

Содержание

Предисловие	22
Благодарности ко второму изданию	23
Благодарности Дэвида ко второму изданию	23
Благодарности Нико ко второму изданию	24
Благодарности Дуга ко второму изданию	24
Благодарности к первому изданию	25
Благодарности Нико	26
Благодарности Дэвида	26
О книге	27
Что необходимо знать, приступая к чтению этой книги	28
Структура книги в целом	28
Как читать эту книгу	29
Некоторые замечания о стиле программирования	30
Стандарты C++11, C++14 и C++17	31
Примечание редактора перевода	33
Исходные тексты примеров и дополнительная информация	33
Обратная связь с авторами	34
Ждем ваших отзывов!	34
Часть I. Основы	35
Зачем нужны шаблоны	35
Глава 1. Шаблоны функций	37
1.1. Первое знакомство с шаблонами функций	37
1.1.1. Определение шаблона	37
1.1.2. Применение шаблонов	38
1.1.3. Двухэтапная трансляция	40
1.2. Вывод аргумента шаблона	41
1.3. Несколько параметров шаблона	43
1.3.1. Параметр шаблона для возвращаемого типа	44
1.3.2. Вывод возвращаемого типа	45
1.3.3. Возвращаемый тип как общий тип	47

1.4. Аргументы шаблона по умолчанию	47
1.5. Перегрузка шаблонов функций	49
1.6. А не должны ли мы?..	54
1.6.1. Передача по значению или по ссылке	54
1.6.2. Об использовании <code>inline</code>	55
1.6.3. Об использовании <code>constexpr</code>	55
1.7. Резюме	56
Глава 2. Шаблоны классов	57
2.1. Реализация шаблона класса <code>Stack</code>	57
2.1.1. Объявление шаблона класса	58
2.1.2. Реализация функций-членов	59
2.2. Использование шаблона класса <code>Stack</code>	61
2.3. Частичное использование шаблонов классов	62
2.3.1. Концепты	63
2.4. Друзья	64
2.5. Специализации шаблонов классов	66
2.6. Частичная специализация	67
2.7. Аргументы шаблона класса по умолчанию	70
2.8. Псевдонимы типов	71
2.9. Вывод аргументов шаблона класса	74
2.10. Шаблонизированные агрегаты	77
2.11. Резюме	78
Глава 3. Нетиповые параметры шаблонов	79
3.1. Параметры шаблонов классов, не являющиеся типами	79
3.2. Параметры шаблонов функций, не являющиеся типами	82
3.3. Ограничения на параметры шаблонов, не являющиеся типами	83
3.4. Тип параметра шаблона <code>auto</code>	84
3.5. Резюме	88
Глава 4. Вариативные шаблоны	89
4.1. Шаблоны с переменным количеством аргументов	89
4.1.1. Примеры вариативных шаблонов	89
4.1.2. Перегрузка вариативных и невариативных шаблонов	90
4.1.3. Оператор <code>sizeof...</code>	91
4.2. Выражения свертки	92
4.3. Применения шаблонов с переменным количеством аргументов	94
4.4. Вариативные шаблоны классов и вариативные выражения	95
4.4.1. Вариативные выражения	96

4.4.2. Вариативные индексы	97
4.4.3. Вариативные шаблоны класса	97
4.4.4. Вариативные правила вывода	98
4.4.5. Вариативные базовые классы и <code>using</code>	99
4.5. Резюме	100
Глава 5. Основы работы с шаблонами	101
5.1. Ключевое слово <code>typename</code>	101
5.2. Инициализация нулем	102
5.3. Использование <code>this-></code>	104
5.4. Шаблоны для массивов и строковых литералов	105
5.5. Шаблоны-члены	108
5.5.1. Конструкция <code>.template</code>	114
5.5.2. Обобщенные лямбда-выражения и шаблоны членов	114
5.6. Шаблоны переменных	115
5.7. Шаблонные параметры шаблонов	118
5.8. Резюме	123
Глава 6. Семантика перемещения и <code>enable_if<></code>	125
6.1. Прямая передача	125
6.2. Шаблоны специальных функций-членов	129
6.3. Отключение шаблонов с помощью <code>enable_if<></code>	132
6.4. Использование <code>enable_if<></code>	134
6.5. Применение концептов для упрощения выражений <code>enable_if<></code>	138
6.6. Резюме	139
Глава 7. По значению или по ссылке?	141
7.1. Передача по значению	142
7.2. Передача по ссылке	144
7.2.1. Передача с помощью константной ссылки	144
7.2.2. Передача с помощью неконстантной ссылки	146
7.2.3. Передача с помощью передаваемой ссылки	148
7.3. Использование <code>std::ref()</code> и <code>std::cref()</code>	149
7.4. Работа со строковыми литералами и массивами	151
7.4.1. Специальные реализации для строковых литералов и обычных массивов	152
7.5. Работа с возвращаемыми значениями	153
7.6. Рекомендуемые объявления параметров шаблона	155
7.7. Резюме	158

Глава 8. Программирование времени компиляции	159
8.1. Шаблонное метапрограммирование	159
8.2. Вычисления с использованием <code>constexpr</code>	161
8.3. Выбор пути выполнения с помощью частичной специализации	163
8.4. SFINAE	165
8.4.1. Выражение SFINAE с <code>decltype</code>	169
8.5. Инструкция <code>if</code> времени компиляции	170
8.6. Резюме	172
Глава 9. Применение шаблонов на практике	173
9.1. Модель включения	173
9.1.1. Ошибки компоновщика	173
9.1.2. Шаблоны в заголовочных файлах	175
9.2. Шаблоны и <code>inline</code>	176
9.3. Предкомпилированные заголовочные файлы	177
9.4. Расшифровка романов об ошибках	180
9.5. Некоторые замечания	187
9.6. Резюме	188
Глава 10. Основные термины в области шаблонов	189
10.1. “Шаблон класса” или “шаблонный класс”	189
10.2. Подстановка, инстанцирование и специализация	190
10.3. Объявления и определения	191
10.3.1. Полные и неполные типы	192
10.4. Правило одного определения	193
10.5. Аргументы и параметры шаблонов	193
10.6. Резюме	195
Глава 11. Обобщенные библиотеки	197
11.1. Вызываемые объекты	197
11.1.1. Поддержка функциональных объектов	198
11.1.2. Работа с функциями-членами и дополнительными аргументами	200
11.1.3. Оборачивание вызовов функций	202
11.2. Другие утилиты для реализации обобщенных библиотек	205
11.2.1. Свойства типов	205
11.2.2. <code>std::addressof()</code>	207
11.2.3. <code>std::declval()</code>	207
11.3. Прямая передача временных значений	208
11.4. Ссылки в качестве параметров шаблонов	209

11.5. Откладывание вычислений	212
11.6. О чем следует подумать при написании обобщенных библиотек	214
11.7. Резюме	215
Часть II. Углубленное изучение шаблонов	217
Глава 12. Вглубь шаблонов	219
12.1. Параметризованные объявления	219
12.1.1. Виртуальные функции-члены	224
12.1.2. Связывание шаблонов	224
12.1.3. Первичные шаблоны	226
12.2. Параметры шаблонов	227
12.2.1. Параметры типа	227
12.2.2. Параметры, не являющиеся типами	228
12.2.3. Шаблонные параметры шаблонов	229
12.2.4. Пакеты параметров шаблонов	231
12.2.5. Аргументы шаблона по умолчанию	233
12.3. Аргументы шаблонов	234
12.3.1. Аргументы шаблонов функций	235
12.3.2. Аргументы типов	237
12.3.3. Аргументы, не являющиеся типами	237
12.3.4. Шаблонные аргументы шаблонов	240
12.3.5. Эквивалентность	242
12.4. Вариативные шаблоны	243
12.4.1. Раскрытие пакета	244
12.4.2. Где может происходить раскрытие пакета	246
12.4.3. Пакеты параметров функций	248
12.4.4. Множественные и вложенные раскрытия пакетов	250
12.4.5. Раскрытия пакетов нулевой длины	251
12.4.6. Выражения свертки	252
12.5. Друзья	254
12.5.1. Дружественные классы шаблонов классов	254
12.5.2. Дружественные функции шаблонов классов	256
12.5.3. Дружественные шаблоны	258
12.6. Заключительные замечания	258
Глава 13. Имена в шаблонах	261
13.1. Систематизация имен	261
13.2. Поиск имен	263
13.2.1. Поиск, зависящий от аргументов	265

13.2.2. ADL объявлений друзей	267
13.2.3. Внесение имен классов	268
13.2.4. Текущие инстанцирования	270
13.3. Синтаксический анализ шаблонов	271
13.3.1. Зависимость от контекста в нешаблонных конструкциях	272
13.3.2. Зависимые имена типов	275
13.3.3. Зависимые имена шаблонов	278
13.3.4. Зависимые имена в объявлениях <code>using</code>	279
13.3.5. ADL и явные аргументы шаблонов	281
13.3.6. Зависимые выражения	282
13.3.7. Ошибки компиляции	284
13.4. Наследование и шаблоны классов	285
13.4.1. Независимые базовые классы	285
13.4.2. Зависимые базовые классы	286
13.5. Заключительные замечания	290
Глава 14. Инстанцирование	293
14.1. Инстанцирование по требованию	293
14.2. Отложенное инстанцирование	295
14.2.1. Частичное и полное инстанцирование	296
14.2.2. Инстанцированные компоненты	296
14.3. Модель инстанцирования C++	300
14.3.1. Двухфазный поиск	300
14.3.2. Точки инстанцирования	301
14.3.3. Модель включения	305
14.4. Схемы реализации	306
14.4.1. Жадное инстанцирование	308
14.4.2. Инстанцирование по запросу	309
14.4.3. Итеративное инстанцирование	311
14.5. Явное инстанцирование	312
14.5.1. Ручное инстанцирование	313
14.5.2. Объявления явного инстанцирования	315
14.6. Инструкции <code>if</code> времени компиляции	316
14.7. В стандартной библиотеке	318
14.8. Заключительные замечания	319
Глава 15. Вывод аргументов шаблона	321
15.1. Процесс вывода	321
15.2. Выводимые контексты	323
15.3. Особые ситуации вывода	325

15.4. Списки инициализаторов	326
15.5. Пакеты параметров	327
15.5.1. Шаблоны оператора литерала	329
15.6. Ссылки на r-значения	330
15.6.1. Правила свертки ссылок	330
15.6.2. Передаваемые ссылки	331
15.6.3. Прямая передача	332
15.6.4. Сюрпризы вывода	335
15.7. SFINAE	336
15.7.1. Непосредственный контекст	337
15.8. Ограничения вывода	339
15.8.1. Допустимые преобразования аргументов	340
15.8.2. Аргументы шаблона класса	341
15.8.3. Аргументы вызова по умолчанию	342
15.8.4. Спецификации исключений	342
15.9. Явные аргументы шаблонов функций	343
15.10. Вывод из инициализаторов и выражений	346
15.10.1. Спецификатор типа <code>auto</code>	346
15.10.2. Запись типа выражения с помощью <code>decltype</code>	352
15.10.3. <code>decltype(auto)</code>	354
15.10.4. Особые случаи вывода <code>auto</code>	357
15.10.5. Структурированное связывание	360
15.10.6. Обобщенные лямбда-выражения	364
15.11. Шаблоны псевдонимов	367
15.12. Вывод аргументов шаблонов классов	368
15.12.1. Правила вывода	369
15.12.2. Неявные правила вывода	371
15.12.3. Прочие тонкости	373
15.13. Заключительные замечания	376
Глава 16. Специализация и перегрузка	379
16.1. Когда обобщенный код недостаточно хорош	379
16.1.1. Прозрачная настройка	380
16.1.2. Семантическая прозрачность	381
16.2. Перегрузка шаблонов функций	382
16.2.1. Сигнатуры	383
16.2.2. Частичное упорядочение перегруженных шаблонов функций	385
16.2.3. Правила формального упорядочения	386

16.2.4. Шаблоны и нешаблоны	388
16.2.5. Вариативные шаблоны функций	391
16.3. Явная специализация	393
16.3.1. Полная специализация шаблона класса	394
16.3.2. Полная специализация шаблона функции	398
16.3.3. Полная специализация шаблона переменной	400
16.3.4. Полная специализация члена	400
16.4. Частичная специализация шаблона класса	403
16.5. Частичная специализация шаблонов переменных	407
16.6. Заключительные замечания	408
Глава 17. Дальнейшее развитие	409
17.1. Ослабленные правила применения typename	410
17.2. Обобщенные параметры шаблонов, не являющиеся типами	410
17.3. Частичная специализация шаблонов функций	413
17.4. Именованные аргументы шаблонов	414
17.5. Перегруженные шаблоны классов	415
17.6. Вывод неконечных раскрытий пакетов	416
17.7. Регуляризация void	417
17.8. Проверка типов для шаблонов	418
17.9. Рефлексивное метапрограммирование	420
17.10. Работа с пакетами	422
17.11. Модули	423
Часть III. Шаблоны и проектирование	425
Глава 18. Полиморфная мощь шаблонов	427
18.1. Динамический полиморфизм	427
18.2. Статический полиморфизм	430
18.3. Сравнение динамического и статического полиморфизма	433
18.4. Применение концептов	435
18.5. Новые виды проектных шаблонов	437
18.6. Обобщенное программирование	437
18.7. Заключительные замечания	441
Глава 19. Реализация свойств типов	443
19.1. Пример: суммирование последовательности	443
19.1.1. Фиксированные свойства	444
19.1.2. Свойства-значения	447
19.1.3. Параметризованные свойства	452

19.2. Стратегии и классы стратегий	452
19.2.1. Различие между свойствами и стратегиями	454
19.2.2. Шаблоны членов и шаблонные параметры шаблонов	456
19.2.3. Комбинирование нескольких стратегий и/или свойств	457
19.2.4. Накопление с обобщенными итераторами	458
19.3. Функции типов	459
19.3.1. Типы элементов	460
19.3.2. Преобразующие свойства	462
19.3.3. Свойства-предикаты	468
19.3.4. Свойства типов результатов	471
19.4. Свойства на основе SFINAE	474
19.4.1. Принцип SFINAE и перегрузки функций	474
19.4.2. SFINAE и частичные специализации	479
19.4.3. Применение обобщенных лямбда-выражений со SFINAE	480
19.4.4. SFINAE и свойства	484
19.5. IsConvertibleT	488
19.6. Обнаружение членов	491
19.6.1. Обнаружение членов-типов	491
19.6.2. Обнаружение произвольных членов-типов	493
19.6.3. Обнаружение членов, не являющихся типами	494
19.6.4. Использование обобщенных лямбда-выражений для обнаружения членов	498
19.7. Прочие методы работы со свойствами	500
19.7.1. If-Then-Else	500
19.7.2. Обнаружение операций, не генерирующих исключения	503
19.7.3. Повышение удобства свойств	506
19.8. Классификация типов	508
19.8.1. Определение фундаментальных типов	509
19.8.2. Определение составных типов	511
19.8.3. Идентификация типов функций	515
19.8.4. Обнаружение типов классов	517
19.8.5. Обнаружение типов перечислений	518
19.9. Свойства стратегий	518
19.9.1. Типы параметров только для чтения	519
19.10. В стандартной библиотеке	522
19.11. Заключительные замечания	523
Глава 20. Перегрузка свойств типов	525
20.1. Специализация алгоритма	525
20.2. Диспетчеризация дескрипторов	527

20.3. Включение/отключение шаблонов функций	528
20.3.1. Предоставление нескольких специализаций	531
20.3.2. Откуда берется EnableIf?	533
20.3.3. if времени компиляции	534
20.3.4. Концепты	536
20.4. Специализация класса	537
20.4.1. Включение/отключение шаблонов классов	538
20.4.2. Диспетчеризация дескрипторов для шаблонов классов	539
20.5. Шаблоны, безопасные по отношению к инстанцированию	542
20.6. В стандартной библиотеке	547
20.7. Заключительные замечания	548
Глава 21. Шаблоны и наследование	551
21.1. Оптимизация пустого базового класса	551
21.1.1. Принципы размещения	551
21.1.2. Члены как базовые классы	554
21.2. Странно рекурсивный шаблон проектирования	557
21.2.1. Метод Бартона–Нэкмана	559
21.2.2. Реализации операторов	562
21.2.3. Фасады	563
21.3. Миксины	570
21.3.1. Странные миксины	572
21.3.2. Параметризованная виртуальность	573
21.4. Именованные аргументы шаблона	574
21.5. Заключительные замечания	578
Глава 22. Статический и динамический полиморфизм	581
22.1. Функциональные объекты, указатели и <code>std::function<></code>	581
22.2. Обобщение указателей на функции	583
22.3. Интерфейс моста	586
22.4. Стирание типа	586
22.5. Условная передача владения	588
22.6. Вопросы производительности	591
22.7. Заключительные замечания	591
Глава 23. Метапрограммирование	593
23.1. Состояние современного метапрограммирования	593
23.1.1. Метапрограммирование значений	593
23.1.2. Метапрограммирование типов	595

23.1.3. Гибридное метапрограммирование	596
23.1.4. Гибридное метапрограммирование с типами единиц	599
23.2. Размерности рефлексивного метапрограммирования	602
23.3. Стоимость рекурсивного инстанцирования	603
23.3.1. Отслеживание всех инстанцирований	605
23.4. Вычислительная полнота	606
23.5. Рекурсивное инстанцирование и рекурсивные аргументы шаблонов	607
23.6. Значения перечислений и статические константы	608
23.7. Заключительные замечания	610
Глава 24. Списки типов	613
24.1. Анатомия списков типов	613
24.2. Алгоритмы над списками типов	615
24.2.1. Индексация	615
24.2.2. Поиск наилучшего соответствия	616
24.2.3. Добавление в список типов	619
24.2.4. Обращение порядка типов в списке	621
24.2.5. Преобразование списка типов	622
24.2.6. Накопление списков типов	623
24.2.7. Сортировка вставками	626
24.3. Списки нетиповых параметров	629
24.3.1. Выводимые нетиповые параметры	632
24.4. Оптимизация алгоритмов с помощью раскрытий пакетов	632
24.5. Списки типов в стиле LISP	634
24.6. Заключительные замечания	636
Глава 25. Кортежи	637
25.1. Базовый дизайн Tuple	637
25.1.1. Хранилище	637
25.1.2. Создание кортежа	640
25.2. Базовые операции кортежей	641
25.2.1. Сравнение	641
25.2.2. Вывод	642
25.3. Алгоритмы для работы с кортежами	643
25.3.1. Кортежи как списки типов	643
25.3.2. Добавление элементов в кортеж и удаление их оттуда	644
25.3.3. Обращение порядка элементов кортежа	646
25.3.4. Индексные списки	647

25.3.5. Обращение с использованием индексных списков	648
25.3.6. Тасование и выбор	650
25.4. Распаковка кортежей	653
25.5. Оптимизация кортежей	654
25.5.1. Кортежи и оптимизация пустого базового класса	654
25.5.2. <code>get()</code> с константным временем работы	659
25.6. Индексы кортежа	660
25.7. Заключительные замечания	663
Глава 26. Контролируемые объединения	665
26.1. Хранилище	666
26.2. Дизайн	668
26.3. Запрос и извлечение значения	672
26.4. Инициализация, присваивание и уничтожение элементов	673
26.4.1. Инстанцирование	673
26.4.2. Уничтожение	674
26.4.3. Присваивание	675
26.5. Посетители	680
26.5.1. Возвращаемый тип <code>visit()</code>	683
26.5.2. Общий возвращаемый тип	684
26.6. Инициализация и присваивание <code>Variant</code>	686
26.7. Заключительные замечания	690
Глава 27. Шаблоны выражений	693
27.1. Временные объекты и отдельные циклы	693
27.2. Программирование выражений в аргументах шаблонов	699
27.2.1. Операнды шаблонов выражений	700
27.2.2. Тип <code>Array</code>	703
27.2.3. Операторы	705
27.2.4. Подведем итог	707
27.2.5. Присваивание шаблонам выражений	709
27.3. Производительность и ограничения шаблонов выражений	710
27.4. Заключительные замечания	711
Глава 28. Отладка шаблонов	715
28.1. Поверхностное инстанцирование	715
28.2. Статические утверждения	718
28.3. Архетипы	719

28.4. Трассировщики	721
28.5. Оракулы	725
28.6. Заключительные замечания	726
Приложение А. Правило одного определения	727
А.1. Единицы трансляции	727
А.2. Объявления и определения	728
А.3. Детали правила одного определения	729
А.3.1. Ограничения “одно на программу”	729
А.3.2. Ограничения “одно на единицу трансляции”	732
А.3.3. Ограничения эквивалентности единиц перекрестной трансляции	733
Приложение Б. Категории значений	737
Б.1. Традиционные l- и r-значения	737
Б.1.1. Преобразования l-значений в r-значения	738
Б.2. Категории значений, начиная с C++11	738
Б.2.1. Временная материализация	741
Б.3. Проверка категорий значений с помощью decltype	743
Б.4. Ссылочные типы	743
Приложение В. Разрешение перегрузки	745
В.1. Когда используется разрешение перегрузки	745
В.2. Упрощенное разрешение перегрузки	746
В.2.1. Неявный аргумент для функций-членов	749
В.2.2. Улучшение точного соответствия	751
В.3. Детали перегрузки	752
В.3.1. Предпочтение нешаблонных функций или более специализированных шаблонов	752
В.3.2. Последовательности преобразований	753
В.3.3. Преобразования указателей	754
В.3.4. Списки инициализаторов	755
В.3.5. Функторы и функции-суррогаты	758
В.3.6. Другие контексты перегрузки	759
Приложение Г. Стандартные утилиты для работы с типами	761
Г.1. Использование свойств типов	761
Г.1.1. <code>std::integral_constant</code> и <code>std::bool_constant</code>	762
Г.1.2. Что вы должны знать при использовании свойств	764

Г.2. Основные и составные категории типов	765
Г.2.1. Проверка основных категорий типов	766
Г.2.2. Проверка составных категорий типов	770
Г.3. Характеристики и операции над типами	771
Г.3.1. Прочие характеристики типов	771
Г.3.3. Взаимоотношения между типами	790
Г.4. Построение типов	792
Г.5. Прочие свойства	796
Г.6. Комбинирование свойств типов	799
Г.7. Прочие утилиты	801
Приложение Д. Концепты	803
Д.1. Использование концептов	803
Д.2. Определение концептов	806
Д.3. Перегрузка ограничений	808
Д.3.1. Поглощение ограничений	808
Д.3.2. Ограничения и диспетчеризация дескрипторов	810
Д.4. Советы	811
Д.4.1. Проверка концептов	811
Д.4.2. Гранулированность концептов	811
Д.4.3. Бинарная совместимость	812
Библиография	815
Форумы	815
Книги и веб-сайты	816
Глоссарий	821
Предметный указатель	833