
Концепты и обобщенное программирование .....	113
8.1 Введение.....	113
8.2 Концепты .....	114
8.2.1 Использование концептов.....	114
8.2.2 Перегрузка основанная на концептах .....	116
8.2.3 Правильный код.....	117
8.2.4 Определение концептов .....	117
8.2.4.1 Проверка определения .....	120
8.2.5 Концепты и <code>auto</code> .....	120
8.2.6 Концепты и типы.....	121
8.3 Обобщённое программирование.....	122
8.3.1 Использование концептов.....	122
8.3.2 Абстракции использующие шаблоны.....	123
8.4 Шаблоны с переменным числом аргументов .....	124
8.4.1 Выражения свёртки.....	125
8.4.2 Передача аргументов.....	126
8.5 Модель компиляции шаблонов.....	127
8.6 Советы .....	128
Обзор стандартной библиотеки .....	129
9.1 Введение.....	129
9.2 Компоненты стандартной библиотеки .....	130
9.3 Организация стандартной библиотеки .....	131
9.3.1 Пространства имён .....	131
9.3.2 Пространство имён <code>ranges</code> .....	132
9.3.3 Модули .....	133
9.3.4 Заголовочные файлы.....	133
9.4 Советы .....	134
Строки и регулярные выражения.....	135
10.1 Введение .....	135
10.2 Строки .....	135
10.2.1 Реализация <code>string</code> .....	137
10.3 Строковые представления .....	138
10.4 Регулярные выражения.....	139
10.4.1 Поиск .....	140
10.4.2 Описание регулярных выражений .....	141
10.4.3 Итераторы .....	145
10.5 Советы.....	145

Ввод и вывод.....	148
11.1 Введение.....	148
11.2 Вывод.....	149
11.3 Ввод.....	150
11.4 Состояния потоков I/O.....	151
11.5 I/O пользовательских типов.....	152
11.6 Форматирование вывода.....	153
11.6.1 Форматирование потока.....	153
11.6.2 Форматирование в стиле printf().....	155
11.7 Потоки.....	157
11.7.1 Стандартные потоки.....	157
11.7.2 Файловые потоки.....	158
11.7.3 Строковые потоки.....	158
11.7.4 Потоки памяти.....	159
11.7.5 Синхронизированные потоки.....	159
11.8 I/O в стиле Си.....	160
11.9 Файловая система.....	160
11.9.1 Пути.....	161
11.9.2 Файлы и каталоги.....	163
11.10 Советы.....	164
Контейнеры.....	166
12.1 Введение.....	166
12.2 vector.....	166
12.2.1 Элементы.....	169
12.2.2 Проверка диапазона.....	169
12.3 list.....	171
12.5 map.....	173
12.6 unordered_map.....	174
12.7 Аллокаторы.....	175
12.8 Обзор контейнеров.....	177
12.9 Советы.....	178
Алгоритмы.....	181
13.1 Введение.....	181
13.2 Применение итераторов.....	183
13.3 Типы итераторов.....	185
13.3.1 Поточковые итераторы.....	186
13.4 Использование предикатов.....	188
13.5 Обзор алгоритмов.....	189
13.6 Параллельные алгоритмы.....	190
13.7 Советы.....	191
Диапазоны.....	192
14.1 Введение.....	192
14.2 Представления.....	193
14.3 Генераторы.....	195
14.4 Конвейеры.....	195
14.5 Обзор концептов.....	196
14.5.1 Концепты типов.....	197

14.5.2	Концепты итераторов .....	199
14.5.3	Концепты диапазонов .....	200
14.6	Советы .....	201
Умные указатели и контейнеры .....		202
15.1	Введение .....	202
15.2	Указатели .....	203
15.2.1	unique_ptr и shared_ptr .....	204
15.2.2	span .....	207
15.3	Контейнеры .....	208
15.3.1	array .....	209
15.3.2	bitset .....	211
15.3.3	pair .....	212
15.3.4	tuple .....	214
15.4	Альтернативы .....	215
15.4.1	variant .....	215
15.4.2	optional .....	217
15.4.3	any .....	218
15.5	Советы .....	218
Утилиты .....		220
16.1	Введение .....	220
16.2	Время .....	220
16.2.1	Часы .....	221
16.2.2	Календари .....	221
16.2.3	Временные зоны .....	222
16.3	Адаптация функций .....	223
16.3.1	Лямбды как адапторы .....	223
16.3.2	mem_fn() .....	223
16.3.3	function .....	224
16.4	Функция типа .....	224
16.4.1	Предикаты типа .....	225
16.4.2	Условные свойства .....	227
16.4.3	Генераторы типов .....	228
16.4.4	Связанные типы .....	228
16.5	source_location .....	229
16.6	move() and forward() .....	229
16.7	Битовые манипуляции .....	231
16.8	Выход из программы .....	232
16.9	Советы .....	232
Числовые вычисления .....		234
17.1	Введение .....	234
17.2	Математические функции .....	235
17.3	Численные алгоритмы .....	236
17.3.1	Многопоточные численные алгоритмы .....	236
17.4	Комплексные числа .....	237
17.5	Случайные числа .....	238
17.6	Векторная арифметика .....	240
17.7	Числовые ограничения .....	240

17.8 Псевдонимы типов.....	240
17.9 Математические константы .....	241
17.10 Советы.....	241
Параллелизм.....	243
18.1 Введение.....	243
18.2 Задачи и <code>thread</code> .....	244
18.2.1 Передача аргументов .....	245
18.2.2 Возвращение результатов .....	246
18.3 Обмен данными .....	247
18.3.1 <code>mutex</code> и блокировки .....	247
18.3.2 <code>atomic</code> .....	248
18.4 Ожидание событий .....	249
18.5 Коммуникации задач .....	250
18.5.1 <code>future</code> и <code>promise</code> .....	251
18.5.2 <code>packaged_task</code> .....	252
18.5.3 <code>async()</code> .....	253
18.5.4 Остановка <code>thread</code> .....	254
18.6 Корутины (сопрограммы).....	255
18.6.1 Кооперативная многозадачность .....	256
18.7 Советы .....	259
История и совместимость .....	261
19.1 История .....	261
19.1.1 Временная шкала.....	262
19.1.2 Ранние годы .....	263
19.1.3 Стандарты ISO C++ .....	266
19.1.4 Стандарты и стиль.....	268
19.1.5 Использование C++ .....	269
19.1.6 Модель C++.....	269
19.2 Эволюция функций C++.....	270
19.2.1 Языковые особенности C++11 .....	270
19.2.2 Языковые особенности C++14 .....	271
19.2.3 Языковые особенности C++17 .....	271
19.2.4 Языковые особенности C++20 .....	272
19.2.5 Компоненты стандартной библиотеки C++11 .....	272
19.2.6 Компоненты стандартной библиотеки C++14.....	273
19.2.7 Компоненты стандартной библиотеки C++17 .....	273
19.2.8 Компоненты стандартной библиотеки C++20.....	273
19.2.9 Удаленные и устаревшие функции.....	274
19.3 Совместимость C/C++ .....	275
19.3.1 C и C++ - родные братья .....	275
19.3.2 Проблемы совместимости .....	277
19.3.2.1 Проблемы стиля .....	277
19.3.2.2 <code>void*</code> .....	278
19.3.2.3 Линковка .....	279
19.4 Библиография .....	279
19.5 Советы .....	282



Модуль std.....	284
A.1 Введение .....	284
A.2 Используйте то, что предлагает Ваша реализация .....	285
A.3 Используйте заголовки .....	285
A.4 Сделайте свой собственный module std .....	285
A.5 Советы .....	286
Index .....	287

**Тур по С++**  
**Третъе издание**  
**Bjarne Stroustrup**

Coverphoto by: Marco Pregolato ([Unsplash.com](https://unsplash.com/@marco_pregolato): @marco\_pregolato).

Author photo courtesy of Bjarne Stroustrup.

This book was typeset in Times and Helvetica by the author.

ISBN-13: 978-0-13-681648-5

ISBN-10: 0-13-681648-7

First printing, October 2022

# Предисловие

*Когда вы захотите проинструктировать,  
будьте кратки.  
– Цицерон*

C++ ощущается как новый язык. Сегодня я могу выразить свои идеи чётче, проще и непосредственнее, чем в C++98 или C++11. Кроме того, в результате программы лучше проверяются компилятором и быстрее работают.

В этой книге дается обзор C++, как он определен в C++20, текущем стандарте ISO C++, и реализован основными поставщиками C++. Кроме того, в ней упоминается пара библиотечных компонентов, используемых в настоящее время, но не планируемых к включению в стандарт до C++23.

Как и другие современные языки, C++ велик, и для эффективного использования требуется большое количество библиотек. Цель этой небольшой книги - дать опытному программисту представление о том, что представляет собой современный C++. Книга охватывает большинство основных языковых функций и основные компоненты стандартной библиотеки. Эту книгу можно прочитать всего за день или два, но очевидно, что требуется гораздо больше времени, чтобы научиться писать хороший код на C++. Однако цель здесь не в овладении мастерством, а в том, чтобы дать обзор языка, привести ключевые примеры и помочь программисту начать работу.

Предполагается, что вы уже программировали раньше. Если нет, пожалуйста, подумайте о прочтении учебника например *Программирование: принципы и практика использования C++ (Второе издание)* [[Stroustrup, 2014](#)], прежде чем продолжать чтение этой книги. Даже если вы программировали раньше, язык, который вы использовали, или приложения, которые вы писали, могут сильно отличаться от стиля C++, представленного здесь.

Представьте себе обзорную экскурсию по какому-нибудь городу, например, Копенгагену или Нью-Йорку. Всего за несколько часов вам покажут основные достопримечательности, расскажут несколько историй и дадут несколько советов о том, что делать дальше. После такой экскурсии вы *не* знаете город. Вы *не* понимаете всего, что видели и слышали; некоторые истории могут показаться странными или даже неправдоподобными. Вы *не* знаете, как ориентироваться в формальных и неформальных правилах, которые управляют жизнью в городе. Чтобы по-настоящему узнать город, нужно пожить в нем, часто несколько лет. Однако, если вам немного повезет, вы получите общее представление о том, что особенного в городе, и идеи о том, что может вас заинтересовать. После экскурсии может начаться настоящее исследование города.

В этой экскурсии представлены основные возможности языка C++, поддерживаемые ими стили программирования, такие как объектно-ориентированное и обобщённое программирование. В нем не делается попытка предоставить подробное, похожее на справочник руководство, представление языка по отдельным функциям. В лучших традициях учебников я пытаюсь объяснить ту или иную функцию перед ее использованием, но это не всегда возможно, и не все читают текст строго последовательно. Я предполагаю некоторую техническую зрелость от моих читателей. Итак, читателю рекомендуется использовать перекрестные ссылки и алфавитный указатель.

Аналогично, стандартные библиотеки представлены в виде примеров, а не в виде полного описания. Читателю рекомендуется искать дополнительные и вспомогательные материалы по мере необходимости. Экосистема C++ - это гораздо больше, чем просто возможности, предлагаемые стандартом ISO (например, библиотеки, системы сборки, инструменты анализа и среды разработки). В Интернете доступно огромное количество материалов (разного качества). Большинство читателей найдут полезные обучающие и обзорные видеоролики с конференций CppCon и Meeting C++. Для получения технических подробностей о языке и библиотеке, предлагаемых стандартом ISO C++, я рекомендую [\[Cplusplusreference\]](#). Например, когда я упоминаю функцию или класс стандартной библиотеки, можно легко найти их определение в других источниках, а изучив документацию по ним, можно найти множество связанных с ними средств.

В этом туре C++ представлен как единое целое, а не как слоеный пирог. Следовательно, я редко идентифицирую языковые функции как присутствующие в C, C++98 или более поздних стандартах ISO. Такую информацию можно найти в [главе 19](#) (История и совместимость). Я сосредотачиваюсь на основах и стараюсь быть кратким, но я не полностью устоял перед искушением представить новые функции, такие как модули ([§3.2.2](#)), концепты ([§8.2](#)) и корутины ([§18.6](#)). Небольшое предпочтение свежим разработкам также, по-видимому, удовлетворяет любопытство многих читателей, которые уже знакомы с какой-либо более старой версией C++.

Справочное руководство по языку программирования или стандарт просто указывают, что можно сделать с помощью языка, но программисты часто больше заинтересованы в том, чтобы научиться хорошо использовать язык программирования. Этот аспект частично рассматривается в ряде затронутых тем, частично в тексте и, в частности, в разделах с рекомендациями. Дополнительные рекомендации о том, что собой представляет хороший современный C++, можно найти в руководстве C++ Core Guidelines [\[Stroustrup, 2015\]](#). Core Guidelines могут стать хорошим источником для дальнейшего изучения идей, представленных в этой книге. Вы можете отметить поразительное сходство формулировок рекомендаций и даже нумерации рекомендаций между Core Guidelines и этой книгой. Одна из причин заключается в том, что первое издание A Tour of C++ было основным источником первоначальных Core Guidelines.

## Благодарность

Спасибо всем, кто помогал дополнять и исправлять предыдущие выпуски "Экспедиции по C++", особенно студентам моего курса "Проектирование с использованием C++" в Колумбийском университете. Спасибо Morgan Stanley за то, что дали мне время написать это третье издание. Спасибо Чаку Эллисону, Гаю Дэвидсону, Стивену Дьюхерсту, Кейт Грегори, Дэнни Калеву, Гору Нишанову и Дж.Си ван Винкелю за рецензию на книгу и предложения по многим улучшениям.

Эта книга была создана автором с использованием troff с использованием макросов Брайана Кернигана.

*Manhattan, New York*

*Bjarne Stroustrup*