

Эффективный и современный C++ — отличный инструмент для повышения вашего уровня как программиста на современном C++. Книга не только учит тому, как, когда и где эффективно использовать современный C++, но и *почему* делать это именно так. Вне всякого сомнения, эта книга Скотта Мейерса даст программистам гораздо лучшее понимание языка.

— Барт Вандвустин (*Bart Vandewoestyne*)
Инженер, исследователь и просто энтузиаст C++

Я люблю C++, он десятилетиями был моей рабочей лошадкой. А с новыми копытами эта лошадка стала еще сильнее и привлекательнее, чем я мог ранее себе представить. Но при больших изменениях всегда встает вопрос “Когда и как пользоваться всем этим богатством?” Как всегда, книга Скотта Мейерса компетентно и исчерпывающе отвечает на поставленный вопрос.

— Дамьен Уоткинс (*Damien Watkins*)
Руководитель группы программной инженерии в CSIRO

Отличное чтение для перехода к современному C++ — новинки языка C++11/14 описаны наряду с C++98, разделение содержимого книги на разделы позволяет легко найти интересующую тему, а в конце каждого раздела приведены итоговые рекомендации. Книга интересна и полезна для программистов на C++ всех уровней.

— Рейчел Ченг (*Rachel Cheng*)
F5 Networks

Если вы переходите с C++98/03 на C++11/14, вам нужна точная практическая информация, которую вам предоставляет Скотт Мейерс в книге *Эффективный и современный C++*. Если вы уже пишете код на C++11, то, вероятно, сталкивались с проблемами при использовании новых возможностей, которые легко решаются с помощью книги Скотта. В любом случае можно уверенно утверждать, что время, затраченное на чтение этой книги, не пропадет впустую.

— Роб Стюарт (*Rob Stewart*)
Член Boost Steering Committee (boost.org)

Об авторе

Скотт Мейерс — один из ведущих мировых экспертов по C++, широко востребованный как инструктор, консультант и докладчик на разных конференциях. Более чем 20 лет книги Скотта Мейерса серии *Эффективный C++* являются критерием уровня книг по программированию на C++. Скотт Мейерс имеет степень доктора философии (Ph.D.) в области компьютерных наук в Университете Брауна (Brown University). Его сайт находится по адресу aristeia.com.

Об изображении на обложке

На обложке книги *Эффективный и современный C++* изображен *розовошапочный пестрый голубь* (*Ptilinopus regina*). Еще одно имя этого вида голубя — *Свенсонов пестрый голубь* (Swainson's fruit dove). Он отличается ярким оперением: серые голова и грудь, оранжевый живот, беловатое горло, желто-оранжевый цвет радужки и серо-зеленые ноги.

Голубь распространен в равнинных лесах восточной Австралии, муссонных лесах северной части Австралии, а также на малых Зондских островах и Молуккских островах Индонезии. Рацион голубя состоит из различных фруктов наподобие инжира (который он поедает целиком), пальм и лоз. Еще одним источником пищи голубя является камфорный лавр, большое вечнозеленое дерево. Они питаются парами, небольшими стайками или поодиночке в тропических лесах, обычно утром или поздно вечером. Для питья они используют воду из листьев или росу, но не воду с земли.

Пестрый голубь в Новом Южном Уэльсе считается видом, находящимся под угрозой исчезновения из-за изменения среды обитания — лесозаготовок, очистки и улучшения земель, а также вырубки камфорного лавра без адекватной альтернативы.

Многие из животных, представленных на обложках O'Reilly, находятся под угрозой исчезновения. Все они имеют очень важное значение для нашего мира. Чтобы узнать больше о том, как вы можете помочь им, посетите сайт animals.oreilly.com.

Изображение для обложки взято из *Иллюстрированной истории природы* Вуда (Wood), из тома, посвященного птицам.

Дарле,
выдающемуся черному лабрадору-ретриверу

Благодарности

Я начал исследовать то, что тогда было известно как C++0x (зарождающийся C++11), в 2009 году. Я опубликовал множество вопросов в группе новостей Usenet `comp.std.c++` и благодарен членам этого сообщества (в особенности Дэниелу Крюглеру (Daniel Krugler)) за их очень полезные сообщения. В последние годы с вопросами по C++11 и C++14 я обращался к Stack Overflow, и я многим обязан этому сообществу за его помощь в понимании тонкостей современного C++.

В 2010 году я подготовил материалы для учебного курса по C++0x (в конечном итоге опубликованные как *Overview of the New C++*, Artima Publishing, 2010). На эти материалы, как и на мои знания, большое влияние оказала техническая проверка, выполненная Стивеном Т. Лававеем (Stephan T. Lavavej), Бернгардом Меркль (Bernhard Merkle), Стенли Фризенем (Stanley Friesen), Леором Зорманом (Leor Zolman), Хендриком Шобером (Hendrik Schober) и Энтони Вильямсом (Anthony Williams). Без их помощи я не смог бы довести мою книгу до конца. Кстати, ее название было предложено несколькими читателями в ответ на мое сообщение в блоге от 18 февраля 2014 года (“Помогите мне назвать мою книгу”), и Андрей Александреску (Andrei Alexandrescu) (автор книги *Modern C++ Design*¹, Addison-Wesley, 2001) был достаточно великодушен, чтобы не считать это название незаконным вторжением на его территорию.

Я не могу указать источники всей информации в этой книге, но некоторые из них непосредственно повлияли на мою книгу. Применение в разделе 1.4 неопределенного шаблона для получения информации о типе от компилятора было предложено Стивеном Т. Лававеем, а Мэтт П. Дзюбински (Matt P. Dziubinski) обратил мое внимание на Boost.TypeIndex. В разделе 2.1 пример `unsigned std::vector<int>::size_type` взят из статьи Андрея Карпова (Andrey Karpov) от 28 февраля 2010 года “*In what way can C++0x standard help you eliminate 64-bit errors*”. Пример `std::pair<std::string, int>/std::pair<const std::string, int>` в том же разделе книги взят из сообщения “*STL11: Magic && Secrets*” Стивена Т. Лававея на *Going Native 2012*. Раздел 2.2 появился благодаря статье Герба Саттера (Herb Sutter) “*GotW*

¹ Русский перевод — Александреску Андрей. *Современное проектирование на C++*. — М.: Издательский дом “Вильямс”, 2002.

#94 Solution: AAA Style (Almost Always Auto)” от 12 августа 2013 года, а раздел 3.3 — благодаря сообщению в блоге Мартинго Фернандеса (Martinho Fernandes) от 27 мая 2012 года — “Handling dependent names”. Пример в разделе 3.6 демонстрирует перегрузку квалификаторов ссылок, основанную на ответе Кейси (Casey) на вопрос “What’s a use case for overloading member functions on reference qualifiers?” в сообществе Stack Overflow от 14 января 2014 года. В разделе 3.9 описание расширенной поддержки constexpr-функций в C++14 включает информацию, которую я получил от Рейна Халберсма (Rein Halbersma). Раздел 3.10 основан на презентации Герба Саттера на конференции C++ and Beyond 2012 под названием “You don’t know const и mutable”. Совет в разделе 4.1, гласящий, что фабричная функция должна возвращать std::unique_ptr, основан на статье Герба Саттера “GotW# 90 Solution: Factories” от 13 мая 2013 года. fastLoadWidget в разделе 4.2 получен из презентации Герба Саттера “My Favorite C++ 10-Liner” на конференции Going Native 2013. В моем описании std::unique_ptr и неполных типов в разделе 4.5 использованы статья Герба Саттера от 27 ноября 2011 года “GotW #100: Compilation Firewalls”, а также ответ Говарда Хиннанта (Howard Hinnant) от 22 мая 2011 года на вопрос в Stack Overflow “Is std::unique_ptr<T> required to know the full definition of T?” Дополнительный пример Matrix в разделе 5.3 основан на письме Дэвида Абрахамса (David Abrahams). Комментарий Джо Аргонна (Joe Argonne) от 8 декабря 2012 года к материалу из блога “Another alternative to lambda move capture” от 30 ноября 2013 года стал источником для описанного в разделе 6.2 подхода к имитации инициализации на основе std::bind в C++11. Пояснения в разделе 7.3 проблемы с неявным отключением в деструкторе std::thread взяты из статьи Ганса Бехма (Hans-J. Boehm) “N2802: A plea to reconsider detach-on-destruction for thread objects” от 4 декабря 2008 года. Раздел 8.1 появился благодаря обсуждению материала в блоге Дэвида Абрахамса “Want speed? Pass by value” от 15 августа 2009 года. Идея о том, что типы, предназначенные только для перемещения, должны рассматриваться отдельно, взята у Мэттью Фьораванте (Matthew Fioravante), в то время как анализ копирования на основе присваивания взят из комментариев Говарда Хиннанта (Howard Hinnant). В разделе 8.2 Стивен Т. Лававей и Говард Хиннант помогли мне понять вопросы, связанные с относительной производительностью функций размещения и вставки, а Майкл Винтерберг (Michael Winterberg) привлек мое внимание к тому, как размещение может приводить к утечке ресурсов. (Майкл, в свою очередь, называет своим источником презентацию “C++ Seasoning” Шона Парента (Sean Parent) на конференции Going Native 2013. Майкл также указал, что функции размещения используют непосредственную инициализацию, в то время как функции вставки используют копирующую инициализацию.)

Проверка черновиков технической книги является длительной и критичной, но совершенно необходимой работой, и мне повезло, что так много людей были готовы за нее взяться. Черновики этой книги были официально просмотрены такими специалистами, как Кассио Нери (Cassio Neri), Нейт Кель (Nate Kohl), Герхард Крейцер (Gerhard Kreuzer), Леон Золман (Leor Zolman), Барт Вандевоистин (Bart Vandewoestyne), Стивен Т. Лававей (Stephan T. Lavavej), Невин Либер (Nevin “:-)” Liber), Речел Ченг (Rachel Cheng), Роб Стюарт (Rob Stewart), Боб Стигалл (Bob Steagall), Дамьен Уоткинс (Damien Watkins), Брэдли Нидхам (Bradley E. Needham), Рейнер Гримм (Rainer Grimm), Фредрик Винклер (Fredrik Winkler), Джонатан Уокели (Jonathan Wakely), Герб Саттер (Herb Sutter), Андрей

Александреску (Andrei Alexandrescu), Эрик Ниблер (Eric Niebler), Томас Беккер (Thomas Becker), Роджер Опп (Roger Orr), Энтони Вильямс (Anthony Williams), Майкл Винтерберг (Michael Winterberg), Бенджамин Хахли (Benjamin Huchley), Том Кирби-Грин (Tom Kirby-Green), Алексей Никитин (Alexey A. Nikitin), Вильям Дилтрай (William Dealtry), Хуберт Мэттьюс (Hubert Matthews) и Томаш Каминьски (Tomasz Kamiński). Я также получил отзывы ряда читателей с помощью сервисов O'Reilly's Early Release EBooks и Safari Books Online's Rough Cuts, посредством комментариев в моем блоге (*The View from Aristeia*) и электронной почтой. Я благодарен каждому, кто высказал свои замечания. Эта книга получилась *гораздо* лучше, чем она была бы без этой помощи. В особенности я признателен Стивену Т. Лававею и Робу Стюарту, чьи чрезвычайно подробные и всеобъемлющие замечания заставили меня забеспокоиться: кто из нас потратил больше сил и времени на эту книгу — я или они? Моя особая благодарность — Леору Золману (Leor Zolman), который не только просмотрел рукопись, но и дважды проверил все приведенные в ней примеры кода.

Черновики цифровых версий книги были подготовлены Герхардом Крейцером (Gerhard Kreuzer), Эмиром Вильямсом (Emyr Williams) и Брэдли Нидхэмом (Bradley E. Needham).

Мое решение ограничить длину строки кода 64 символами (максимум для правильного отображения на печати, а также на различных цифровых устройствах при разной ориентации и конфигурации шрифтов) было основано на данных, предоставленных Майклом Махером (Michael Maher).

С момента первой публикации я исправил ряд ошибок и внес некоторые усовершенствования, предложенные такими читателями, как Костас Влахавас (Kostas Vlahavas), Даниэль Алонсо Алеман (Daniel Alonso Alemany), Такатоши Кондо (Takatoshi Kondo), Бартек Сургот (Bartek Szurgot), Тайлер Брок (Tyler Brock), Джай Ципник (Jay Zipnick), Барри Ревзин (Barry Revzin), Роберт Маккейб (Robert McCabe), Оливер Брунс (Oliver Bruns), Фабрис Ферино (Fabrice Ferino), Дэнез Джонитис (Dainis Jonitis), Петр Валашек (Petr Valasek) и Барт Вандевоистин (Bart Vandewoestyne). Большое спасибо всем им за помощь в повышении точности и ясности изложенного материала.

Эшли Морган Вильямс (Ashley Morgan Williams) готовила отличные обеды у себя в Lake Oswego Pizzicato. Им (и ей) нет равных.

И более двадцати лет моя жена, Нэнси Л. Урбано (Nancy L. Urbano), как обычно во время моей работы над новой книгой, терпит мою раздражительность и оказывает мне всемерную поддержку. В ходе написания книги постоянным напоминанием о том, что за пределами клавиатуры есть другая жизнь, служила мне наша собака Дарла.