Отзывы о третьей редакции *Эффективного использования C++*

Книга Скотта Мейерса *«Эффективное использование C++», третья редакция* – это концентрация опыта программирования – того опыта, который без нее достался бы вам дорогой ценой. Эта книга – великолепный источник, который я рекомендую всем, кто пишет на C++ профессионально.

Питер Дулимов, ME, инженер,
подразделение оценки и исследований NAVSYSCOM, Австралия

Третья редакция остается лучшей книгой, посвященной тому, как сложить вместе все части C++ для создания эффективных и внутренне целостных программ. Если вы претендуете на то, чтобы быть программистом C++, то должны ее прочитать.

Эрик Наглер, консультант, преподаватель и автор «Изучая C++»

Первая редакция этой книги была одной из небольшого (весьма небольшого) числа книг, благодаря которым я ощутимо повысил свой уровень как профессионального разработчика программного обеспечения. Как и другие книги из этого ряда, она оказалась практичной и легкой для чтения, но при этом содержала множество важных советов. *«Эффективное использование C++», третья редакция*, продолжает эту традицию. C++ – очень мощный язык программирования. Если C дает веревку, по которой можно забраться на вершину горы, то C++ – это целый магазин, в котором самые разные люди готовы помочь вам завязать на этой веревке узлы. Овладение материалом, приведенным в этой книге, определенно повысит вашу способность эффективно использовать C++ и не умереть при этом от напряжения.

Джек В. Ривес, исполнительный директор Bleading Edge Software Technologies

Каждый новый разработчик, который приходит в мою команду, сразу получает задание – прочесть эту книгу.

Майкл Ланцетта, ведущий инженер по программному обеспечению

Я прочитал первую редакцию «Эффективного использования C++» около 9 лет назад, и эта книга сразу стала одной из моих любимых книг по C++. На мой взгляд, третье издание «Эффективного использования C++» остается обязательным к прочтению для всех, кто желает эффективно программировать на C++. Мы будем жить в лучшем мире, если программисты C++ прочтут эту книгу прежде, чем написать первую строку профессионального кода.

Дэнни Раббани, инженер по программному обеспечению

Первое издание «Эффективного использования C++» Скотта Мейерса попалось мне, когда я был рядовым программистом и напряженно старался как можно лучше выполнить порученную работу. И это было спасением! Я обнаружил, что советы Мейерса практически полезны и эффективны, что они на 100% реализуют то, что обещают. Третья редакция помогает в практическом применении C++ при работе над современными серьезными программными проектами, предоставляя информацию о самых новых средствах и возможностях языка. Я с удовольствием обнаружил, что могу найти много нового и интересного для себя в третьем издании книги, которую, как мне казалось, знаю очень хорошо.

Майкл Топик, технический программный менеджер

Это авторитетное руководство от Скотта Мейерса, гуру C++, предназначенное для каждого, кто хочет применять C++ безопасно и эффективно, или же переходит к C++ от любого другого объектно-ориентированного языка. Эта книга содержит ценную информацию, изложенную в ясном, сжатом, занимательном и проницательном стиле.

Сиддхартха Каран Сингх, разработчик программного обеспечения

Скотт Мэйерс

Эффективноеиспользование C++

Scott Meyers

Effective C++

Third Edition

55 Specific Ways to Improve Your Programs and Designs

ADDISON-WESLEY

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Скотт Мэйерс

Эффективное использование C++

Третье издание

55 верных советов улучшить структуру и код ваших программ

ДМК ПРЕСС
ИЗДАТЕЛЬСТВО
Москва, 2006

УДК 004.4
ББК 32.973.26-018.2
М97

Мэйерс С.
М97 Эффективное использование C++. 55 верных способов улучшить структуру и код ваших прграмм – М.: ДМК Пресс, 2006. – 300 с.: ил.

ISBN 5-94074-304-8

Эта книга представляет собой перевод третьего издания американского бестселлера Effective C++ и является руководством по грамотному использованию языка C++. 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 C++ и имеющих навыки его практического применения.

УДК 004.4
ББК 32.973.26-018.2

Original Eglish language edition publihed 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)

© 2005 Pearson Education, Inc.
© Перевод на русский язык, оформление, издание. ДМК-Пресс, 2006



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 компилфятору 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 насколько 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 tr::function .. 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 typename ... 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 new и delete ... 237

Правило 49: Разберитесь в поведении обработчика new ... 238
Правило 50: Когда имеет смысл заменять new и delete ... 244
Правило 51: Придерживайтесь принятых соглашений при написании new и delete ... 249
Правило 52: Если вы написали оператор new с размещением, напишите и соответствующий оператор delete ... 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 A. За пределами «Эффективного использования C++» ... 268

Приложение B. Соответствие правил во втором и третьем изданиях ... 272