

*Посвящается Бет, благодаря которой стало возможным  
написание этой и всех остальных книг.*

*Посвящается Дэниелю и Анне,  
для которых возможно практически все.*

Стэнли Б. Липпман

*Посвящается Марку и маме,  
за их безграничную любовь и поддержку.*

Жози Лажойе

*Посвящается Энди, научившему меня программированию  
и многому другому.*

Барбара Му

## ВВЕДЕНИЕ


Благодаря предыдущим изданиям книги язык C++ изучило множество программистов. За истекшее время язык C++ претерпел существенные усовершенствования, а основное внимание сообщества программистов переместилось главным образом с эффективности использования аппаратных средств к эффективности программирования.

В 2011 году комитет по стандартам C++ выпустил новую основную версию стандарта ISO C++. Этот пересмотренный стандарт является последним этапом развития языка C++, его основное внимание уделено эффективности программирования. Основные задачи нового стандарта таковы.

- Сделать язык более единообразным, упростить его преподавание и изучение.
- Упростить, обезопасить и повысить эффективность использования стандартных библиотек.
- Облегчить написание эффективных абстракций и библиотек.

Это издание книги полностью пересмотрено так, чтобы использовать последний стандарт языка. Просмотрев раздел “Новые средства C++11” после оглавления, вы можете получить представление о том, насколько сильно новый стандарт повлиял на язык C++. Там перечислены только те разделы, в которых рассматривается новый материал.

Некоторые из нововведений в новом стандарте, такие как ключевое слово `auto` для вывода типов, весьма распространены. Эти средства существенно облегчают чтение кода в данном издании и делают его понятней. Программисты, конечно, могут игнорировать те средства, которые облегчают концентрацию на том, что программа призвана делать. Другие новшества, такие как интеллектуальные указатели и контейнеры с поддержкой перемещения, позволяют писать более сложные классы без необходимости справляться со сложностями управления ресурсами. В результате мы можем начать изучение создания собственных классов намного раньше, чем в предыдущем издании. Мы (и вы) больше не должны волноваться о большинстве деталей, которые стояли на нашем пути в предыдущем стандарте.

 Этой пиктограммой отмечены места, в которых рассматриваются средства, определенные новым стандартом. Надеемся, что читатели, которые уже знакомы с ядром языка C++, найдут эти отметки полезными при решении, на чем сосредоточить внимание. Мы также ожидаем, что эти пиктограммы помогут объяснить сообщения об ошибках тех компиляторов, которые могут еще не поддерживать все новые средства. Хотя

практически все примеры этой книги были откомпилированы на текущем выпуске компилятора GNU, мы понимаем, что у некоторых читателей еще не будет новейшего компилятора. Даже при том, что по последнему стандарту было добавлено множество возможностей, базовый язык остается неизменным и формирует основной объем материала, который мы рассматриваем.

## Для кого написана эта книга

Можно считать, что современный язык C++ состоит из трех частей.

- Низкоуровневый язык, большая часть которого унаследована от языка C.
- Дополнительные возможности языка, позволяющие определять собственные типы данных, организовать крупномасштабные программы и системы.
- Стандартная библиотека, использующая эти дополнительные возможности для обеспечения набора структур данных и алгоритмов.

В большинстве книг язык C++ представлен в порядке его развития. Сначала они знакомят с частью C в языке C++, а в конце книги представляются более абстрактные средства C++ как дополнительные возможности. У этого подхода есть две проблемы: читатели могут увязнуть в подробностях, унаследованных от низкоуровневого программирования, и сдаться. Те же, кто будет упорствовать в изучении, наживут плохие привычки, от которых впоследствии придется избавляться.

Мы придерживаемся противоположного подхода: с самого начала используем средства, которые позволяют программистам игнорировать детали, унаследованные от низкоуровневого программирования. Например, мы вводим и используем библиотечные типы `string` и `vector` наряду со встроенными цифровыми типами и массивами. Программы, которые используют эти библиотечные типы, проще писать, проще понимать, и ошибок в них много меньше.

Слишком часто библиотеки преподносят как “дополнительную” тему. Вместо того чтобы использовать библиотеки, во многих книгах используют низкоуровневые способы программирования с использованием указателей на символьные массивы и динамического управления памятью. Заставить правильно работать программы, которые используют эти низкоуровневые подходы, куда труднее, чем написать соответствующий код C++, используя библиотеку.

Повсюду в этой книге мы демонстрируем хороший стиль программирования: мы хотим помочь вам выработать хорошие привычки сразу и избежать борьбы с плохими привычками впоследствии, когда вы получите более сложные навыки. Мы подчеркиваем особенно сложные моменты и предупреждаем о наиболее распространенных заблуждениях и проблемах.

Мы также объясняем причины, по которым были введены те или иные правила, а не просто принимаем их как данность. Мы полагаем, что понимание причин поможет читателю быстрее овладеть возможностями языка.

Хотя для изучения этой книги знание языка С необязательно, мы подразумеваем, что вы знакомы с программированием достаточно, чтобы написать, откомпилировать и запустить программу хотя бы на одном из современных языков. В частности, мы подразумеваем, что вы умеете использовать переменные, создавать и вызывать функции, а также использовать компилятор.

## Изменения в пятом издании

Нововведением этого издания являются пиктограммы на полях, призванные помочь читателю. Язык С++ обширен, он предоставляет возможности для решения разнообразных специфических проблем программирования. Некоторые из этих возможностей весьма важны для больших групп разработчиков, но маловажны для малых проектов. В результате не каждому программисту следует знать каждую деталь каждого средства. Мы добавили эти пиктограммы, чтобы помочь читателю узнать, какие элементы могут быть изучены позже, а какие темы являются насущными.



Разделы, рассматривающие основные принципы языка, отмечены изображением человека, читающего книгу. Темы, затронутые в этих разделах, являются базовой частью языка. Все эти разделы следует прочитать и понять.



Мы также отметили те разделы, которые затрагивают дополнительные или специальные темы. Эти разделы можно пропустить или только просмотреть при первом чтении. Мы отметили такие разделы стопкой книг, указав, что на этом месте вы можете спокойно отложить книгу. Вероятно, имеет смысл просмотреть такие разделы и узнать, какие возможности существуют. Тем не менее нет никакой причины тратить время на изучение этих тем, пока вам фактически не придется использовать в своих программах описанное средство.



Особенно сложные концепции выделены пиктограммой с изображением лупы. Надеемся, что читатели уделят время, чтобы хорошо усвоить материал, представленный в таких разделах.

Еще одна помощь читателю этой книги — обширное употребление перекрестных ссылок. Мы надеемся, что эти ссылки облегчат читателю переход в середину книги и возвращение назад к прежнему материалу, на который ссылаются более поздние примеры.

Но что остается неизменным в книге, так это четкое и ясное, корректное и полное руководство по языку С++. Мы излагаем язык, представляя наборы все более и более сложных примеров, которые объясняют его средства и демонстрируют способы наилучшего использования С++.

## Структура книги

Мы начинаем с рассмотрения основ языка и библиотеки в частях I и II. Эти части содержат достаточно материала, чтобы позволить читателю писать работоспособные программы. Большинство программистов C++ должны знать все, что описано в этих частях.

Кроме обучения основам языка C++, материал частей I и II служит и другой важной цели: при использовании абстрактных средств, определенных библиотекой, вы научитесь использовать методики высокоуровневого программирования. Библиотечные средства сами являются абстрактными типами данных, которые обычно пишут на языке C++. Библиотека может быть создана с использованием тех же средств построения класса, которые доступны для любого программиста C++. Наш опыт в обучении языку C++ свидетельствует о том, что, если читатели с самого начала используют хорошо разработанные абстрактные типы, то впоследствии им проще понять, как создавать собственные типы.

Только после полного освоения основ использования библиотеки (и написания разных абстрактных программ при помощи библиотеки) мы переходим к тем средствам языка C++, которые позволяют писать собственные абстракции. В частях III и IV главное внимание уделяется написанию абстракции в форме классов. В части III рассматриваются общие принципы, а в части IV — специализированные средства.

В части III мы рассматриваем проблемы управления копированием, а также другие способы создания классов, которые так же удобны, как и встроенные типы. Классы — это основа объектно-ориентированного и обобщенного программирования, которое также будет рассмотрено в части III. Книга заканчивается частью IV, рассматривающей средства, обычно используемые в больших и сложных системах. В приложении A приведено краткое описание библиотечных алгоритмов.

## Соглашения, принятые в книге

Каждая глава завершается резюме и словарем терминов. Читатели могут использовать эти разделы как контрольный список: если вы не понимаете термин, следует повторно изучить соответствующую часть главы.

Здесь используются соглашения, общепринятые в компьютерной литературе.

- Новые термины в тексте выделяются *курсивом*. Чтобы привлечь внимание читателя на отдельные фрагменты текста, также применяется *курсив*.
- Текст программ, функций, переменных, URL веб-страниц и другой код представлен моноширинным шрифтом.
- Все, что придется вводить с клавиатуры, выделено **полужирным моноширинным** шрифтом.

- Знакоместо в описаниях синтаксиса выделено курсивом. Это указывает на необходимость заменить знакоместо фактическим именем переменной, параметром или другим элементом, который должен находиться на этом месте. Например: `BINDSIZE=(максимальная ширина колонки) * (номер колонки)`.
- Пункты меню и названия диалоговых окон представлены следующим образом: Menu Option (Пункт меню).

## Примечание о компиляторах

На момент написания этой книги (июль 2012 года) поставщики компиляторов интенсивно работали, модифицируя свои компиляторы в соответствии с последним стандартом ISO. Чаще всего мы использовали компилятор GNU версии 4.7.0. В этой книге использовано лишь несколько средств, которые в этом компиляторе еще не реализованы: наследование конструкторов, квалификаторы ссылок для функций-членов и библиотека регулярных выражений.

## Благодарности

Мы очень благодарны за помощь в подготовке этого издания нынешним и прежним членам комитета по стандартизации: Дейв Абрахамс (Dave Abrahams), Энди Кёниг (Andy Koenig), Стефан Т. Лававей (Stephan T. Lavavej), Джейсон Меррилл (Jason Merrill), Джон Спайсер (John Spicer) и Герб Саттер (Herb Sutter). Они оказали нам неоценимую помощь в понимании некоторых нюансов нового стандарта. Мы также хотели бы поблагодарить многих других людей, которые работали над модификацией компилятора GNU и сделали стандарт реальностью.

Как и в предыдущих изданиях этой книги, мы хотели бы выразить отдельную благодарность Бьярне Страуструпу (Bjarne Stroustrup) за его неустанную работу над языком C++ и многолетнюю дружбу с авторами. Хотелось бы также поблагодарить Алекса Степанова (Alex Stepanov) за его объяснения по теме контейнеров и алгоритмов, составляющих ядро стандартной библиотеки. И наконец, сердечная благодарность членам комитета по стандарту C++ за их упорную многолетнюю работу по утверждению и усовершенствованию стандарта языка C++.

Авторы также выражают глубокую благодарность рецензентам, чьи комментарии, замечания и полезные советы помогли улучшить книгу. Спасибо Маршаллу Клоу (Marshall Clow), Джону Калбу (Jon Kalb), Невину Либеру (Nevin Liber), др. К. А. Тондо (Dr. C. L. Tondo), Дэвиду Вандеворду (Daveed Vandevoorde) и Стиву Виноски (Steve Vinoski).

Эта книга была набрана при помощи системы LaTeX и прилагаемых к ней пакетов. Авторы выражают глубокую благодарность членам сообщества LaTeX, сделавшим доступным такой мощный инструмент.

И наконец, благодарим сотрудников издательства Addison-Wesley, которые курировали процесс публикации этой книги: Питер Гордон (Peter Gordon) — наш редактор, который предложил пересмотреть эту книгу еще раз; Ким Бодихаймер (Kim Voedigheimer) контролировал график выполнения работ; Барбара Вуд (Barbara Wood) нашла множество наших ошибок на этапе редактирования, а Элизабет Райан (Elizabeth Ryan) снова помогала авторам на протяжении всего проекта.

## От издательства

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 интересно услышать и любые другие замечания, которые вам хотелось бы высказать авторам.

Мы ждем ваших комментариев. Вы можете прислать письмо по электронной почте или просто посетить наш веб-сервер, оставив на нем свои замечания. 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 подходящими для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш e-mail. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию следующих книг. Наши координаты:

E-mail: [info@williamspublishing.com](mailto:info@williamspublishing.com)

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

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

в Украине: 03150, Киев, а/я 152