

Эффективное использование C++

«У каждого профессионала C++ должна быть книга «Эффективное использование C++». Это издание – обязательно к прочтению для каждого, кто собирается выполнять серьезную разработку на C++. Если вы не читали эту книгу и думаете, что знаете все о C++, подумайте еще раз.»

- **Стив Ширрипа**, инженер по программному обеспечению, Google.

«C++ и сообщество C++ сильно выросло за последние пятнадцать лет, и третья редакция «Эффективного использования C++» отражает это. Ясный и точный стиль этой книги – свидетельство глубоких знаний Скотта Мейерса и его исключительной способности донести их до читателя.»

- **Герхард Крейзер**, инженер по разработкам и исследованиям, Siemens AG.

Первые две редакции «Эффективного использования C++» были приняты сотнями тысяч программистов по всему миру. Причина проста: практический подход Скотта Мейерса к C++ и описанию правил, которыми руководствуются эксперты – то есть того, что они всегда делают и чего избегают, чтобы создать ясный, корректный и эффективный код.

Эта книга организована в виде 55 советов, помогающих написать лучший код на C++. Все эти руководства сопровождаются конкретными примерами. В третьем издании содержится более половина нового материала - включая дополнения, касающиеся управления ресурсами и использования шаблонов. Темы из второго издания были серьезно переработаны с тем, чтобы отобразить современные подходы к проектированию, в том числе исключения, паттерны и многопоточность.

Важные компоненты, включенные в книгу «Эффективное использование C++»:

- Экспертное руководство по проектированию эффективных классов, функций, шаблонов и иерархий наследования.
- Приложение новой функциональности стандартной библиотеки TR1, в сравнении с существующими компонентами стандартной библиотеки.
- Тщательный анализ различий между C++ и другими языками (таким, как Java, C#, C), помогающее разработчикам, имеющим опыт работы на других языках, привыкнуть к методике работы с C++.

Скотт Мейерс – один из ведущих мировых экспертов в области разработки программного обеспечения на C++. Он – автор серии бестселлеров («Эффективное использование C++», «Наиболее эффективное использование C++» и «Effective STL»), автор новаторского CD «Effective C++», редактор-консультант серии «Effective Software Development», издаваемой Addison-Wesley, а также член экспертной комиссии журнала Software Development. Мейерс получил титул доктора философии в области компьютерных наук в Brown University в 1993 году. Его web-сайт: www.aristeia.com.

Интернет-магазин:

www.abook.ru

Книга - почтой:

Россия, 123242, Москва, а/я 20

e-mail: post@abook.ru

Оптовая продажа:

«Альянс-книга»

Тел./факс: (095)258-9195

e-mail: abook@abook.ru



Эффективное использование C++

С. Мейерс

Эффективное использование C++

55 верных советов
улучшить структуру и код
ваших программ

Скотт Мейерс



Третье издание



Профессиональная серия от Addison-Wesley

Скотт Мэйерс

Эффективное использование C++

Третье издание

**55 верных советов улучшить
структуру и код ваших программ**



Москва

УДК 004.4
ББК 32.973.26-018.2
М97

Мэйерс С.

М97 Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ – М.: ДМК Пресс. – 300 с.: ил.

ISBN 5-94074-304-8

Эта книга представляет собой перевод третьего издания американского бестселлера Effective С++ и является руководством по грамотному использованию языка С++. Она поможет сделать ваши программы более понятными, простыми в сопровождении и эффективными. Помимо материала, описывающего общую стратегию проектирования, книга включает в себя главы по программированию с применением шаблонов и по управлению ресурсами, а также множество советов, которые позволят усовершенствовать ваши программы и сделать работу более интересной и творческой. Книга также включает новый материал по принципам обработки исключений, паттернам проектирования и библиотечным средствам.

Издание ориентировано на программистов, знакомых с основами С++ и имеющих навыки его практического применения.

УДК 004.4
ББК 32.973.26-018.2

Original English language edition published by Pearson Education, Inc. Copyright © 2005 by Pearson Education, Inc. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 0-321-33487-6 (англ.)
ISBN 5-98453-304-8 (АйТи)

© Pearson Education, Inc.
© Перевод на русский язык, оформление,
издание. ДМК-Пресс



Содержание

Благодарности	11
Предисловие	15
Введение	17
Терминология	18
Соглашения об именах	23
Многопоточность	24
Библиотеки TR1 и Boost	24
Глава 1. Приучайтесь к C++	26
Правило 1: Относитесь к C++ как к конгломерату языков	26
Правило 2: Предпочитайте const, enum и inline использованию #define	28
Правило 3: Везде, где только можно, используйте const	31
Константные функции-члены	33
Как избежать дублирования в константных и неконстантных функциях-членах	37
Правило 4: Прежде чем использовать объекты, убедитесь, что они инициализированы	40
Глава 2. Конструкторы, деструкторы и операторы присваивания	47
Правило 5: Какие функции C++ создает и вызывает молча	47
Правило 6: Явно запрещайте компилятору генерировать функции, которые вам не нужны	50
Правило 7: Объявляйте деструкторы виртуальными в полиморфном базовом классе	52
Правило 8: Не позволяйте исключениям покидать деструкторы	56
Правило 9: Никогда не вызывайте виртуальные функции в конструкторе или деструкторе	60

Правило 10: Операторы присваивания должны возвращать ссылку на *this	64
Правило 11: В operator= осуществляйте проверку на присваивание самому себе	65
Правило 12: Копируйте все части объекта	68
Глава 3. Управление ресурсами	72
Правило 13: Используйте объекты для управления ресурсами	72
Правило 14: Тщательно продумывайте поведение при копировании классов, управляющих ресурсами	76
Правило 15: Предоставляйте доступ к самим ресурсам из управляющих ими классов	79
Правило 16: Используйте одинаковые формы new и delete	83
Правило 17: Помещение в «интеллектуальный» указатель объекта, выделенного с помощью new, лучше располагать в отдельном предложении	85
Глава 4. Проектирование программ и объявления	87
Правило 18: Проектируйте интерфейсы так, что их легко было использовать правильно и трудно – неправильно	87
Правило 19: Рассматривайте проектирование класса как проектирование типа	92
Правило 20: Предпочитайте передачу по ссылке на const передаче по значению	94
Правило 21: Не пытайтесь вернуть ссылку, когда должны вернуть объект	98
Правило 22: Объявляйте данные-члены закрытыми	102
Правило 23: Предпочитайте функциям-членам функции, не являющиеся ни членами, ни друзьями класса	105
Правило 24: Объявляйте функции, не являющиеся членами, когда преобразование типов должно быть применимо ко всем параметрам	109
Правило 25: Подумайте о поддержке функции swap, не возбуждающей исключений	112
Глава 5. Реализация	119
Правило 26: Откладывайте определение переменных настолько возможно	119
Правило 27: Не злоупотребляйте приведением типов	122

Правило 28: Избегайте возвращения «дескрипторов» внутренних данных	128
Правило 29: Стремитесь, чтобы программа была безопасна относительно исключений	132
Правило 30: Тщательно обдумывайте использование встроенных функций	139
Правило 31: Уменьшайте зависимости файлов при компиляции	144

Глава 6. Наследование и объектно-ориентированное проектирование 153

Правило 32: Используйте открытое наследование для моделирования отношения «является»	153
Правило 33: Не скрывайте унаследованные имена	159
Правило 34: Различайте наследование интерфейса и наследование реализации	164
Правило 35: Рассмотрите альтернативы виртуальным функциям	171
Реализация паттерна «Шаблонный метод» с помощью идиомы невиртуального интерфейса	172
Реализация паттерна «Стратегия» посредством указателей на функции	173
Реализация паттерна «Стратегия» посредством класса <code>tr::function</code> ..	175
«Классический» паттерн «Стратегия»	177
Резюме	178
Правило 36: Никогда не переопределяйте наследуемые невиртуальные функции	179
Правило 37: Никогда не переопределяйте наследуемое значение аргумента функции по умолчанию	181
Правило 38: Моделируйте отношение «содержит» или «реализуется посредством» с помощью композиции	185
Правило 39: Продумывайте подход к использованию закрытого наследования	188
Правило 40: Продумывайте подход к использованию множественного наследования	193

Глава 7. Шаблоны и обобщенное программирование 200

Правило 41: Разберитесь в том, что такое неявные интерфейсы и полиморфизм на этапе компиляции	200
--	-----

Правило 42: Усвойте оба значения ключевого слова <code>typename</code> ...	204
Правило 43: Необходимо знать, как обращаться к именам в шаблонных базовых классах	207
Правило 44: Размещайте независимый от параметров код вне шаблонов	212
Правило 45: Разрабатывайте шаблоны функций-членов так, чтобы они принимали «все совместимые типы»	217
Правило 46: Определяйте внутри шаблонов функции, не являющиеся членами, когда желательны преобразования типа	221
Правило 47: Используйте классы-характеристики для предоставления информации о типах	225
Правило 48: Изучите метапрограммирование шаблонов	231
Глава 8. Настройка <code>new</code> и <code>delete</code>	237
Правило 49: Разберитесь в поведении обработчика <code>new</code>	238
Правило 50: Когда имеет смысл заменять <code>new</code> и <code>delete</code>	244
Правило 51: Придерживайтесь принятых соглашений при написании <code>new</code> и <code>delete</code>	249
Правило 52: Если вы написали оператор <code>new</code> с размещением, напишите и соответствующий оператор <code>delete</code>	252
Глава 9. Разное	258
Правило 53: Обращайте внимание на предупреждения компилятора	258
Правило 54: Ознакомьтесь со стандартной библиотекой, включая TR1	259
Правило 55: Познакомьтесь с Boost	264
Приложение А. За пределами «Эффективного использования C++»	268
Приложение В. Соответствие правил во втором и третьем изданиях	272

Предисловие

Я написал первый вариант книги «Эффективное использование C++» в 1991 г. Когда в 1997 г. настало время для второго издания, я существенно обновил материал, но, не желая смутить читателей, знакомых с первым изданием, постарался сохранить существующую структуру: 48 из оригинальных 50 правил остались по сути неизменными. Если сравнивать книгу с домом, то второе издание было похоже на косметический ремонт – переклейку обоев, окраску в другие цвета и замену осветительных приборов.

В третьем издании я решился на гораздо большее. (Был момент, когда хотелось перестроить заново все, начиная с фундамента.) Язык C++ с 1991 года изменился очень сильно, и цели этой книги – выявить все наиболее важное и представить в виде компактного сборника рекомендаций – уже не отвечал набору правил, сформулированных 15 лет назад. В 1991 году было резонно предполагать, что на язык C++ переходят программисты, имеющие опыт работы с С. Теперь же к ним с равной вероятностью можно отнести и тех, кто раньше писал на языках Java или C#. В 1991 году наследование и объектно-ориентированное программирование были чем-то новым для большинства программистов. Теперь же это – хорошо известные концепции, а областями, в разъяснении которых люди нуждаются в большей степени, стали исключения, шаблоны и обобщенное программирование теми. В 1991 году никто не слышал о паттернах проектирования. Теперь без их упоминания вообще трудно обсуждать программные системы. В 1991 году работа над формальным стандартом C++ только начиналась, теперь этому стандарту уже 8 лет, и ведется работа над следующей версией.

Чтобы учесть все эти изменения, я решил начать с чистого листа и спросил себя: «Какие советы стоит дать практикующим программистам C++ в 2005 году?» В результате и появился набор правил, включенных в новое издание. Эта книга включает новые главы по программированию с применением шаблонов и управлению ресурсами. Фактически шаблоны красной нитью проходят через весь текст, поскольку мало что в современном C++ обходится без них. В книгу включен также материал по программированию при наличии исключений, паттернам проектирования и новым библиотечным средствам, описанным в документе «Technical Report 1» (TR1) (этот документ рассматривается в правиле 54). Известно также тот факт, что подходы и методы, которые хорошо работают в однопоточных системах, могут быть неприменимы к многопоточным. Больше половины материалов этого издания – новые темы. Однако значительная часть основополагающей информации из второго издания остается актуальной, поэтому я нашел способ в той или иной форме повторить ее (соответствие между правилами второго и третьего изданий вы найдете в приложении В).

Я старался по мере сил сделать эту книгу максимально полезной, но, конечно, не считаю ее безупречной. Если вам покажется, что какие-то из приведенных правил нельзя считать универсально применимыми, что есть лучший способ решить сформулированную задачу либо что обсуждение некоторых технических вопросов недостаточно ясно, неполно, может ввести в заблуждение, пожалуйста, сообщите мне. Если вы обнаружите ошибки любого рода – технические, грамматические, типографские, – любые, – напишите мне и об этом. При выпуске следующего тиража я с удовольствием упомяну каждого, кто обратит мое внимание на какую-то проблему.

Несмотря на то что в новом издании количество правил увеличено до 55, конечно, нельзя сказать, что рассмотрены все и всяческие вопросы. Но сформулировать набор таких правил, которых следует придерживаться почти во всех приложениях почти всегда, труднее, чем может показаться на первый взгляд. Если у вас есть предложения по поводу того, что стоило бы включить еще, я с удовольствием их рассмотрю.

Начиная с момента выхода в свет первого издания этой книги, я вел перечень изменений, в котором отражены исправления ошибок, уточнения и технические обновления. Он доступен на Web-странице «Effective C++ Errata» по адресу <http://aristeia.com/BookErrata/ec++3e-errata.html>. Если вы хотите получать уведомления при обновлении этого перечня, присоединяйтесь к моему списку рассылки. Я использую его для того, чтобы делать объявления, которые, вероятно, заинтересуют людей, следящих за моей профессиональной деятельностью. Подробности см. на <http://aristeia.com/MailingList>.

Скотт Дуглас Мэйерс
<http://aristeia.com/>

Стаффорд, Орегон, апрель 2005