

УДК 004.4
ББК 32.973.202-018.2
Д22

Д22 Артур О'Двайр
Осваиваем C++17 STL / пер. с англ. А. Н. Киселева – М.: ДМК Пресс, 2019. – 352 с.: ил.

ISBN 978-5-97060-663-6

Стандарт C++17, которому посвящена книга, удвоил объем библиотеки в сравнении с C++11. Вы узнаете о наиболее важных особенностях стандартной библиотеки C++17 со множеством примеров, научитесь создавать свои типы итераторов, диспетчеры памяти, пулы потоков выполнения. Также рассмотрены отличия мономорфизма, полиморфизма и обобщенных алгоритмов.

Издание адресовано разработчикам, желающим овладеть новыми особенностями библиотеки C++17 STL и в полной мере использовать ее компоненты. Знакомство с языком C++ является обязательным условием.

УДК 004.4
ББК 32.973.202-018.2

Copyright ©Packt Publishing 2018. First published in the English language under the title Mastering the C++17 STL – (9781787126824)

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 978-1-78712-682-4 (англ.)
ISBN 978-5-97060-663-6 (рус.)

Copyright © 2017 Packt Publishing.
© Оформление, перевод на русский язык, издание,
ДМК Пресс, 2019

Оглавление

Предисловие от разработчиков C++ в России и Беларуси.....	10
Об авторе	11
О научном редакторе	11
Предисловие	12
Содержание книги	12
Что потребуется для работы с книгой.....	13
Кому адресована эта книга.....	13
Типографские соглашения	14
Отзывы и пожелания	14
Скачивание исходного кода примеров	15
Список опечаток.....	15
Нарушение авторских прав.....	15
Глава 1. Классический полиморфизм и обобщенное программирование	16
Конкретные мономорфные функции	16
Классические полиморфные функции.....	17
Обобщенное программирование с шаблонами	19
Итоги	22
Глава 2. Итераторы и диапазоны	24
Проблема целочисленных индексов	24
За границами указателей	25
Константные итераторы	28
Пара итераторов определяет диапазон	29
Категории итераторов	31
Итераторы ввода и вывода	33
Объединяем все вместе	36
Устаревший std::iterator.....	39
Итоги	42
Глава 3. Алгоритмы с парами итераторов.....	44
Замечание о заголовках.....	44
Диапазонные алгоритмы только для чтения.....	44
Манипулирование данными с std::copy.....	51
Вариации на тему: std::move и std::move_iterator	54
Непростое копирование с std::transform	57
Диапазонные алгоритмы только для записи.....	59

Алгоритмы, влияющие на жизненный цикл объектов	60
Наш первый перестановочный алгоритм: <code>std::sort</code>	62
Обмен местами, обратное упорядочение и разделение	63
Ротация и перестановка	67
Кучи и пирамидальная сортировка	69
Слияние и сортировка слиянием	71
Поиск и вставка в сортированный массив с <code>std::lower_bound</code>	71
Удаление из сортированного массива с <code>std::remove_if</code>	73
Итоги	77
Глава 4. Зоопарк контейнеров	78
Понятие владения	78
Простейший контейнер: <code>std::array<T, N></code>	80
Рабочая лошадка: <code>std::vector<T></code>	84
Изменение размера <code>std::vector</code>	85
Вставка и стирание в <code>std::vector</code>	89
Ловушки <code>vector<bool></code>	90
Ловушки в конструкторах перемещения без <code>noexcept</code>	91
Быстрый гибрид: <code>std::deque<T></code>	93
Особый набор возможностей: <code>std::list<T></code>	94
Какие отличительные особенности имеет <code>std::list</code> ?	95
Список без удобств <code>std::forward_list<T></code>	97
Абстракции с использованием <code>std::stack<T></code> и <code>std::queue<T></code>	98
Удобный адаптер: <code>std::priority_queue<T></code>	99
Деревья: <code>std::set<T></code> и <code>std::map<K, V></code>	100
Замечание о прозрачных компараторах	104
Необычные <code>std::multiset<T></code> и <code>std::multimap<K, V></code>	105
Перемещение элементов без перемещения	107
Хеши: <code>std::unordered_set<T></code> и <code>std::unordered_map<K, V></code>	109
Фактор загрузки и списки в корзинах	111
Откуда берется память?	112
Итоги	113
Глава 5. Словарные типы	114
История <code>std::string</code>	114
Маркировка ссылочных типов с <code>reference_wrapper</code>	116
C++11 и алгебраические типы	117
Работа с <code>std::tuple</code>	118
Манипулирование значениями кортежа	120
Замечание об именованных классах	121
Выражение альтернатив с помощью <code>std::variant</code>	122
Чтение вариантов	123
О <code>make_variant</code> и семантике типа-значения	125

Задержка инициализации с помощью <code>std::optional</code>	127
И снова <code>variant</code>	131
Бесконечное число альтернатив с <code>std::any</code>	132
<code>std::any</code> и полиморфные типы.....	134
Коротко о стирании типа.....	135
<code>std::any</code> и копирование.....	137
И снова о стирании типов: <code>std::function</code>	138
<code>std::function</code> , копирование и размещение в динамической памяти.....	140
Итоги.....	141
Глава 6. Умные указатели.....	142
История появления умных указателей.....	142
Умные указатели никогда ничего не забывают.....	143
Автоматическое управление памятью с <code>std::unique_ptr<T></code>	144
Почему в C++ нет ключевого слова <code>finally</code>	147
Настройка обратного вызова удаления.....	148
Управление массивами с помощью <code>std::unique_ptr<T[]></code>	149
Подсчет ссылок с <code>std::shared_ptr<T></code>	150
Не допускайте двойного управления!.....	153
Удерживание обнуляемых дескрипторов с помощью <code>weak_ptr</code>	153
Сообщение информации о себе с <code>std::enable_shared_from_this</code>	156
Странно рекурсивный шаблон проектирования.....	159
Заключительное замечание.....	160
Обозначение неисключительности с <code>observer_ptr<T></code>	160
Итоги.....	162
Глава 7. Конкуренция.....	163
Проблемы с <code>volatile</code>	163
Использование <code>std::atomic<T></code> для безопасного доступа	
в многопоточной среде.....	166
Атомарное выполнение сложных операций.....	168
Большие атомарные типы.....	170
Поочередное выполнение с <code>std::mutex</code>	171
Правильный порядок «приобретения блокировок».....	173
Всегда связывайте мьютекс с управляемыми данными.....	176
Специальные типы мьютексов.....	180
Повышение статуса блокировки для чтения/записи.....	183
Понижение статуса блокировки для чтения/записи.....	183
Ожидание условия.....	184
Обещания о будущем.....	187
Подготовка заданий для отложенного выполнения.....	190
Будущее механизма <code>future</code>	192
Поговорим о потоках.....	194
Идентификация отдельных потоков и текущего потока.....	196

Исчерпание потоков и <code>std::async</code>	198
Создание своего пула потоков.....	200
Оптимизация производительности пула потоков	204
Итоги	206
Глава 8. Диспетчеры памяти	208
Диспетчер памяти обслуживает ресурс памяти.....	209
Еще раз об интерфейсах и понятиях	210
Определение кучи с помощью <code>memory_resource</code>	212
Использование стандартных ресурсов памяти	215
Выделение из ресурса пулов.....	217
500-головый стандартный диспетчер памяти	218
Метаданные, сопровождающие причудливые указатели.....	222
Прикрепление контейнера к единственному ресурсу памяти.....	227
Использование диспетчеров памяти стандартных типов.....	229
Настройка ресурса памяти по умолчанию.....	230
Создание контейнера с поддержкой выбора диспетчера памяти.....	231
Передача вниз с <code>scoped_allocator_adaptor</code>	237
Передача разных диспетчеров памяти	240
Итоги	242
Глава 9 Потоки ввода/вывода	244
Проблемы ввода/вывода в C++	244
Буферизация и форматирование.....	246
POSIX API.....	247
Стандартный C API.....	250
Буферизация в стандартном C API	252
Форматирование с помощью <code>printf</code> и <code>snprintf</code>	257
Классическая иерархия потоков ввода/вывода	260
Потоки данных и манипуляторы	264
Потоки данных и обертки	267
Решение проблемы манипуляторов	269
Форматирование с <code>ostringstream</code>	270
Примечание о региональных настройках.....	271
Преобразование чисел в строки	273
Преобразование строк в числа.....	275
Чтение по одной строке или по одному слову.....	279
Итоги	281
Глава 10. Регулярные выражения	282
Что такое регулярное выражение?.....	283
Замечание об экранировании обратными слешами.....	284
Воплощение регулярных выражений в объектах <code>std::regex</code>	286
Сопоставление и поиск	287

Извлечение совпадений с подвыражениями	288
Преобразование совпадений в значения данных.....	292
Итерации по нескольким совпадениям.....	293
Использование регулярных выражений для замены строк	297
Грамматика регулярных выражений ECMAScript	299
Непоглощающие конструкции	302
Малопонятные особенности и ловушки ECMAScript	303
Итоги	305
Глава 11. Случайные числа	306
Случайные и псевдослучайные числа.....	306
Проблема функции rand()	308
Решение проблем с <random>	310
Генераторы	310
Истинно случайные биты и std::random_device	311
Псевдослучайные биты с std::mt19937	311
Фильтрация вывода генераторов с помощью адаптеров.....	313
Распределения.....	316
Имитация броска игровой кости с uniform_int_distribution	316
Генерирование выборок с normal_distribution	318
Взвешенный выбор с discrete_distribution	319
Перемешивание карт с std::shuffle.....	320
Итоги	321
Глава 12. Файловая система.....	323
Примечание о пространствах имен	323
Очень длинное примечание об уведомлениях об ошибках.....	325
Использование <system_error>	327
Коды ошибок и условия ошибок.....	331
Возбуждение ошибок с std::system_error	334
Файловые системы и пути	336
Представление путей в C++	338
Операции с путями	340
Получение информации о файлах с directory_entry.....	342
Обход каталогов с directory_iterator.....	343
Рекурсивный обход каталогов	343
Изменение файловой системы	344
Получение информации о диске	345
Итоги	346
Предметный указатель	347