Мэтью Уилсон

РАСШИРЕНИЕ БИБЛИОТЕКИ STL ДЛЯ C++

Наборы и итераторы

Мэтью Уилсон

EXTENDED STL, VOLUME 1

Collections and Iterators

Addison-Wesley

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

Мэтью Уилсон

РАСШИРЕНИЕ БИБЛИОТЕКИ STL ДЛЯ C++

Наборы и итераторы





Москва, Санкт-Петербург, 2008

УДК 681.3.068+800.92С++
ББК 32.973.26-018.1
У35

Уилсон М.

Расширение библиотеки STL для C++. Наборы и итераторы: Пер. с англ. Слинкина А. А. – М.: ДМК Пресс, СПб, БХВ-Петербург, 2008. – 608 с.: ил. + CD-ROM

ISBN 978-5-94074-442-9 («ДМК Пресс»)

ISBN 978-5-9775-0196-5 («БХВ-Петербург»)

В книге известный специалист по языку C++ Мэтью Уилсон демонстрирует, как выйти за пределы стандарта C++ и расширить стандартную библиотеку шаблонов, применив лежащие в ее основе принципы к различным API и нестандартным наборам, чтобы получить более эффективные, выразительные, гибкие и надежные программы.

Автор описывает передовые приемы, которые помогут вам в совершенстве овладеть двумя важными темами: адаптация API библиотек и операционной системы к STL-совместимым наборам и определение нетривиальных адаптеров итераторов. Это даст вам возможность в полной мере реализовать заложенные в STL возможности для написания эффективных и выразительных программ. На реальных примерах Уилсон иллюстрирует ряд важных концепций и технических приемов, позволяющих расширить библиотеку STL в таких направлениях, о которых ее создатели даже не думали, в том числе: наборы, категории ссылок на элементы, порча итераторов извне и выводимая адаптация интерфейса.

Эта книга станет неоценимым подспорьем для любого программиста на C++, хотя бы в минимальной степени знакомого с STL.

На прилагаемом компакт-диске находится обширная коллекция открытых библиотек, созданных автором. Также включено несколько тестовых проектов и три дополнительных главы.

УДК 681.3.068+800.92С++
ББК 32.973.26-018.1

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. RUSSIAN language edition published by DMK PUBLISHERS, Copyright © 2007.

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 978-0-321-30550-7 (англ.)
ISBN 978-5-94074-442-9 («ДМК Пресс»)
ISBN 978-5-9775-0196-5 («БХВ-Петербург»)

Copyright © 2007, Pearson Education, Inc.
© Перевод на русский язык, оформление ДМК Пресс, 2008
© Издание, БХВ-Петербург, 2008

Содержание

Предисловие 22

Цели 22
Предмет обсуждения 23
Организация книги 24
Дополнительные материалы 25

Благодарности 26

Об авторе 28

Пролог 29

Дихотомия объекта исследования 29
Принципы программирования в системе UNIX 30
Семь признаков успешных библиотек на C++ 31
Эффективность 31
Понятность и прозрачность 33
Выразительные возможности 34
Надежность 36
Гибкость 37
Модульность 38
Переносимость 38
Поиск компромиссов: довольство тем, что имеешь, диалектизм и идиомы 39
Примеры библиотек 40
STLSoft 41
Подпроекты STLSoft 41
Boost 43
Open-RJ 43
Pantheios 43
recls 44

Типографские соглашения 45

Шрифты 45
. . . сравни 45
Предварительное вычисление концевого итератора 46
Квалификация типа вложенного класса 46

NULL ... 47
Имена параметров шаблона ... 47
Имена типов-членов и типов в области видимости пространства имен ... 48
Соглашения о вызове ... 48
Концевые итераторы ... 48
Пространство имен для имен из стандартной библиотеки C ... 49
Адаптеры классов и адаптеры экземпляров ... 49
Имена заголовочных файлов ... 49

Часть I. Основы ... 50

Глава 1. Стандартная библиотека шаблонов ... 52

1.1. Основные понятия ... 52
1.2. Контейнеры ... 53
1.2.1. Последовательные контейнеры ... 53
1.2.2. Ассоциативные контейнеры ... 54
1.2.3. Непрерывность памяти ... 54
1.2.4. swap ... 54
1.3. Итераторы ... 55
1.3.1. Итераторы ввода ... 55
1.3.2. Итераторы вывода ... 56
1.3.3. Однонаправленные итераторы ... 57
1.3.4. Двунаправленные итераторы ... 57
1.3.5. Итераторы с произвольным доступом ... 58
1.3.6. Оператор выбора члена ... 58
1.3.7. Предопределенные адаптеры итераторов ... 60
1.4. Алгоритмы ... 61
1.5. Объекты-функции ... 62
1.6. Распределители ... 62

Глава 2. Концепции расширения STL, или Как STL ведет себя при встрече с реальным миром ... 63

2.1. Терминология ... 63
2.2. Наборы ... 64
2.2.1. Изменчивость ... 66
2.3. Итераторы ... 67
2.3.1. Изменчивость ... 68
2.3.2. Обход ... 68
2.3.3. Определение характеристик на этапе компиляции ... 68
2.3.4. Категория ссылок на элементы ... 68
2.3.5. Общее и независимое состояние ... 68
2.3.6. Не пересмотреть ли классификацию итераторов? ... 69

Глава 3. Категории ссылок на элементы 71

3.1. Введение 71
3.2. Ссылки в C++ 71
3.2.1. Ссылки на элементы STL-контейнеров 72
3.3. Классификация ссылок на элементы 73
3.3.1. Перманентные 73
3.3.2. Фиксированные 74
3.3.3. Чувствительные 74
3.3.4. Недолговечные 76
3.3.5. Временные по значению 78
3.3.6. Отсутствующие 79
3.4. Использование категорий ссылок на итераторы 79
3.4.1. Определение категории на этапе компиляции 79
3.4.2. Как компилятор может помочь избежать неопределенного поведения итератора 80
3.5. Определение оператора operator ->() 81
3.6. Еще о категориях ссылок на элементы 82

Глава 4. Забавная безвременная ссылка 83

Глава 5. Принцип DRY SPOT 85

5.1. Принцип DRY SPOT в C++ 85
5.1.1. Константы 85
5.1.2. Оператор dimensionof() 86
5.1.3. Порождающие функции 87
5.2. Когда в C++ приходится нарушать принцип DRY SPOT 87
5.2.1. Родительские классы 87
5.2.2. Типы значений, возвращаемых функциями 88
5.3. Замкнутые пространства имен 89

Глава 6. Закон дырявых абстракций 91

Глава 7. Программирование по контракту 93

7.1. Виды контроля 93
7.2. Механизмы контроля 95

Глава 8. Ограничения 96

8.1. Поддержка со стороны системы типов 96
8.2. Статические утверждения 97

Глава 9. Прокладки 99

9.1. Введение 99
9.2. Основные прокладки 100
9.2.1. Атрибутные прокладки 100
9.2.2. Конвертирующие прокладки 101
9.3. Составные прокладки 104
9.3.1. Прокладки строкового доступа 104

Глава 10. Утка и гусь, или Занимательные основы частичного структурного соответствия 108

10.1. Соответствие 108
10.1.1. Соответствие по имени 108
10.1.2. Структурное соответствие 110
10.1.3. Утка и гусь 111
10.2. Явное семантическое соответствие 113
10.2.1. Концепции 113
10.2.2. Пометка с помощью типов-членов 114
10.2.3. Прокладки 114
10.3. Пересекающееся соответствие 115

Глава 11. Идиома RAII 116

11.1. Изменчивость 116
11.2. Источник ресурса 116

Глава 12. Инструменты для работы с шаблонами 118

12.1. Характеристические классы 118
12.1.1. Класс base_type_traits 120
12.1.2. Класс sign_traits 121
12.1.3. Свойства типа: мини-характеристики 122
12.1.4. Класс is_integral_type 122
12.1.5. Класс is_signed_type 124
12.1.6. Класс is_fundamental_type 124
12.1.7. Класс is_same_type 125
12.2. Генераторы типов 126
12.2.1. Класс stlsoft::allocator_selector 126
12.3. Истинные typedef 127

Глава 13. Выводимая адаптация интерфейса: адаптации типов с неполными интерфейсами на этапе компиляции 128

13.1. Введение 128

13.2. Адаптация типов с неполными интерфейсами 129
13.3. Адаптация неизменяемых наборов 130
13.4. Выводимая адаптация интерфейса 131
13.4.1. Выбор типа 132
13.4.2. Распознавание типа 133
13.4.3. Исправление типа 134
13.5. Применение IIA к диапазону 136

Глава 14. Гипотеза Хенни, или Шаблоны атакуют! 138

Глава 15. Независимые автономии друзей equal() 140

15.1. Опасайтесь неправильного использования функций-друзей, не являющихся членами 140
15.2. Наборы и их итераторы 143

Глава 16. Важнейшие компоненты 144

16.1. Введение 144
16.2. Класс auto_buffer 144
16.2.1. Это не контейнер! 145
16.2.2. Интерфейс класса 146
16.2.3. Копирование 147
16.2.4. Воспитанные распределители идут последними 147
16.2.5. Метод swap() 148
16.2.6. Производительность 148
16.3. Класс filesystem_traits 149
16.3.1. Типы-члены 149
16.3.2. Работа со строками 150
16.3.3. Работа с именами из файловой системы 150
16.3.4. Операции с состоянием объектов файловой системы 153
16.3.5. Операции управления файловой системой 154
16.3.6. Типы возвращаемых значений и обработка ошибок 154
16.4. Класс file_path_buffer 154
16.4.1. Класс basic_?? 156
16.4.2. UNIX и PATH_MAX 157
16.4.3. Windows и MAX_PATH 158
16.4.4. Использование буферов 159
16.5. Класс scoped_handle 159
16.6. Функция dl_call() 160

Часть II. Наборы 163

Глава 17. Адаптация API glob 167

17.1. Введение 167

17.1.1. Мотивация 167
17.1.2. API glob 169
17.2. Анализ длинной версии 171
17.3. Класс unixstl::glob_sequence 174
17.3.1. Открытый интерфейс 174
17.3.2. Типы-члены 175
17.3.3. Переменные-члены 176
17.3.4. Флаги 176
17.3.5. Конструирование 179
17.3.6. Размер и доступ к элементам 180
17.3.7. Итерация 181
17.3.8. Метод init_glob_() 182
17.4. Анализ короткой версии 187
17.5. Резюме 188

Глава 18. Интерлюдия: конфликты в конструкторах и дизайн, который не то чтобы плох, но мало подходит для беспрепятственного развития 190

Глава 19. Адаптация API opendir/readdir 193

19.1. Введение 193
19.1.1. Мотивация 193
19.1.2. API opendir/readdir 195
19.2. Анализ длинной версии 195
19.3. Класс unixstl::readdir_sequence 197
19.3.1. Типы и константы-члены 199
19.3.2. Конструирование 200
19.3.3. Методы, относящиеся к размеру и итерированию 200
19.3.4. Методы доступа к атрибутам 202
19.3.5. const_iterator, версия 1 202
19.3.6. Использование версии 1 206
19.3.7. const_iterator, версия 2: семантика копирования 207
19.3.8. operator ++() 210
19.3.9. Категория итератора и адаптируемые типы-члены 210
19.3.10. operator ->() 211
19.3.11. Поддержка флагов fullPath и absolutePath 211
19.4. Альтернативные реализации 214
19.4.1. Хранение элементов в виде мгновенного снимка 214
19.4.2. Хранение элементов в виде итератора 215
19.5. Резюме 215

Глава 20. Адаптация API FindFirstFile/FindNextFile 217

20.1. Введение 217