

УДК 004.42
ББК 32.372
Г82

Главный научный редактор:

Романов А. Ю. – канд. техн. наук, доцент Московского института электроники и математики им. А. Н. Тихонова Национального исследовательского университета «Высшая школа экономики».

Райнер Гримм

Г82 С++20 в деталях / пер. с англ. А. В. Борескова; под науч. ред. А. Ю. Романова, И. И. Романовой. – М.: ДМК Пресс, 2023. – 518 с.: ил.

ISBN 978-5-97060-956-9

В этой книге подробно рассказывается о новом стандарте С++20. Для тех, кто незнаком с С++20, приводится его краткий обзор, а далее рассматриваются ключевые возможности языка. Вы получите представление о ключевых изменениях в ядре языка (концепты и модули), новой библиотеке диапазонов, корутинах, а затем сможете применить теорию на практике, изучив ряд примеров. Книгу можно использовать как справочное руководство и изучать главы в удобном для вас порядке.

Издание будет полезно разработчикам, желающим освоить последнюю версию С++, изучить передовые возможности и добавления в язык, а также заглянуть за кулисы разработки новых стандартов языка и узнать, как предлагаются, обсуждаются и утверждаются новые изменения в стандарт С++ и чем вызваны эти изменения.

Книга, которую вы держите в руках, открывает серию «Книжная полка Истового Инженера», которая издается при поддержке компании YADRO. Это издание подготовлено к публикации Московским институтом электроники и математики им. А. Н. Тихонова НИУ ВШЭ совместно с «ДМК Пресс».

УДК 004.42
ББК 32.372

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.

ISBN (анг.) 979-8-73298-945-8
ISBN (рус.) 978-5-97060-956-9

© 2020 Rainer Grimm
© Оформление, издание, перевод, ДМК Пресс, 2023
© Научное редактирование, НИУ ВШЭ, 2023

Оглавление

Предисловие от издательства	11
Отзывы и пожелания.....	11
Список опечаток.....	11
Нарушение авторских прав	11
Предисловие от главного редактора русского перевода	12
Истории читателей.....	16
Введение	17
Соглашения	17
Специальные шрифты	17
Специальные блоки.....	18
Исходный код.....	18
Компиляция программ	18
Как вам следует читать эту книгу?	20
Личные замечания	20
Благодарности	20
Сиппи	20
Редакторы русского перевода.....	21
Обо мне	22
О ЯЗЫКЕ C++.....	24
1. Исторический контекст	25
1.1 C++98.....	25
1.2 C++03.....	26
1.3 TR1	26
1.4 C++11.....	26
1.5 C++14.....	26
1.6 C++17.....	26

2. Стандартизация	27
2.1 Стадия 3	28
2.2 Стадия 2	28
2.3 Стадия 1	28
КРАТКИЙ ОБЗОР C++20	30
3. C++20	31
3.1 Большая четверка	32
3.1.1 Концепты	32
3.1.2 Модули	33
3.1.3 Библиотека диапазонов	34
3.1.4 Корутины	35
3.2 Ядро языка	37
3.2.1 Оператор трехстороннего сравнения	37
3.2.2 Назначенная инициализация	38
3.2.3 consteval и constexpr	40
3.2.4 Улучшения работы с шаблонами	41
3.2.5 Улучшения лямбд	42
3.2.6 Новые атрибуты	42
3.3 Стандартная библиотека	43
3.3.1 std::span	43
3.3.2 Улучшения контейнеров	44
3.3.3 Арифметические утилиты	44
3.3.4 Календарь и временные зоны	44
3.3.5 Библиотека для форматированного вывода	45
3.4 Параллельность	46
3.4.1 Атомарные операции	46
3.4.2 Семафоры	47
3.4.3 Защелки и барьеры	47
3.4.4 Кооперативное прерывание	48
3.4.5 std::jthread	50
3.4.6 Синхронные выходные потоки	51
ПОДРОБНО ПРО C++20	54
4. Ядро языка	55
4.1 Концепты	55
4.1.1 Два неправильных подхода	56
4.1.2 Преимущества концептов	62
4.1.3 Длинная, длинная история	62
4.1.4 Исползования концептов	63
4.1.5 Ограниченные или неограниченные заполнители	75
4.1.6 Сокращенные шаблонные функции	78
4.1.7 Предопределенные концепты	82
4.1.8 Определение концептов	88

4.1.9 Применение концептов.....	96
4.2 Модули.....	108
4.2.1 Для чего нужны модули?.....	108
4.2.2 Преимущества использования модулей.....	114
4.2.3 Простой пример использования модулей.....	115
4.2.5 Экспорт из модуля.....	120
4.2.6 Рекомендации по структуре модуля.....	121
4.2.7 Блок интерфейса модуля и блок реализации модуля.....	122
4.2.8 Подмодули и разделы модулей.....	125
4.2.9 Шаблоны в модулях.....	129
4.2.10 Линковка на уровне модулей.....	132
4.2.11 Заголовочные блоки.....	134
4.3 Оператор трехстороннего сравнения.....	136
4.3.1 Упорядочение до C++20.....	136
4.3.2 Упорядочение начиная со стандарта C++20.....	138
4.3.3 Категории сравнения.....	141
4.3.4 Создаваемый компилятором оператор трехстороннего сравнения.....	143
4.3.5 Переписывание выражений.....	148
4.3.6 Задаваемые пользователем и создаваемые автоматически операторы сравнения.....	151
4.4 Назначенная инициализация.....	154
4.4.1 Агрегированная инициализация.....	154
4.4.2 Именованная инициализация членов класса.....	156
4.5 consteval и constexpr.....	161
4.5.1 consteval.....	161
4.5.2 constexpr.....	163
4.5.3 Выполнение функций.....	164
4.5.4 Инициализация переменных.....	166
4.5.5 Исправляем проблему порядка статической инициализации.....	167
4.6 Улучшение работы с шаблонами.....	173
4.6.1 Условный явный конструктор.....	173
4.6.2 Нетипизированные параметры шаблона.....	176
4.7 Улучшения лямбд.....	180
4.7.1 Шаблонные параметры для лямбд.....	180
4.7.2 Определение неявного копирования указателя this.....	184
4.7.3 Лямбды в контекстах без выполнения. Использование конструктора по умолчанию и копирования для лямбд без состояния.....	186
4.8 Новые атрибуты.....	190
4.8.1 [[nodiscard("reason")]].....	191
4.8.2 [[likely]] и [[unlikely]].....	195
4.8.3 [[no_unique_address]].....	196
4.9 Дополнительные улучшения.....	199
4.9.1 volatile.....	199
4.9.2 Оператор цикла for с инициализацией на основе диапазона.....	201
4.9.3 Виртуальная функция с constexpr.....	202

4.9.4 Новый символьный тип для utf8-строк: <code>char8_t</code>	204
4.9.5 Использование <code>using enum</code> в локальной области видимости.....	205
4.9.6 Инициализаторы по умолчанию для битовых полей	206

5. Стандартная библиотека209

5.1 Библиотека диапазонов.....	210
5.1.1 Концепты <code>ranges</code> и <code>views</code>	211
5.1.2 Работа алгоритмов непосредственно со всем контейнером	212
5.1.3 Композиция функций.....	216
5.1.4 Отложенное выполнение	218
5.1.5 Определение видов.....	221
5.1.6 Аромат Python.....	224
5.2 <code>std::span</code>	230
5.2.1 Статическая и динамическая длина	230
5.2.2 Автоматический вывод размера непрерывной последовательности объектов	232
5.2.3 Создание <code>std::span</code> из указателя и размера	233
5.2.4 Изменение объектов, к которым происходит обращение через ссылку.....	235
5.2.5 Обращение к элементам <code>std::span</code>	236
5.2.6 Постоянный диапазон изменяемых элементов	238
5.3 Улучшения контейнеров	241
5.3.1 Контейнеры и алгоритмы со спецификатором <code>constexpr</code>	241
5.3.2 <code>std::array</code>	242
5.3.3 Последовательное удаление из контейнеров	244
5.3.4 <code>contains</code> для ассоциативных контейнеров	249
5.3.5 Проверка строки на наличие префикса и суффикса	252
5.4 Арифметические функции.....	255
5.4.1 Безопасное сравнение целых чисел	255
5.4.2 Математические константы.....	260
5.4.3 Вычисление середины отрезка и линейная интерполяция	262
5.4.4 Работа с битами	263
5.5 Календарные зоны и часовые пояса.....	269
5.5.1 Время дня	270
5.5.2 Календарные даты	273
5.5.3 Часовые пояса	289
5.6 Библиотека форматирования	296
5.6.1 Функции форматирования.....	296
5.6.2 Форматная строка.....	298
5.6.3 Задаваемые пользователем типы	306
5.7 Дальнейшие улучшения	312
5.7.1 <code>std::bind_front</code>	312
5.7.2 <code>std::is_constant_evaluated</code>	314
5.7.3 <code>std::source_location</code>	316

6. Параллельность	318
6.1 Корутины	319
6.1.1 Функция-генератор	320
6.1.2 Характеристики	323
6.1.3 Фреймворк	325
6.1.4 Ожидаемые и ожидающие объекты	328
6.1.5 Исполняемый поток процессов	330
6.1.6 co_return	334
6.1.7 co_yield	336
6.1.8 co_wait	339
6.2 Атомарные переменные	349
6.2.1 std::atomic_ref	349
6.2.2 Атомарный умный указатель	358
6.2.3 Расширения std::atomic_flag	362
6.2.4 Расширения std::atomic	370
6.3 Семафоры	374
6.4 Защелки и барьеры	379
6.4.1 std::latch	379
6.4.2 std::barrier	385
6.5 Координированное прерывание	389
6.5.1 std::stop_source	390
6.5.2 std::stop_token	391
6.5.3 std::stop_callback	391
6.6 std::jthread	398
6.6.1 Автоматическое присоединение	399
6.6.2 Кооперативное прерывание std::jthread	401
6.7 Синхронизированные потоки вывода	404
7. Практические примеры	413
7.1 Быстрая синхронизация потоков	414
7.1.1 Условные переменные	415
7.1.2 std::atomic_flag	417
7.1.3 std::atomic<bool>	420
7.1.4 Семафоры	422
7.1.5 Общая статистика	424
7.2 Вариации объектов future	425
7.2.1 Ленивый объект future	428
7.2.2 Выполнение на другом потоке	431
7.3 Модификация и обобщение генератора	436
7.3.1 Изменения	440
7.3.2 Обобщение	443
7.4 Различные потоковые архитектуры, основанные на задачах	447
7.4.1 Прозрачная архитектура ожидающего потока задач	447
7.4.2 Автоматическое возобновление ожидающей задачи	450
7.4.3 Автоматическое возобновление ожидающего объекта на отдельном потоке	453

ЭПИЛОГ	458
ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ	460
8. C++23 и не только	461
8.1 C++23	462
8.1.1 Библиотека сопрограмм	462
8.1.2 Модуляризированная стандартная библиотека	476
8.1.3 Исполнители	479
8.1.4 Сетевая библиотека	483
8.2 C++23 или позже	485
8.2.1 Контракты	485
8.2.2 Рефлексия	488
8.2.3 Сопоставление с образцом	492
8.3 Дополнительная информация о стандарте C++23	496
9. Дополнительное тестирование	497
10. Глоссарий	510