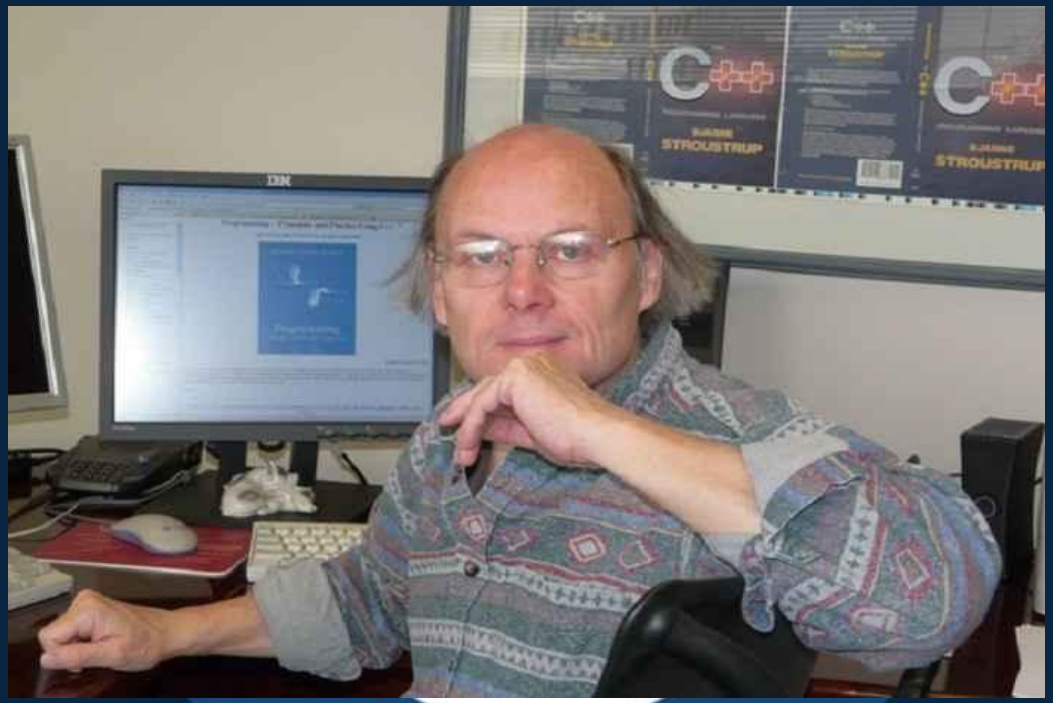
Язык

Канарейкин А. И.

Язык программирования C++ был создан в начале 1980-х годов, его создатель сотрудник фирмы Bell Laboratories — Бьёрн Страуструп.



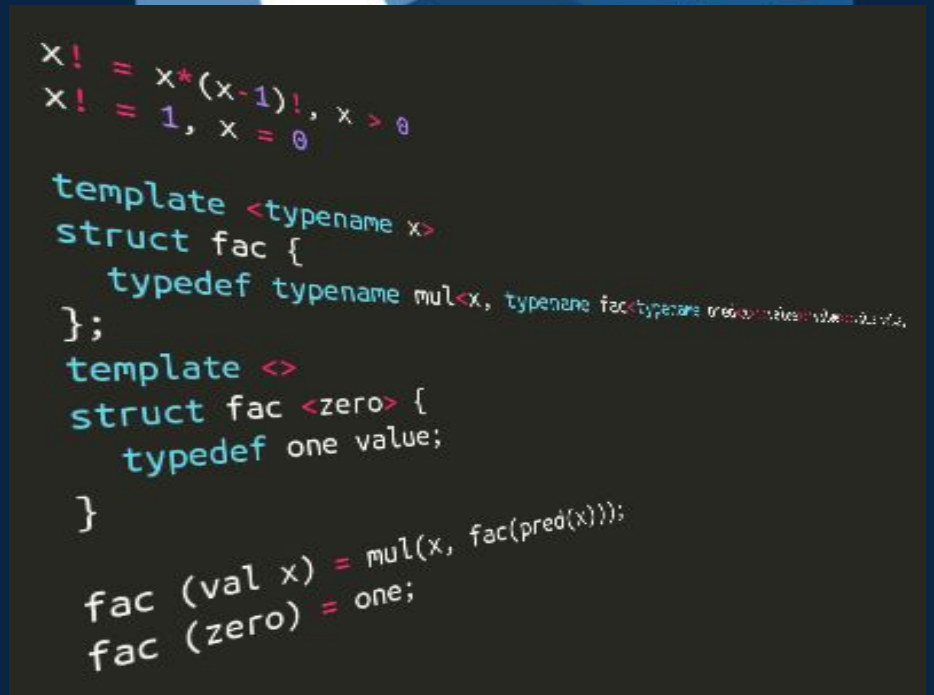
Он придумал ряд усовершенствований к языку программирования C, для собственных нужд. Т. е. изначально не планировалось создания языка программирования C++. Ранние версии языка C++, известные под именем «Си с классами», начали появляться с 1980 года. Язык C, будучи базовым языком системы UNIX, на которой работали компьютеры фирмы Bell, является быстрым, многофункциональным и переносимым. Страуструп добавил к нему возможность работы с классами и объектами, тем самым зародил предпосылки нового, основанного на синтаксисе C, языка программирования. Синтаксис C++ был основан на синтаксисе C, так как Бьёрн Страуструп стремился сохранить

При создании C++ Бьёрн Страуструп ставил цели:

- Получить универсальный язык со статическими типами данных, эффективностью и переносимостью языка C.
- Непосредственно и всесторонне поддерживать множество стилей программирования, в том числе процедурное программирование, абстракцию данных, объектно-ориентированное программирование и обобщённое программирование.
- Дать программисту свободу выбора, даже если это даст ему возможность выбирать неправильно.
- Максимально сохранить совместимость с C: любая конструкция, допустимая в обоих языках, должна в каждом из них обозначать одно и то же и приводить к одному и тому же поведению программы.
- Избегать особенностей, которые зависят от платформы или не являются универсальными.
- «Не платить за то, что не используется» — неиспользуемые языковые средства не должны приводить к снижению производительности программ.
- Не требовать сложной среды программирования.

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

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java и C#.



Общие направления развития C++

По мнению автора языка Бьёрна Страуструпа, говоря о дальнейшем развитии и перспективах языка, можно выделить следующее:
В основном дальнейшее развитие языка будет идти по пути внесения дополнений в стандартную библиотеку. Одним из основных источников этих дополнений является известная библиотека boost.
Изменения в ядре языка не должны приводить к снижению уже достигнутой эффективности C++. С точки зрения Страуструпа, предпочтительнее внесение в ядро нескольких серьёзных больших изменений, чем множества мелких правок.
Базовыми направлениями развития C++ на ближайшее время является расширение возможностей и доработка средств обобщённого программирования, стандартизация механизмов параллельной обработки, а также доработка средств безопасного программирования, таких как различные проверки и безопасные преобразования типов, проверка условий и так далее.
В целом C++ спроектирован и развивается как мультипарадигменный язык, впитывающий в себя различные методы и технологии программирования, но реализующий их на платформе, обеспечивающей высокую техническую эффективность. Поэтому в будущем не исключено добавление в язык средств функционального программирования, автоматической сборки мусора и других отсутствующих в нём сейчас механизмов. Но в любом случае это будет делаться на имеющейся платформе высокоэффективного компилируемого языка.

Стандартная библиотека C++

Язык C++ содержит обширную стандартную библиотеку.

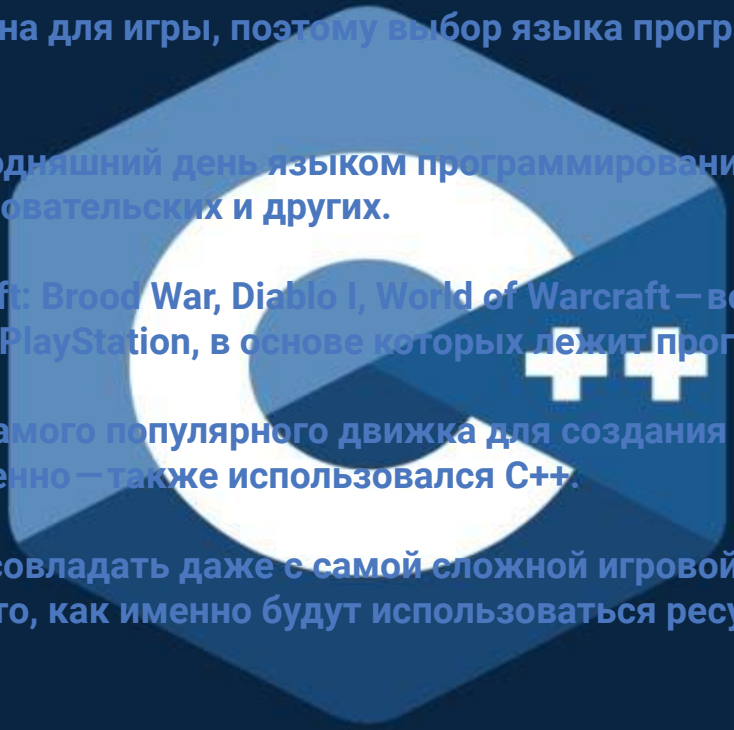
Основные части библиотеки следующие:

Ввод-вывод описан в заголовочных файлах iostream, fstream и других.
Работа со строками описана в файле string и других.
Контейнеры (структуры данных) описаны в большом числе заголовочных файлов в соответствии с типом контейнера. Например, vector - динамический массив, set - множество с возможностью быстрого добавления, удаления, поиска элементов, map - ассоциативный массив (словарь), list - двусвязный список, stack - стек, queue - очередь.
Алгоритмы (линейный и двоичный поиск, нахождение следующей перестановки, случайная перестановка) описаны в заголовочном файле algorithm.
Первоначально часть стандартной библиотеки называлась STL - Standard Template Library и развивалась независимо от языка C++ компаниями HP и SGI. Затем она была добавлена в стандарт языка, но название STL сохранилось и часто употребляется применительно к той части библиотеки, которая относится к контейнерам и алгоритмам.

Разработка игр и игрового движка

Производительность крайне важна для игры, поэтому выбор языка программирования здесь ограничен.

Являясь самым быстрым на сегодняшний день языком программирования, C++ оказывается одним из лучших для игр в 3D, многопользовательских и других.

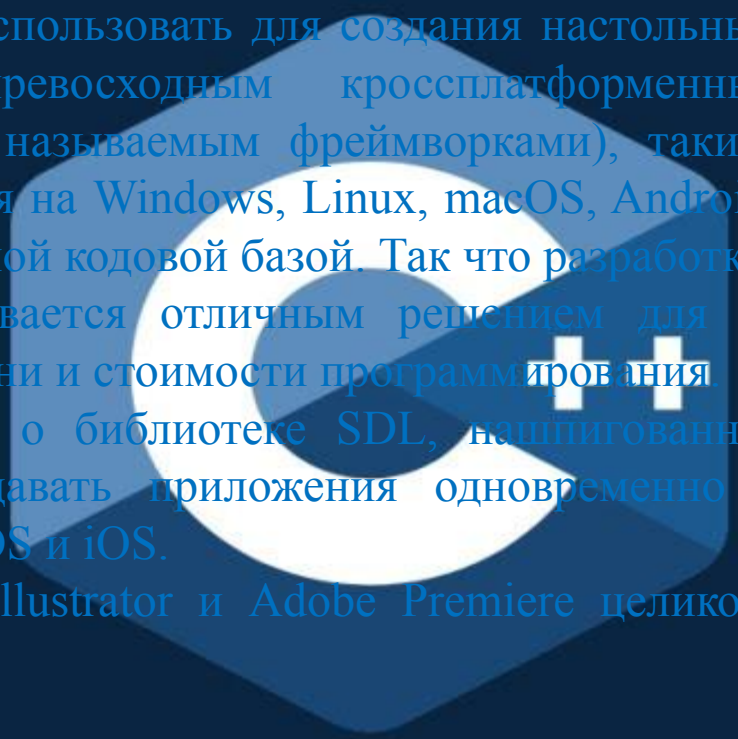


Например, Counter-Strike, StarCraft: Brood War, Diablo I, World of Warcraft — все эти игры написаны на C++. Не говоря уже о консолях Xbox и PlayStation, в основе которых лежит программирование C++.

В ядре игрового движка Unity — самого популярного движка для создания видеоигр под несколько операционных систем одновременно — также использовался C++.

Средства разработки C++ могут совладать даже с самой сложной игровой графикой. Они позволяют оптимизировать и регулировать то, как именно будут использоваться ресурсы памяти и структуры данных в игре.

Разработка настольных и кроссплатформенных приложений



C++ также можно использовать для создания настольных приложений. Всё благодаря превосходным кроссплатформенным средствам разработки (иногда называемым фреймворками), таким как Qt. Оно позволяет нацелиться на Windows, Linux, macOS, Android и встроенные системы—все с единой кодовой базой. Так что разработка приложений с помощью Qt оказывается отличным решением для тех, кто хочет сэкономить на времени и стоимости программирования.

Стоит упомянуть и о библиотеке SDL, нашпигованной функциями, позволяющими создавать приложения одновременно для Windows, Linux, Android, MacOS и iOS.

Кстати, Photoshop, Illustrator и Adobe Premiere целиком написаны на C++.

Достоинства и недостатки

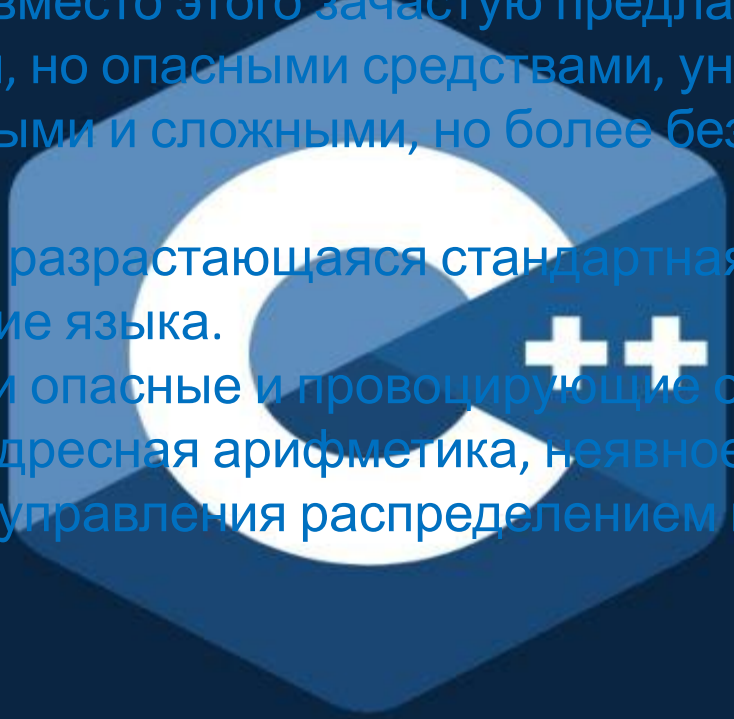
Достоинства

C++ содержит средства разработки программ контролируемой эффективности для широкого спектра задач, от низкоуровневых утилит и драйверов до весьма сложных программных комплексов. В частности:
Доступность. Для C++ существует огромное количество учебной литературы, переведённой на всевозможные языки. Язык имеет высокий порог вхождения, но среди всех языков такого рода обладает наиболее широкими возможностями.
Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Один из базовых принципов C++ — «не платишь за то, что не используешь» — то есть ни одна из языковых возможностей, приводящая к дополнительным накладным расходам, не является обязательной для использования. Имеется возможность работы с памятью на низком уровне.

Достоинства и недостатки

Недостатки

К числу обычно упоминаемых недостатков языка можно отнести:
Язык не поощряет создание надёжного, легко читаемого и удобного в сопровождении кода, вместо этого зачастую предлагая выбор между короткими и простыми, но опасными средствами, унаследованными от Си, и новыми, объёмными и сложными, но более безопасными механизмами.

Сложная и постоянно разрастающаяся стандартная библиотека, затрудняющая изучение языка.

Унаследованные от Си опасные и провоцирующие ошибки возможности (макроопределения, адресная арифметика, неявное приведение типов, возможность прямого управления распределением памяти).