КНИЖНАЯ ПОЛКА ИСТОВОГО ИНЖЕНЕРА

РАЙНЕР ГРИММ

C++20 В ДЕТАЛЯХ

YADRO
ВЫСШАЯ ШКОЛА ЭКОНОМИКИ
МИЭМ
ДМК ПРЕСС ИЗДАТЕЛЬСТВО

Райнер Гримм

C++20 в деталях

C++20

Get the Details

Rainer Grimm

C++20 в деталях

Райнер Гримм

ДМК Пресс
издательство
Москва, 2023

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

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

Оглавление

Предисловие от издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в издании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от непонимания текста и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим их в следующих тиражах.

Нарушение авторских прав

Пиратство в сети Интернет по-прежнему является насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы знаете о незаконной публикации какой-либо из наших книг в сети Интернет, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие от главного редактора русского перевода

Дорогие друзья!

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».

Если вы интересуетесь цифровой электроникой, разработкой на ПЛИС, проектированием на языках описания аппаратуры Verilog или VHDL, то вы, скорее всего, уже знакомы с книгами серии «Цифровой синтез», такими как: Д. Харрис и С. Л. Харрис «Цифровая схемотехника и архитектура компьютера»; «Цифровой синтез: практический курс» (под. ред. А. Ю. Романова и Ю. В. Панчула), Ф. Бруно «Программирование FPGA для начинающих» и др. Эта книга несколько отличается по тематике и в первую очередь ориентирована на программистов, работающих с языками высокого уровня.

Почему я взялся редактировать русский перевод книги Р. Гримма «C++20 в деталях»?

На самом деле я сильно сомневался. Но меня убедила моя коллега Ирина Романова, она же и выполнила первую вычитку и редактуру этой книги. Без ее помощи я бы не согласился редактировать данный материал. Остальные мотивы описаны далее.

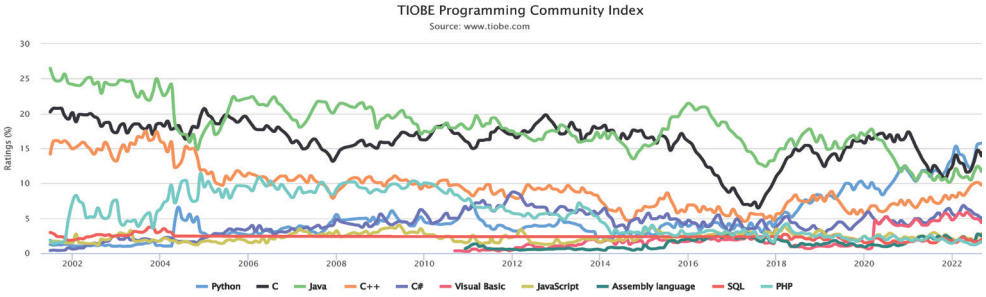
Как я использую C++?

Хотя я применяю C++ в своей практике постоянно, использую я его обычно как язык системного программирования, а также для программирования встраиваемых систем (таких как микроконтроллеры), т. е. вполне могу обходиться подмножеством C; а если нужна объектно-ориентированность, то вполне достаточно тех возможностей, которые предоставляет стандарт C++98.

Зачем изучать C++20?

Зачем же тогда изучать все эти C++11, потом C++14 и т. д., а теперь и C++20? (Ведь на подходе уже и следующий стандарт!)

Ответ: это очень интересно! C++ – это один из ведущих языков. По рейтингу TIOB C/C++ – все еще самый популярный язык в мире. Популярнее, чем Python!



C++ – довольно «старый» язык, который развивается, меняется, создает новые методы и принципы программирования прямо у нас на глазах. Это (во многом) отражение прошлого и будущего всех классических высокоуровневых языков программирования. Каждый новый стандарт – это новые парадигмы и «фишки» программирования. При этом предлагают, описывают и внедряют их обычные программисты в открытом комьюнити.

Поэтому даже если вы заядлый «питонист» или никогда не планируете использовать новшества C++, все равно стоит делать как я: следить за выходом новых стандартов и хотя бы пролистывать релизы о нововведениях в язык и другую литературу по теме – потому что понимать ход современной передовой мысли разработчиков программного кода, знать новые приемы и возможности ключевого языка общения всех программистов очень важно для саморазвития любого, кто любит и практикует разработку программ.

Зачем нужен русский перевод этой книги?

Очень часто в среде программистов встречаю мнение, что изучать специализированную литературу по программированию надо на языке оригинала: английском. В целом я с этим согласен – сам большинство информации по теме получаю на английском. Но это не потому, что читать на английском удобней, а потому, что на русском почти всегда нужной информации нет. Если хотите читать оригинал, пожалуйста, читайте его. Но если кто-то хочет быстро познакомиться с новшествами C++20 или разобраться более детально с какими-то тонкостями этих новшеств на примерах – эта книга для вас.

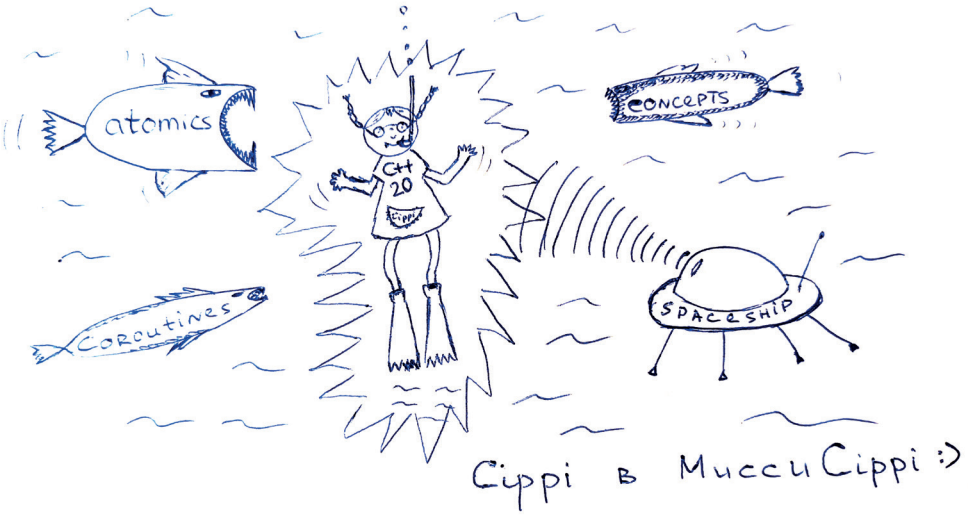
Почему именно книга Р. Гримма?

Потому что это признанный автор, а его книги очень популярны. Потому что Р. Гримм сам из «тусовки» разработчиков новых стандартов C++, и у него сравнительно легкая и интересная подача. Надеюсь, нам удалось это сохранить и в русском переводе. А еще в этой книге довольно информативные и показательные примеры с небольшим количеством ошибок. Этим могут похвастаться далеко не все издания в данной области, даже издания именитых авторов!

Еще для себя я выработал один неформальный принцип оценки зарубежных книг, рассчитанных на большую аудиторию: это то, как она оформлена. Чем больше вложено в оформление издания, тем выше вероятность того (хоть и не всегда!), что содержание книги будет на уровне. Дело в том, что западное

общество довольно инфантильно (чего только стоят «График Шму» или «Принцип печенья Орео»!), поэтому книге без «картинок» довольно сложно добиться популярности. Объединение «картинок» и хорошего содержания является фактором, обеспечивающим успех книги.

Данная книга соответствует этому критерию. Хотя, на мой взгляд, персонаж книги, Сиппи, довольно безобразна, а подписи к рисункам катастрофически неинформативны. Такое представление персонажа я (не художник) тоже выдать могу! Но не включаю его в серьезную книгу. Хотя… а почему бы и нет?



Было желание убрать все эти бессмысленные рисунки из книги, но, в конце концов, было решено смириться с этим.

Для кого эта книга?

Давайте вернемся к серьезному тону и, наконец, ответим на вопрос: кому будет интересна эта книга? Программирование C++ по ней вы не изучите – для этого есть классические книги Х. Дейтела, Б. Страуструпа, С. Прата и др. Для того чтобы воспринимать материал, нужно уже знать концепцию объектно-ориентированного программирования и иметь хотя бы небольшой опыт разработки программ на C++ или «близких по духу» высокоуровневых языках – вроде C#, Java, Python. Неплохо бы знать такие концепции, как лямбда-функции и многопоточные/параллельные программы. Стандарт C++20 глубоко вы по этой книге тоже не изучите. Она нужна для первого быстрого и легкого знакомства с новшествами, которые появились в новом стандарте языка. Заодно читатели смогут узнать немного о причинах появления этих новшеств, принципах и идеях, которые в них заложены. Также в книге очень неплохие и понятные примеры программ. Остальные тонкости вы при необходимости узнаете на практике или изучив стандарт.

Ошибки, терминология и другие замечания

Проблема перевода такого рода книг состоит в том, что довольно часто в русском языке нет устоявшегося аналога английскому термину или прямой русский перевод выглядит глупо и неестественно. В основном это так, потому что сами английские термины являются «новоделами». Например, тот же coroutine. Все уже смирились говорить и писать «корутин», притом что в русском языке есть тоже иностранное, но более привычное слово «сопрограмма», которое имеет приблизительно то же значение. В русском переводе осуществлена попытка соблюсти баланс, обеспечивающий читабельность и понятность текста: какие-то термины используются в форме, более привычной русскоязычному читателю, а какие-то вообще не переведены. По той же причине сознательно не переведены надписи на рисунках и комментарии в листингах.

Эту книгу прочитали два научных редактора, один корректор и еще два студента (огромная благодарность Александру Богомолову и Руслану Нуржанову), но, безусловно, она не идеальна. Я буду очень признателен тем внимательным читателям, которые обнаружат в данном издании какие-либо ошибки или опечатки и сообщат о них по адресу a.romanov@hse.ru или dmkpress@gmail.com (книги постоянно перепечатываются, и в каждом новом тираже все найденные ошибки и недочеты исправляются).

Также вы можете присылать свои рисунки, изображающие Сиппи. Я пока еще не придумал, что с ними делать, но (возможно) получится импровизированная выставка ☺

Истории читателей

Сандор Дарго, старший программист в Amadeus

«C++ 20 в деталях» – это именно та книга, которая вам сейчас нужна, если вы хотите погрузиться в последнюю версию C++. Это полное руководство, в котором Рейнер рассматривает не просто передовые возможности C++20, но и небольшие добавления в язык. Эта книга содержит множество примеров кода, так что даже если у вас еще нет доступа к последним компиляторам, у вас все равно будет хорошее представление о том, чего вы можете ожидать от различных возможностей языка. Крайне рекомендую прочитать.

Адриан Там, директор по Data Science, Synechron Inc

C++ очень сильно развился с момента его появления на свет. C++20 – это практически новый язык. Конечно, это книга не предназначена для того, чтобы научить вас наследованию или перегрузке, но если вы хотите привести свои знания C++ к современному уровню, то это очень подходящая книга. Вы удивитесь тому, сколько всего нового из C++20 вошло в базовый язык C++. Эта книга дает ясные объяснения и краткие примеры. Структура книги позволит вам использовать ее как справочник в дальнейшем. Она поможет вам узнать все современные возможности языка.

Введение

Моя книга по C++ – это и учебник, и справочное руководство. Она научит вас C++20 и даст вам детальную информацию о новом стандарте C++. Наиболее выдающимися возможностями C++20 являются следующие:

- **концепты** – изменяют способ, которым мы думаем о программах с шаблонами. Они являются семантическими категориями для параметров шаблонов. Они позволяют вам ясно выражать свои намерения через систему типов. Если что-то пойдет не так, то компилятор даст вам понятное сообщение об ошибке;
- **модули** – позволяют обойти ограничения заголовочных файлов. Они очень многообещающи. Например, разделение заголовочного файла и исходного файла становится таким уже устаревшим, как и препроцессор. В результате вы получаете более быструю компиляцию и легкий способ создания пакетов;
- новая **библиотека диапазонов** (ranges library) поддерживает применение алгоритмов к контейнерам, включающим алгоритмы, организованные как конвейер (pipe), и «ленивое» (lazy) применение алгоритмов к бесконечным потокам данных;
- благодаря **корутинам** (coroutines) асинхронное программирование в C++ становится широко распространенным. Корутины образуют основу для кооперативных задач, циклов обработки событий, бесконечных потоков данных или конвейеров.

Конечно, это далеко не все. Вот еще некоторые новые возможности C++20:

- генерируемые автоматически операторы сравнения;
- библиотеки для работы с датами и временными зонами;
- библиотека format;
- «виды» (view) непрерывных блоков памяти;
- улучшенные прерываемые потоки;
- атомарные умные указатели;
- семафоры;
- примитивы управления, такие как защелки (latch) и барьеры (barrier).

Соглашения

Есть всего несколько соглашений по форматированию этой книги:

Специальные шрифты

Курсив

Используется *курсив* для выделения цитат.

Жирный

Используется **жирный** шрифт для выделения имен.

Моноширинный

Используется моноширинный шрифт для кода, инструкций, ключевых слов и имен типов, переменных, функций и классов.

Специальные блоки

Блоки содержат подсказки, предупреждения и дополнительную информацию.

Подсказка

Этот блок предоставляет дополнительную информацию по текущему материалу и подсказки по компиляции программ.

Предупреждения

Блоки предупреждений должны помочь вам избегать ошибок.

Собранная информация

Этот блок в конце каждого раздела содержит важную информацию, которую стоит запомнить.

Исходный код

Примеры исходного кода, приведенные в книге, являются завершенными. Это значит, что если у вас есть подходящий компилятор, то вы можете их откомпилировать и выполнить. Имя исходного файла находится в заголовке каждого примера кода. Исходный код использует четыре пробела для табуляции (отступов). Иногда используется два пробела из соображений размещения на странице.

Кроме того, я не большой любитель директив namespace вроде `using namespace std`, поскольку они делают код более сложным для чтения и могут засорять пространства имен. Соответственно, я применяю их только тогда, когда это улучшает читаемость кода (например, `using namespace std::chrono_literals`). Иногда для более удобного размещения кода на странице я использую using, например `using std::chrono::system_clock`.

Таким образом, только в случае необходимости иногда делаются следующие отступления от общих правил оформления кода:

- табуляция на два пробела, а не на четыре;
- используется директива `using namespace std`.

Компиляция программ

Поскольку C++20 – это новый стандарт, то многие примеры могут быть откомпилированы только при помощи подходящих компиляторов. Я использую последние версии GCC[1], Clang[2] и MSVC[3]. При компиляции программы вы должны указать используемый стандарт C++. Это значит, что при использовании GCC

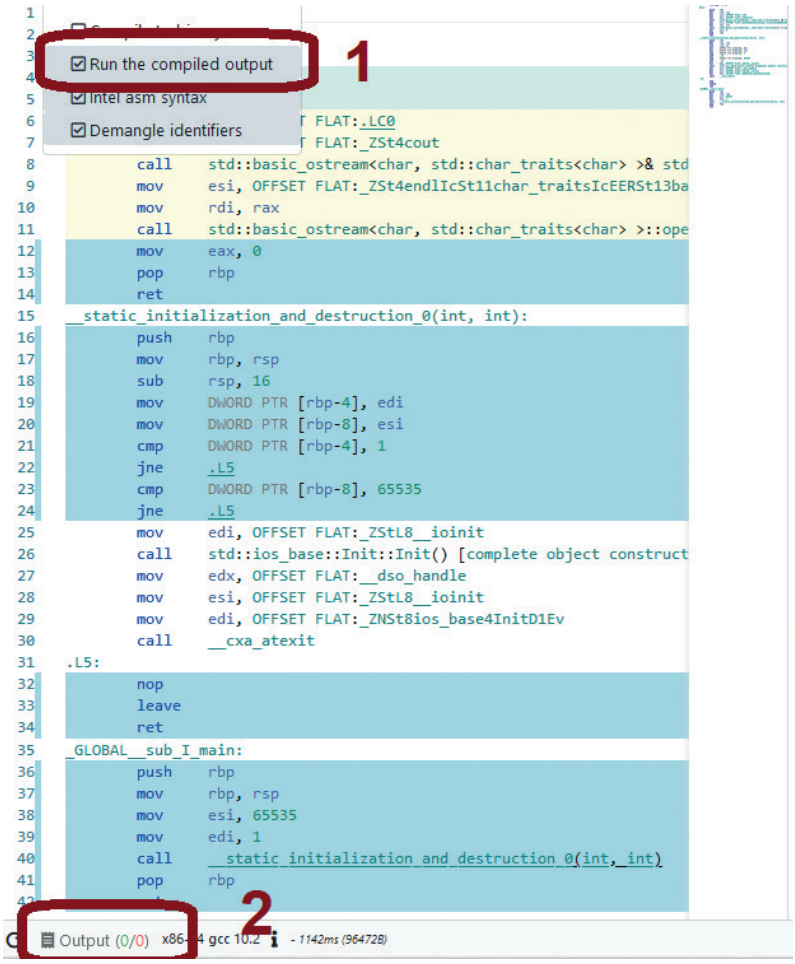
[1] https://gcc.gnu.org.

[2] https://clang.llvm.org.

[3] https://en.wikipedia.org/Microsoft_Visual_C%2B%2B.

или Clang вы должны указать флаг `-std=c++20` и для MSVC флаг `/std:c++latest`. При применении параллельности (concurrency) для GCC и Clang необходимо указать, что вам требуется линковка с библиотекой pthread при помощи опции `-pthread`.

Если у вас нет в распоряжении подходящего компилятора C++, то вы можете использовать онлайн-компилятор вроде Wandbox[1] или Compiler Explorer[2]. При применении Compiler Explorer с выбором компилятора GCC или Clang вы также можете выполнить откомпилированную программу. Для этого, во-первых, вам нужно выбрать `Run the compiled output` и, во-вторых, открыть окно `Output`.



Run code in the Compiler Explorer

Вы можете получить дополнительную информацию о совместимости с C++20 для различных компиляторов на cppreferebce.com[3].

[1] https://wandbox.org.

[2] https://godbolt.org.

[3] https://en.cppreferebce.com/w/cpp/compiler_support.

Как вам следует читать эту книгу?

Если вы незнакомы со стандартом C++20, то начните с краткого обзора для получения общей картины.

После получения общей картины вы можете переходить к ключевым возможностям языка (core language). Рассказ о каждой новой возможности языка является самодостаточным, но наилучшим вариантом будет прочтение всей книги от начала до конца. При первом прочтении вы можете пропустить то, что не упомянуто в кратком обзоре.

Личные замечания

Благодарности

Я просил читателей проверить на ошибки мой англоязычный блог Modernes-Cpp[1] и получил гораздо больше откликов, чем ожидал. Огромное спасибо вам всем. Вот имена тех, кто внес в это свой вклад: Bob Bird, Nicola Bombace, Dave Burchill, Sandor Dargo, James Drobina, Frank Grimm, Kilian Henneberger, Ivan «espkk» Kondakov, Péter Kardos, Rakesh Mane, Jonathan O'Connor, John Plaice, Iwan Smith, Paul Targosz, Steve Vinoski и Greg Wagner.

Особенное спасибо моей дочери Джульетте и моей жене Беатрис. Джульетта поправила мой язык и исправила множество опечаток. Беатрис создала персонажа Сиппи (Cippi, девочка на рисунках) и иллюстрации к этой книге.

Сиппи

Позвольте представить вам Сиппи, которая будет сопровождать вас в этой книге. Надеюсь, она вам понравится.

Я Сиппи (по аналогии с Пеппи Длинныйчулок), любознательная, умная и женственная!

[1] http://modernescpp.com.

Редакторы русского перевода

Романов Александр Юрьевич (https://www.hse.ru/staff/a.romanov) – главный научный редактор русского перевода данной книги, доцент Московского института электроники и математики им. А. Н. Тихонова Национального исследовательского университета «Высшая школа экономики» (МИЭМ НИУ ВШЭ). С 2014 г. работает в МИЭМ НИУ ВШЭ, где возглавляет лабораторию САПР (https://miem.hse.ru/edu/ce/cadsystem), специализирующуюся на проектной деятельности, а также разработке цифровых систем на ПЛИС/микроконтроллерах, робототехнических комплексов, аппаратных реализаций систем искусственного интеллекта, многопроцессорных систем, систем удаленного доступа к лабораторному оборудованию и т. д. В 2015 г. защитил диссертацию в Институте проблем проектирования в микроэлектронике РАН (г. Зеленоград), является автором более 150 научных статей, патентов и книг. А. Ю. Романов преподает C++ с 2009 г. в качестве лекционного и практического курса для студентов и постоянно использует данный язык в практической работе.

Романова Ирина Ивановна (https://www.hse.ru/staff/iromanova) – научный редактор русского перевода – занимается преподаванием компьютерных и инженерных дисциплин с 2010 г. В настоящее время работает старшим преподавателем в Московском институте электроники и математики им. А. Н. Тихонова Национального исследовательского университета «Высшая школа экономики» (МИЭМ НИУ ВШЭ), ведущий преподаватель и лектор дисциплины «Информатика» для студентов 1-го курса, а также соавтор известной книги «Цифровой синтез: практический курс» (М.: ДМК Пресс, 2020). Автор более 30 научных статей. Специализируется на методике преподавания компьютерных дисциплин студентам младших курсов.

Обо мне

Я работал в качестве архитектора программных систем, руководителя группы (team lead) и лектора начиная с 1999-го. В 2002 году я создавал тренинги для обучения. Я начал вести занятия с 2002 года. Моими первыми тренингами были занятия по управлению проприетарным программным обеспечением, но вскоре я начал обучать языкам Python и C++. В мое свободное время мне нравится писать статьи о C++, Python и Haskell. Также мне нравится выступать на различных конференциях. Каждую неделю я пишу что-то в моем англоязычном блоге ModernesCpp и в блоге на немецком языке[1], размещенном у Heise Developer.

С 2016 года я выступал в качестве независимого инструктора с семинарами по современному C++ и Python. Я написал несколько книг на разных языках о современном C++ и, в частности, по параллелизму. В связи с моей профессией я всегда ищу лучшие пути обучения современному C++.

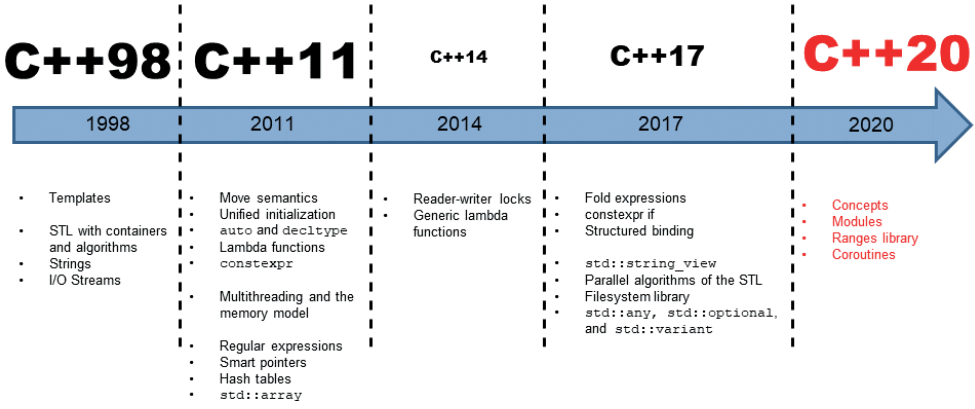
Райнер Гримм

[1] https://www.grimm-jaud.de/index.php/blog.

О ЯЗЫКЕ C++

1. Исторический контекст

C++20 – это следующий большой шаг после стандарта C++11. Как и C++11, C++20 меняет способ, как мы программируем, используя современный C++. Эти изменения произошли в основном вследствие добавления в язык таких понятий, как концепты (Concepts), модули (Modules), диапазоны (Ranges) и корутины (Coroutines). Для понимания данного шага в эволюции C++ позвольте мне сказать несколько слов об историческом контексте C++20.



История C++

1.1 C++98

В конце 80-х Бьярн Страуструп и Маргарет А. Эллис написали свою знаменитую книгу Annotated C++ Reference[1] (ARM). Данная книга выполняла две функции – определяла функциональность C++ в мире, в котором существуют различные реализации C++, и стала основой для первого стандарта C++98 (ISO/IEC 14882). В число важных возможностей языка вошли шаблоны и стандартная библиотека шаблонов (Standard Template Library, STL) со своими контейнерами, алгоритмами и строками, а также потоки ввода/вывода.

[1] https://stroustrup.com/arm.html.

1.2 C++03

Со стандартом C++03 (14882:2003) C++98 получил небольшое уточнение, настолько небольшое, что ему даже не выделено места на таймлайне. В сообществе C++03, включающий C++98, обычно называется **legacy C++**.

1.3 TR1

В 2005 году произошло очень важное событие. Был опубликован документ под названием Technical Report 1. TR1 был огромным шагом к C++11 и, соответственно, современному C++. TR1 (TR 19768) был основан на проекте Boost[1], который был создан членами комитета по стандартизации C++. TR1 содержит 13 библиотек, которые должны были стать частью C++11. В их число входят библиотека регулярных выражений, библиотека для работы со случайными числами, умные указатели и хеш-таблицы. А вот специальным математическим функциям пришлось ждать до C++17.

1.4 C++11

Мы называем C++11 *современным C++*. Это же название *современный C++* используется и для C++14 и C++17. C++11 внес много новых возможностей, которые принципиально изменили наше программирование на C++. Например, в C++11 вошли добавления из TR1, а также семантика перемещения (move semantics), форвардинг (perfect forwarding), вариадические шаблоны (variadic templates) и `constexpr`. Но это еще не все. С появлением стандарта C++11 мы также получили, причем в первый раз, модель памяти как основу работы с нитями и стандартизированный API для работы с нитями.

1.5 C++14

C++14 – это довольно небольшой стандарт. Он привнес read-write блокировки, обобщенные лямбда-функции и расширенные contsexpr-функции.

1.6 C++17

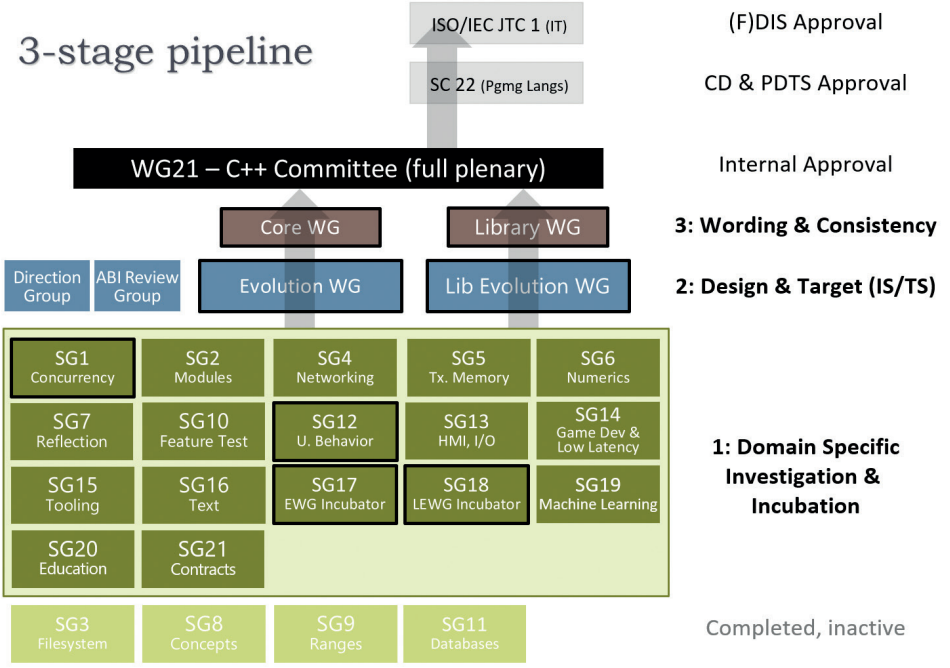
C++ не является ни большим, ни маленьким. Он привнес две выдающиеся возможности: параллельный STL и стандартный API для работы с файловой системой. Порядка 80 алгоритмов из стандартной библиотеки шаблонов могут выполняться параллельно или быть векторизованы. Как и с C++11, библиотеки boost оказали огромное влияние на C++17. Библиотеки boost предоставили библиотеку для работы с файловой системой и новые типы: `std::string_view`, `std::optional`, `std::variant` и `std::any`.

[1] https://www.boost.org.

2. Стандартизация

Процесс стандартизации C++ довольно демократичен. Комитет по стандартизации языка называется WG21 (Working Group 21) и был сформирован в 1990–1991 годах. Подкомитетами являются:

- организатор (Convener): возглавляет WG21, назначает график встреч и группы изучения;
- редактор проекта: применяет изменения к черновику стандарта C++;
- секретарь: назначает встречи WG21.



Группы стандартизации C++

Комитет организован в конвейер из трех стадий, состоящий из нескольких подгрупп.

2.1 Стадия 3

Стадия 3 предназначена для уточнения формулировок и обеспечения непротиворечивости предлагаемых добавлений и состоит из двух групп: формулировок по самому языку (core language wording, CWG) и формулировок по библиотекам (library wording).

2.2 Стадия 2

Стадия 2 состоит из двух групп: развитие самого языка (Core Language Evolution, EWG) и развитие библиотек (Library Evolution, LWEG). EWG и LWEG отвечают за новые возможности, включающие в себя расширения языка и стандартных библиотек соответственно.

2.3 Стадия 1

Стадия 1 предназначена для изучения направлений развития языка. Члены групп встречаются лицом к лицу, между встречами по телефону и видеоконференциями. Центральные группы могут изучать результаты целевых групп для обеспечения согласованности и последовательности изменений языка.

Ниже приводится список целевых групп (Study Groups):

- **SG1, Concurrency** – параллельные вычисления, включая модель памяти;
- **SG2, Modules** – темы, связанные с модулями;
- **SG3, File system** – файловая система;
- **SG4, Networking** – развитие сетевой библиотеки;
- **SG5, Transactional memory** – транзакционная память для включения в будущие релизы;
- **SG6, Numerics** – численные расчеты, числа с фиксированной точкой, числа с плавающей точкой и дроби;
- **SG7, Compile time programming** – программирование в контексте компиляции программы;
- **SG8, Concepts** – концепты;
- **SG9, Ranges** – диапазоны;
- **SG10, Feature test** – переносимые тесты для проверки того, поддерживает ли конкретная реализация конкретную возможность (feature);
- **SG11, Databases** – интерфейсы для взаимодействия с базами данных;
- **SG12, UB & Vulnerabilities** – улучшения, направленные борьбу с уязвимостями и неопределенным/незаданным поведением текущей версии стандарта языка;
- **SG13 HMI & I/O (Human/Machine Interface)** – поддержка устройств ввода/вывода;
- **SG14, Game development & low latency** – поддержка требования работы с небольшой задержкой;
- **SG15, Tooling** – инструменты для разработчиков, включая модули и пакеты;
- **SG16, Unicode** – обработка на C++ текста на юникоде;
- **SG17, EWG Incubator** – ранние обсуждения развития языка;

- **SG18, LWEG Incubator** – ранние обсуждения развития библиотек;
- **SG19, Machine Learning** – темы, связанные с искусственным интеллектом, и линейная алгебра;
- **SG20, Education** – материалы для обучающих курсов по C++;
- **SG21, Contracts** – поддержка языком обусловленного проектирования (Design by contract);
- **SG22 C/C++ Liason** – обсуждения взаимодействия C и C++.

Этот раздел дал вам краткий обзор стандартизации C++ и структуры комитета по стандартизации C++. Вы можете получить дополнительную информацию на сайте: https://isocpp.org/std.