

История развития языка программирования

C++

Подготовил: Акимов Н.

Павел

Что представляет из себя

С++ — компилируемый, статически
типизированный язык программирования общего
назначения.

Поддерживает такие парадигмы программирования, как:
процедурное программирование,
объектно-ориентированное программирование,
обобщённое программирование.

Язык имеет богатую стандартную библиотеку, которая
включает в себя распространённые контейнеры и алгоритмы,
ввод-вывод, регулярные выражения, поддержку
многопоточности и другие возможности. С++ сочетает свойства
как высокоуровневых, так и низкоуровневых языков

Область применения и отличие от Си

Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр

Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C

Общая структура и состав

Стандартная библиотека C++ включает в себя набор средств, которые должны быть доступны для любой реализации языка, чтобы обеспечить программистам удобное пользование языковыми средствами и создать базу для разработки как прикладных приложений самого широкого спектра, так и специализированных библиотек. Стандартная библиотека C++ включает в себя часть стандартной библиотеки C. Стандарт C++ содержит нормативную ссылку на стандарт C от 1990 года и не определяет самостоятельно те функции стандартной библиотеки, которые заимствуются из стандартной библиотеки C.

Доступ к возможностям стандартной библиотеки C++ обеспечивается с помощью включения в программу (посредством директивы `#include`) соответствующих стандартных заголовочных файлов. Всего в стандарте C++11 определено 79 таких файлов. Средства стандартной библиотеки объявляются как входящие в пространство имён `std`. Заголовочные файлы, имена которых соответствуют шаблону «сХ», где Х — имя заголовочного файла стандартной библиотеки C без расширения (`cstdlib`, `cstring`, `cstdio` и пр.), содержат объявления, соответствующие данной части стандартной библиотеки C. Стандартные функции библиотеки C также находятся в пространстве имён `std`.

Стандартная библиотека включает в себя следующие разделы:

- Поддержка языка. Включает средства, которые необходимы для работы программ, а также сведения об особенностях реализации. Выделение памяти, RTTI, базовые исключения, пределы значений для числовых типов данных, базовые средства взаимодействия со средой, такие как системные часы, обработка сигналов UNIX, завершение программы.
- Стандартные контейнеры. В стандартную библиотеку входят шаблоны для следующих контейнеров: динамический массив, статический массив, одно- и двунаправленные списки, стек, дек, ассоциативные, множества, очередь с приоритетом.
- Основные утилиты. В этот раздел входит описание основных базовых элементов, применяемых в стандартной библиотеке, распределителей памяти и поддержка времени и даты в стиле C.
- Итераторы. Обеспечивают шаблоны итераторов, с помощью которых в стандартной библиотеке реализуется стандартный механизм группового применения алгоритмов обработки данных к элементам контейнеров.
- Алгоритмы. Шаблоны для описания операций обработки, которые с помощью механизмов стандартной библиотеки могут применяться к любой последовательности элементов, в том числе к элементам в контейнерах. Также в этот раздел входят описания функций `bsearch()` и `qsort()` из стандартной библиотеки C.
- Строки. Шаблоны строк в стиле C++. Также в этот раздел попадает часть библиотек для работы со строками и символами в стиле C.
- Ввод-вывод. Шаблоны и вспомогательные классы для потоков ввода-вывода общего вида, строкового ввода-вывода, манипуляторы (средства управления форматом потокового ввода-вывода в стиле C++).
- Локализация. Определения, используемые для поддержки национальных особенностей и форматов представления (дат, валют и т. д.) в стиле C++ и в стиле C.
- Диагностика. Определения ряда исключений и механизмов проверки утверждений во время выполнения (`assert`). Поддержка обработки ошибок в стиле C.
- Числа. Определения для работы с комплексными числами, математическими векторами, поддержка общих математических функций, генератор случайных чисел.

Создание и развитие

C++



Язык C++, как развитие C, был создан в 1979г. Бьярни Страуструпом, который в то время начал работать в Bell Labs. Разработка и усовершенствование языка C++ потребовали от создателя больших усилий на протяжении 80-х и большей части 90-х годов.

Наконец, в 1998г. был принят ANSI/ISO-стандарт для языка C++. Говоря в самых общих словах, язык C++ является объектно-ориентированной, усовершенствованной версией языка C. C++ построен на основе версии C89, включающей все изменения 1995г., и теперь эту версию C89 называют C-подмножеством языка C++. Несмотря на то, что язык C++ был задуман как набор объектно-ориентированных расширений для языка C, вскоре он развился в самостоятельный язык программирования. В настоящее время его новые средства почти удвоили объем исходного языка. Вряд ли стоит доказывать, что C++ — один из самых мощных компьютерных языков, разработанных до сих пор.

В октябре 2011г. вышел новый стандарт языка C++, обозначаемый как C++11 или ISO/IEC 14882:2011. Полная его поддержка обещана в GCC 4.7.



Си и C++

- По мнению Линуса Торвальдса, «C++ провоцирует на написание ... значительного объёма кода, не имеющего принципиального значения с точки зрения функциональности программы»¹
- Поддержка ООП, шаблоны и STL не являются решающим преимуществом C++, так как всё, для чего они применяются, реализуемо и средствами Си. При этом устраняется раздувание кода, а некоторое усложнение, которое к тому же далеко не обязательно, компенсируется большей гибкостью, более простым тестированием, лучшими показателями производительности.
- Автоматизация доступа к памяти в C++ увеличивает затраты памяти и замедляет работу программ.
- Использование исключений C++ вынуждает следовать RAII, приводит к росту исполняемых файлов, замедлению программ. Дополнительные трудности возникают в параллельных и распределённых программах. Показательно, что стандарт кодирования на C++ компании Google прямо запрещает использование исключений.
- Код на C++ сложнее для понимания и тестирования, его отладка затрудняется использованием сложных иерархий классов с наследованием поведения и шаблонов. К тому же в средах программирования на C++ больше ошибок, как в компиляторах, так и в библиотеках.
- Многие детали поведения кода стандартом C++ не специфицированы, что ухудшает переносимость и может являться причиной трудно обнаруживаемых ошибок.
- Квалифицированных программистов на Си существенно больше, чем на C++.

ИСТОЧНИКИ

- <http://cppstudio.com/post/1980/>
- <http://victor192007.narod.ru/files/cpp00.html>
- https://ru.wikipedia.org/wiki/C%2B%2B#Стандартная_библиотека