

Алессандре и Кассандре
— Дэвид

Тем, кто заботится о людях и человечестве
— Нико

Эми, Тессе и Молли
— Дуг

Предисловие

Концепции шаблонов в C++ более 30 лет. Шаблоны C++ были описаны еще в 1990 году в книге *The Annotated C++ Reference Manual* (ARM; см. [39]), а до этого рассматривались в ряде более специализированных публикаций. Однако даже через десять лет ощущался недостаток литературы, которая бы фокусировалась на основных концепциях и передовых методах этой увлекательной, сложной и мощной возможности C++. Первым изданием данной книги мы хотели (возможно, несколько самонадеянно) решить эту проблему.

Со времени публикации первого издания в конце 2002 года в C++ изменилось очень многое. В стандарт C++ были добавлены новые возможности, кардинально изменившие сам язык, а непрерывные исследования сообщества программистов на C++ обнаружили новые методы программирования на основе шаблонов. Поэтому, сохранив цели первого издания, мы, по сути, написали новую книгу — о шаблонах в *современном C++*.

Авторы подошли к написанию этой книги с различным уровнем подготовки и с различными намерениями. Дэвид, опытный разработчик компиляторов и активный участник рабочих групп Комитета по стандартизации C++, которые развивают основы языка, заинтересован в точном и подробном описании всей мощи (и проблем) шаблонов. Нико, “обычный” программист и член рабочей группы Комитета по стандартизации библиотеки C++, заинтересован в таком понимании всех методов применения шаблонов, которое помогло бы активно и с выгодой их использовать. Дуг, разработчик библиотеки шаблонов, превратившийся в разработчика компиляторов и проектировщика языка, заинтересован в сборе, классификации и оценке многочисленных методов, используемых для построения библиотек шаблонов. Кроме того, все мы хотели бы поделиться этими знаниями с вами и всем сообществом, чтобы помочь избежать дальнейших недоразумений, путаниц или опасений при работе с шаблонами C++.

Как следствие, книга содержит и концептуальное введение с примерами из повседневной практики, и подробное описание точного поведения шаблонов. По дороге от основных принципов применения шаблонов до “искусства программирования шаблонов” вам будут открываться (или переоткрываться заново) такие методы, как статический полиморфизм, свойства типов, метапрограммирование и шаблоны выражений. Вы получите более глубокое понимание стандартной библиотеки C++, в которой почти весь код включает шаблоны.

Во время написания этой книги авторы узнали много нового и получили массу удовольствия. Надеемся, что с читателями произойдет то же самое. Приятного чтения!

Благодарности ко второму изданию

Написание книги — трудная работа. Ее поддержка — работа еще сложнее. Нам потребовалось более пяти лет, “размазанных” по последнему десятилетию, чтобы написать это новое издание книги, и эта работа не могла бы быть выполнена без поддержки и терпения множества людей.

В частности, это программисты, которые дали свои отзывы, сообщили о найденных ошибках и предложили возможные улучшения к первому изданию за последние 15 лет. Их просто слишком много, чтобы перечислить всех... Вы сами знаете, кто вы, и мы действительно благодарны вам за время, найденное для того, чтобы написать нам свои мысли и замечания. Пожалуйста, простите, если наши ответы были не всегда оперативны.

Мы хотели бы также поблагодарить всех, кто просматривал черновики этой книги и предоставил нам свои ценные отзывы и уточнения. Эти обзоры подняли книгу на значительно более высокий уровень качества и в очередной раз продемонстрировали, что хорошие вещи требуют вклада многих мудрых людей. Поэтому мы очень признательны таким людям, как Стив Дьюхарст (Steve Dewhurst), Говард Хиннант (Howard Hinnant), Микель Кипелайнен (Mikael Kilpeläinen), Дитмар Кюль (Dietmar Kühl), Даниэль Крюглер (Daniel Krügler), Невин Либер (Nevin Lieber), Андреас Найзер (Andreas Neiser), Эрик Ниблер (Eric Niebler), Ричард Смит (Richard Smith), Эндрю Саттон (Andrew Sutton), Хьюбер Тонг (Hubert Tong) и Вилль Вутилайнен (Ville Voutilainen).

Конечно же, спасибо всем поддерживавшим нас сотрудникам издательства Addison-Wesley/Pearson. В наше время профессиональная поддержка автора книги — большая редкость, но все эти люди были терпеливы и мудры и оказывались в нужное время в нужном месте со своими знаниями и профессионализмом. Мы благодарим таких сотрудников издательства, как Питер Гордон (Peter Gordon), Ким Бодихаймер (Kim Boedigheimer), Грег Денч (Greg Doench), Джули Нагил (Julie Nahil), Дана Вильсон (Dana Wilson) и Кэрол Лаллье (Carol Lallier).

Огромное спасибо всему сообществу \LaTeX за отличную текстовую систему и отдельно Фрэнку Миттельбаху (Frank Mittelbach) за помощь в решении проблем, связанных с \LaTeX .

Благодарности Дэвида ко второму изданию

Выход второго издания книги ожидался очень долгое время, и теперь, когда книга готова, я благодарен всем тем людям из моего окружения, кто сделал это возможным. В первую очередь это относится к моей жене Карине (Karina) и дочерям Алессандре (Alessandra) и Кассандре (Cassandra), которые дружно позволили мне занять под работу над книгой значительное время из моего “семейного расписания”, особенно в последний год работы. Всегда проявляли интерес к моей книге и мои родители, и всякий раз, когда я бывал у них в гостях, активно интересовались состоянием этого проекта.

Очевидно, что это техническая книга, и ее содержание отражает знания и опыт в программировании. Однако одних знаний и опыта недостаточно, чтобы

довести такую работу до завершения. Поэтому я очень благодарен Нико за принятое на себя (в дополнение к его вкладу как автора) решение производственных вопросов и вопросов менеджмента. Если эта книга полезна для вас, и вы в один прекрасный день встретитесь с Нико — не забудьте его за это поблагодарить. Я также благодарен Дугу за согласие вступить в команду и не выйти из нее несмотря на кучу своей собственной работы.

Своими находками с нами делились многие программисты на C++, и я благодарен всем им. Однако хочу выразить особую благодарность Ричарду Смиуту (Richard Smith), который на протяжении многих лет всегда терпеливо и со знанием дела отвечал на мои электронные письма с массой технических вопросов. Спасибо также моим коллегам Джону Спайсеру (John Spicer), Майку Миллеру (Mike Miller) и Майку Херрику (Mike Herrick) за обмен знаниями и создание атмосферы, способствующей дальнейшим исследованиям.

Благодарности Нико ко второму изданию

В первую очередь я хочу поблагодарить двух суперэкспертов в области языка, Дэвида и Дуга, которым я, как прикладной программист и эксперт в области библиотеки, задавал много глупых вопросов и от которых узнал много нового. Теперь я иногда чувствую себя тоже экспертом (конечно, только до того момента, пока у меня не возникнет очередной вопрос). Парни, это было здорово!

Все остальные благодарности — Ютте Экштейн (Jutta Eckstein). Ютта обладает прекрасной способностью поддерживать людей в их идеалах, идеях и целях. Хотя большинство людей испытывают это лишь изредка при встрече с ней или общей работе в области информационных технологий, мне повезло пользоваться этим в повседневной жизни. После всех этих лет я все же надеюсь, что это никогда не закончится.

Благодарности Дуга ко второму изданию

Моя сердечная благодарность — моей прекрасной жене Эми (Amy) и нашим девочкам Молли (Molly) и Тесса (Tessa). Их любовь и общение приносят мне ежедневно радость и уверенность, так необходимую для решения серьезнейших проблем в жизни и работе. Я хотел бы также поблагодарить моих родителей, которые привили мне любовь к учебе и поощряли меня к этой работе на протяжении всех этих лет.

Было очень приятно работать с Дэвидом и Нико, которые, будучи во многом противоположностями, хорошо дополняют друг друга. Дэвид привносит большую ясность в технические тексты, оттачивая их и делая предельно точными и информативными. Нико, с его исключительными организаторскими способностями, являлся силой, цементирующей наш коллектив, и привносил в него уникальную способность разбирать любые сложные технические вопросы и делать их более простыми, доступными и существенно более понятными.

Благодарности к первому изданию

Эта книга вобрала в себя мысли, концепции, решения и примеры из множества различных источников, и теперь мы хотели бы выразить свою признательность всем, кто оказывал нам помощь и поддержку на протяжении последних нескольких лет.

Прежде всего огромное спасибо нашим рецензентам, а также тем, кто высказывал свое мнение по самым первым вариантам рукописи. Без их участия нам не удалось бы довести книгу до такого уровня, какой она имеет сегодня. Нашими рецензентами были Кайл Блейни (Kyle Blaney), Томас Гшвинд (Thomas Gschwind), Деннис Менкл (Dennis Mancl), Патрик Мак-Киллен (Patrick McKillen), Ян Христиан ван Винкель (Jan Christian van Winkel). Особая благодарность Дитмару Кюлю (Dietmar Kühl), который тщательно прорецензировал и отредактировал всю книгу. Обратная связь, которую обеспечил Дитмар, помогла значительно повысить уровень книги.

Хотелось бы также выразить признательность всем людям и организациям, которые предоставили нам возможность протестировать вошедшие в книгу примеры на разных платформах с помощью различных компиляторов. Большое спасибо Edison Design Group за замечательный компилятор и его поддержку. Сотрудники этой группы помогли нам не только создавать эту книгу, но и приводить ее в соответствие со стандартами. Большое спасибо всем разработчикам свободно распространяемых компиляторов GNU и egcs (особая благодарность Джесону Меррилу (Jason Merrill) за отзывчивость), а также компании Microsoft за бета-версию Visual C++ (здесь мы контактировали с Джонатаном Кейвсом (Jonathan Caves), Гербом Саттером (Herb Sutter) и Джесоном Ширком (Jason Shirk)).

То, что сегодня составляет “ноосферу C++”, — плод коллективного творчества сетевого сообщества C++. Львиную долю этих знаний обеспечивают модерлируемые конференции Usenet — `comp.lang.c++.moderated` и `comp.std.c++`. Поэтому особое спасибо активным модераторам этих групп, которые смогли сделать обсуждение полезным и конструктивным. Мы хотим также поблагодарить каждого из тех, кто не один год подряд выкраивал время для описания и объяснения своих идей, желая сделать их нашим общим достоянием.

Трудно переоценить тот вклад, который внесли в работу над книгой сотрудники издательства Addison-Wesley. Мы выражаем особую признательность нашему редактору Дебби Лафферти (Debbie Lafferty) за ее деликатные “пинки”, дельные советы и добросовестную упорную работу над книгой. Спасибо также другим сотрудникам издательства — Тайрелль Олбах (Tyrell Albaugh), Банни Эймс (Bunny Ames), Мелани Бак (Melanie Buck), Жаклин Дюсетт (Jacquelyn Doucette), Чанде Лири-Коту (Chanda Leary-Coutu), Кэтрин Охала (Catherine Ohala) и Марти Рабинович (Marty Rabinowitz). Искренне благодарим Марину Ланг (Marina Lang), которая способствовала изданию этой книги в Addison-Wesley, а также Сюзан Винер (Susan Winer), выполнившую первое редактирование, которое помогло очертить контуры будущей книги.

Благодарности Нико

Прежде всего мне хотелось бы передать личную благодарность и бесчисленное количество поцелуев своей семье: Улли (Ulli), Лукасу (Lucas), Анике (Anica) и Фредерику (Frederic) — за их заботу, предупредительность и поддержку, которые так помогали мне во время работы над книгой.

Кроме того, я хотел бы сказать спасибо Дэвиду. Его знания и опыт огромны, но терпение оказалось поистине безграничным (временами я задавал ему на редкость глупые вопросы). Работать с ним очень интересно.

Благодарности Дэвида

Тем, что мне удалось завершить работу над этой книгой, я обязан своей жене Карине (Karina). Я чрезвычайно благодарен ей за ту роль, которую она играет в моей жизни. Когда в твоём ежедневном расписании множество одинаково первоочередных дел, написание книги “в свободное время” быстро превращается в утопию. Именно Карина помогла мне справляться с этим расписанием, учила меня говорить “нет”, чтобы выкроить время для работы и обеспечить постоянное продвижение вперед. А самое главное — она была потрясающей движущей силой этого проекта. Я каждый день благодарю Бога за ее дружбу и любовь.

И еще: я очень рад, что мне пришлось работать с Нико. Его вклад в создание книги измеряется не только непосредственно написанным текстом. Именно опыт и дисциплинированность Нико позволили нам перейти от моих графоманских попыток к хорошо организованному изданию.

“Мистер Шаблон” Джон Спайсер (John Spicer) и “мистер Перегрузка” Стив Адамчик (Steve Adamczyk) — замечательные друзья и коллеги. А также, по моему мнению, этот дуэт является последней инстанцией во всем, что касается основ языка C++. Именно они внесли ясность во многие сложные вопросы, освещенные в данной книге, и если вам случится найти ошибку в описании какого-то элемента языка C++, значит, по этому вопросу я не смог проконсультироваться с ними.

И наконец, я хотел бы выразить признательность всем тем, кто в разной степени помогал нам в работе над этим проектом. Многие из них не внесли непосредственный вклад в этот проект, но их участие и поддержка послужили для него огромной движущей силой. Это прежде всего мои родители; их любовь и ободрение были для меня чрезвычайно важны. Источником вдохновения были для меня и мои друзья, которые постоянно интересовались, как продвигается работа над книгой. Спасибо вам всем: Майкл Бэкманн (Michael Beckmann), Бретт и Джули Бин (Brett and Julie Beene), Джарран Карр (Jarran Carr), Симон Чанг (Simon Chang), Хо и Сара Чо (Ho and Sarah Cho), Кристоф Де Динечин (Christophe De Dinechin), Ева Дилман (Eva Deelman), Нейл Эберли (Neil Eberle), Сэссан Хазеги (Sassan Hazeghi), Викрам Кумар (Vikram Kumar), Джим и Линдсей Лонг (Jim and Lindsay Long), Р.Дж. Морган (R.J. Morgan), Майк Пуритано (Mike Puritano), Рагу Рагавендра (Ragu Raghavendra), Джим и Фуонг Шарп (Jim and Phuong Sharp), Грег Вогн (Greg Vaughn) и Джон Вигли (John Wiegley).

О книге

Первое издание этой книги было опубликовано 15 лет назад. Мы пытались написать полное руководство по шаблонам C++, надеясь на то, что оно будет полезно для практикующих программистов C++. Этот проект оказался успешным: было чрезвычайно приятно получать отзывы читателей, которым пригодился наш материал и которые обращались к книге снова и снова, как к справочнику.

Увы, первое издание уже устарело, и хотя большая часть изложенного в нем материала полностью соответствует современным концепциям C++, нет никаких сомнений, что эволюция языка, приведшая к понятию “современного C++” — стандартам C++11, C++14 и C++17, настоятельно требует существенного пересмотра материала из первого издания.

Во втором издании наша цель “верхнего уровня” остается неизменной: создание руководства по шаблонам C++, которое было бы и надежным справочником, и доступным учебником. Но в этот раз мы работаем с современным языком программирования C++, который представляет собой нечто значительно большее, чем язык, доступный во времена предыдущего издания.

Мы также понимаем, что ресурсы, посвященные программированию на C++, со времени первого издания существенно изменились (в лучшую сторону). Появилось несколько книг, которые весьма глубоко разбираются в конкретных приложениях с использованием шаблонов. Что еще более важно, сейчас в Интернете имеется гораздо больше информации о шаблонах C++ и методах их применения, как и примеров их использования. Так что в этом издании мы решили подчеркнуть широту методов, которые могут использоваться в различных приложениях.

Некоторые из представленных в первом издании методов устарели, потому что язык C++ теперь предлагает куда более прямые пути достижения того же результата. Эти методы убраны из книги (или низведены до небольших примечаний), и вместо них вы найдете новые методы, которые показывают текущее состояние дел при использовании новых возможностей (можно даже сказать — нового) языка.

Даже теперь, после того как мы прожили бок о бок с шаблонами C++ более 20 лет, программисты по-прежнему регулярно находят новые фундаментальные идеи, которые могут идеально вписаться в современные потребности в области развития программного обеспечения. Цель нашей книги — поделиться этими знаниями и обеспечить читателя всей необходимой информацией для развития нового понимания основных методик программирования на C++, а возможно, и новых открытий в этой области.

Что необходимо знать, приступая к чтению этой книги

Чтобы получить максимальную пользу от работы с книгой, читатель должен быть знаком с C++. В данной книге дается детальное описание конкретных возможностей языка программирования, но не основ самого языка. Предполагается знакомство читателя с концепцией классов и наследования, а также умение писать программы на C++ с использованием таких компонентов, как потоки ввода-вывода и контейнеры из стандартной библиотеки C++. Читатель должен быть также знаком с основными возможностями современного C++, такими как `auto`, `decltype`, семантика перемещения и лямбда-выражения. Кроме того, при необходимости в данной книге рассматриваются различные тонкие вопросы, которые могут не иметь прямого отношения к шаблонам. Таким образом обеспечивается доступность изложенного здесь материала как для квалифицированных специалистов, так и для программистов среднего уровня.

Изложение материала основано в первую очередь на стандартах языка C++, принятых в 2011, 2014 и 2017 годах. Однако на момент написания этой книги еще не высохли чернила на стандарте C++17, так что мы не ожидаем, что большинство читателей будет хорошо с ним знакомо. Все упомянутые стандарты оказали существенное влияние на поведение и использование шаблонов. Поэтому мы предоставляем краткое введение с описанием новых возможностей, которые имеют наибольшее отношение к нашей теме. Однако наша цель — не введение в современные стандарты C++ и не предоставление исчерпывающего описания отличий новых стандартов от предыдущих версий ([25] и [26]). Мы сосредоточиваемся в первую очередь на том, как шаблоны проектируются и используются в программах на языке C++, соответствующем современным стандартам ([27], [28] и [29]), и время от времени подчеркиваем ситуации, когда современные стандарты C++ позволяют или поощряют использовать технологии, отличные от предлагавшихся более ранними стандартами.

Структура книги в целом

Цель данной книги — предоставить читателю информацию, необходимую для работы с шаблонами и использования в полной мере их преимуществ; кроме того, книга призвана обеспечить читателей информацией, которая позволит опытным программистам преодолеть современные ограничения в этой области. Исходя из этого, мы разбили материал книги на части.

- Часть I, “Основы”, представляет собой введение в основные концепции, положенные в основу шаблонов. Эта часть написана в стиле учебника.
- Часть II, “Углубленное изучение шаблонов”, предоставляет детальные сведения о языке. Эта часть является неплохим справочником по конструкциям, связанным с шаблонами.
- Часть III, “Нетиповые параметры шаблонов”, поясняет фундаментальные методы проектирования и кодирования, поддерживаемые шаблонами C++. Они простираются от почти тривиальных идей до сложнейших идиом.

Каждая из перечисленных частей книги состоит из нескольких глав. Кроме того, книга включает несколько приложений, которые охватывают материал, относящийся не только к шаблонам (например, вопросы перегрузки в C++). Дополнительное приложение охватывает концепты, которые являются фундаментальными расширениями возможностей шаблонов и которые будут включены в будущие стандарты (вероятно, C++20).

Главы, входящие в состав первой части книги, требуют последовательного изучения. Например, глава 3 основана на материале, рассмотренном в главе 2. Однако в других частях книги связь между главами выражена не столь явно. Перекрестные ссылки помогут вам при необходимости перемещаться между различными темами книги.

Наконец, в книге имеется предметный указатель, который предоставляет еще одну возможность читать книгу не последовательно.

Как читать эту книгу

Если вы являетесь программистом на C++ и хотите получить общее представление о концепции шаблонов и поближе познакомиться с ней, то вам следует тщательно изучить часть I, “Основы”. С этим материалом имеет смысл хотя бы бегло ознакомиться даже тем, кто с шаблонами уже “на ты”, чтобы прочувствовать стиль и освоиться с используемой в книге терминологией. Эта часть также охватывает некоторые “материально-технические” аспекты, касающиеся организации исходного кода, содержащего шаблоны.

В зависимости от того, какой метод изучения материала вы предпочитаете, можно либо основательно изучить детальную информацию о шаблонах из части II, “Углубленное изучение шаблонов”, либо познакомиться с приемами практического программирования в части III, “Шаблоны и проектирование” (обращаясь к части II, если возникнут какие-либо вопросы). Последнее представляется особенно целесообразным в случае, если вы приобрели эту книгу для конкретных практических целей.

Приложения содержат большое количество полезной информации, на которую сделано много ссылок в основной части книги. Кроме того, мы старались сделать их интересными и в качестве самостоятельного материала.

Опыт подсказывает, что лучше всего новые знания усваиваются на примерах. Поэтому их вы найдете в книге большое количество. Иногда это всего лишь несколько строк кода, иллюстрирующих теоретическое положение, иногда — полноценные программы, реализующие конкретное применение материала. В последнем случае примеры снабжены комментариями C++ с описанием пути к файлу, в котором содержится код программы. Все эти файлы можно найти на сайте данной книги по адресу <http://www.tmplbook.com>.

Некоторые замечания о стиле программирования

У каждого программиста на C++ свой стиль программирования, и авторы данной книги также не составляют исключения. Понятие стиля включает обычные вопросы: где помещать пробелы, разделители (скобки, фигурные скобки) и т.п. В целом мы старались придерживаться единого стиля, хотя иногда по ходу изложения приходилось делать исключения. Например, чтобы придать коду больше наглядности, в разделах руководства были широко использованы пробелы и осмысленные имена, в то время как при рассмотрении более сложных вопросов предпочтение отдавалось компактности.

Хотелось бы обратить внимание читателя на то, что в данной книге применяется несколько необычный подход к записи объявлений типов, параметров и переменных. Очевидно, что при объявлении возможно использование нескольких стилей:

```
void foo(const int &x);
void foo(const int& x);
void foo(int const &x);
void foo(int const& x);
```

Хотя этот порядок записи менее распространен, для обозначения целочисленной константы мы решили использовать `int const` вместо `const int`. Сделано это было по двум причинам. Во-первых, такой порядок обеспечивает более очевидный ответ на вопрос: “*Что* именно является константой?” “*Что*” — это всегда то, что находится перед квалификатором `const`. В самом деле, хотя выражение

```
const int N = 100;
```

эквивалентно выражению

```
int const N = 100;
```

не существует аналогичной эквивалентной записи для

```
int* const bookmark; // Указатель не может меняться; значение
                    // же, на которое он указывает - может.
```

в которой можно было бы разместить квалификатор `const` до `*`. В этом примере константой является сам указатель, а не значение типа `int`, на которое он указывает.

Вторая причина связана с синтаксической подстановкой, часто встречающейся при работе с шаблонами. Рассмотрим два следующих определения типов:

```
typedef char* CHARS;
typedef CHARS const CPTR; // Константный указатель на char
```

или с использованием ключевого слова `using`:

```
using CHARS = char *;
using CPTR = CHARS const; // Константный указатель на char
```

Смысл второго объявления сохраняется при текстуальной замене `CHARS` тем, что это слово означает:

```
typedef char* const CPTR; // Константный указатель на char
```

или:

```
using CPTR = char* const; // Константный указатель на char
```

Однако если мы напишем `const` до квалифицируемого типа, то этот принцип окажется неприменим. Рассмотрим альтернативу двум представленным ранее первым определениям типа:

```
typedef char* CHARS;
typedef const CHARS CPTR; // Константный указатель на char
```

Текстуальная замена `CHARS` приводит к типу с иным значением:

```
typedef const char* CPTR; // Указатель на константный char
```

Очевидно, что сказанное выше справедливо и для спецификатора `volatile`.

Что касается расстановки пробелов, то мы решили помещать пробел между амперсандом и именем параметра:

```
void foo(int const& x);
```

Поступая таким образом, мы подчеркиваем разделение между типом параметра и именем параметра. Однако при такой записи еще более запутанными становятся объявления наподобие

```
char* a, b;
```

Здесь согласно правилам, унаследованным из C, `a` является указателем, а `b` — обычной символьной переменной. Чтобы исключить такого рода путаницу, мы просто стараемся избегать объявления нескольких переменных приведенным образом.

Эта книга посвящена в первую очередь возможностям языка программирования. Однако многие методики, возможности и вспомогательные шаблоны в настоящее время входят в стандартную библиотеку C++. Для их объединения мы демонстрируем шаблонные методы, иллюстрируя, как они применяются для реализации некоторых компонентов библиотеки, и используем возможности стандартной библиотеки для создания более сложных примеров. Поэтому мы применяем не только такие заголовочные файлы, как `<iostream>` и `<string>` (которые содержат шаблоны, но не так существенны для определения других шаблонов), но и `<cstdlib>`, `<utilities>`, `<functional>` и `<type_traits>` (предоставляющие “кирпичи” для построения более сложных шаблонов).

Кроме того, приложение Г, “Стандартные утилиты для работы с типами”, содержит вспомогательные возможности, предоставляемые стандартной библиотекой C++, а также детальное описание всех стандартных свойств типов. Они широко используются в сложном программировании шаблонов.

Стандарты C++11, C++14 и C++17

Исходный стандарт C++ был опубликован в 1998 году и дополнен *техническими поправками* в 2003 году, которые внесли незначительные исправления

и уточнения в первоначальный стандарт. Эти “старые стандарты C++” известны как C++98 и C++03.

Стандарт C++11 был первым крупным пересмотром стандарта под руководством Комитета по стандартизации ISO C++, который привнес в язык программирования множество новых возможностей, связанных с работой с шаблонами, в частности таких.

- Вариативные шаблоны (variadic templates, шаблоны с переменным количеством параметров).
- Шаблоны псевдонимов (alias templates).
- Семантика перемещения, ссылки на r-значения и прямая передача (perfect forwarding).
- Стандартные свойства типов (type traits).

За C++11 последовали стандарты C++14 и C++17, которые добавили в язык новые возможности, хотя и не столь драматические, как внесенные стандартом C++11¹. Новые возможности, связанные с работой с шаблонами и описанные в книге, включают следующее (но не ограничиваются перечисленным).

- Шаблоны переменных (C++14).
- Обобщенные лямбда-выражения (C++14).
- Вывод аргументов шаблона класса (C++17).
- `if` времени компиляции (C++17).
- Выражения свертки (C++17).

Мы также рассмотрим *концепты* (интерфейсы шаблонов), которые в настоящее время планируются к включению в стандарт C++20.

На момент написания книги стандарты C++11 и C++14 достаточно широко поддерживались основными компиляторами, как и многие возможности стандарта C++17. Тем не менее компиляторы сильно отличаются в их поддержке различных возможностей языка. Одни компиляторы будут компилировать большую часть кода из этой книги, другие не смогут справиться с рядом приведенных примеров. Однако мы ожидаем, что эта проблема будет вскоре решена — по мере того как программисты будут все активнее требовать поддержку стандарта от поставщиков.

Как бы то ни было, язык программирования C++ продолжает развиваться. Эксперты сообщества C++ (независимо от их участия в работе Комитета по стандартизации C++) обсуждают различные способы улучшения языка, и несколько

¹Заметим, что в C++ синоним типа определяет псевдоним, а не новый тип (см. раздел 2.8), например:

```
typedef int Length; //Length определяется как псевдоним int
int i = 42;
Length l = 88;
i = l; // ОК
l = i; // ОК
```

таких предлагаемых усовершенствований связаны с шаблонами. В главе 17, “Дальнейшее развитие”, описаны некоторые тенденции в этой области.

Примечание редактора перевода

На странице найденных ошибок и опечаток к данной книге имеется следующее примечание авторов книги.

Мы забыли уточнить, что до стандарта C++17 было необходимо добавлять определение к объявлению статического constexpr-члена класса:

```
struct A {
    static constexpr int n = 5; // В C++11/C++14 - объявление,
                               // начиная с C++17 - определение
};
```

До стандарта C++17 этот код представляет собой только объявление `n`. Но начиная со стандарта C++17 это объявление является также и определением. Таким образом, работая со стандартами, предшествующими C++17, в точности в одной единице трансляции необходимо предоставить соответствующее определение (которое в ряде примеров в книге оказалось опущено):

```
// До C++17 - в одной, и только одной единице трансляции:
constexpr int A::n; // Определение в C++11/C++14;
                   // начиная с C++17 - не рекомендуемое
                   // избыточное объявление
```

Причина, по которой никто из нас этого не заметил, кроется в том, что зачастую такое определение оказывается попросту ненужным. Фактически член всегда можно передать по значению. И только при передаче членов по ссылке, при условии, что компилятор не оптимизирует такой вызов, возникают ошибки компоновки:

```
std::cout << A::n; // ОК даже без определения (оператор
                  // ostream::operator<<(int) получает A::n
                  // по значению)
int inc(const int& i); // Передача по ссылке!
std::cout << inc(A::n); // Ошибка компоновки при отсутствии
                       // определения и оптимизации
```

Просьба к читателям учесть это замечание при чтении книги.

Исходные тексты примеров и дополнительная информация

Исходные тексты всех примеров программ и более подробную информацию по книге можно найти на сайте по адресу <http://www.templbook.com>.

Обратная связь с авторами

Мы приветствуем любые конструктивные отклики читателей — как отрицательные, так и положительные. Нам пришлось основательно потрудиться, чтобы создать для вас книгу, которую, надеемся, вы оцените достаточно высоко. Однако в определенный момент мы просто вынуждены были прервать работу над ней, поскольку подошел срок “выпуска продукта”. Следовательно, ни один из наших читателей не застрахован от того, что при изучении материала ему придется столкнуться с ошибками или несогласованностью, а также с отдельными моментами, которые нуждаются в доработке, или с тем, что отдельные темы в книге не освещены вообще. Ваши отклики дают нам возможность проинформировать всех читателей через веб-сайт книги о найденных вами “узких местах” и улучшить таким образом ее последующие издания.

Связываться с нами лучше всего по электронной почте, которую вы найдете на веб-сайте книги (<http://www.tmplbook.com>); однако, прежде чем посылать сообщение, удостоверьтесь, пожалуйста, что найденная вами неточность отсутствует в списке опечаток.

Заранее благодарим вас за сотрудничество.

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш веб-сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>