

Введение в C++

C++ сочетает в себе три различных принципа программирования:

- Процедурное программирование, представленное языком C и позволяющее создавать библиотеки функций;
- Объектно-ориентированное программирование (ООП), представленное таким понятием как класс и позволяющее разрабатывать библиотеки классов;
- Обобщенное программирование, представленное шаблонами языка C++ и его стандартными библиотеками.



Язык C++ был создан в начале 80 годов в компании Bell Laboratories Бьерном Страуструпом с целью повышения эффективности языка программирования. Страуструп выбрал в качестве основы язык C, так как он был кратким, хорошо подходил для системного программирования и был широко доступен. Страуструп добавил в язык C элементы ООП и обобщенного программирования, не меняя при этом существенно сам язык C. Таким образом, язык C++ был разработан как расширение языка C.



Философия C++



- I. *Получить универсальный язык со статическими типами данных, эффективностью и переносимостью языка C.*
- II. *Непосредственно и всесторонне поддерживать множество стилей программирования. Дать программисту свободу выбора, даже если это даст ему возможность выбирать неправильно.*



-
- III. Избегать особенностей, которые зависят от платформы или не являются универсальными.
 - IV. Избегать разногласий между C и C++
 - V. Не требовать слишком усложнённой среды программирования.
 - VI. Не платить за то, что не используется.

Достоинства

- **Высокая совместимость с языком Си** : код на Си может быть с минимальными переделками скомпилирован компилятором C++.
- **Как следствие предыдущего пункта — вычислительная производительность.** Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Имеется возможность работы с памятью на низком уровне.
- **Поддержка различных стилей программирования:** традиционное императивное программирование (структурное, объектно-ориентированное), обобщённое программирование, функциональное программирование.



- Автоматический вызов деструкторов объектов в адекватном порядке (обратном вызову конструкторов) упрощает и повышает надёжность управления памятью и другими ресурсами (открытыми файлами, сетевыми соединениями, соедине



- Шаблоны C++ дают возможность построения обобщённых контейнеров и алгоритмов для разных типов данных. Попутно шаблоны дают возможность производить вычисления на этапе компиляции.
- Возможность встраивания предметно-ориентированных языков программирования в основной код.
- Доступность. Язык имеет высокий порог вхождения, но среди всех языков такого рода обладает наиболее широкими возможностями.
- Перегрузка операторов позволяет кратко и ёмко записывать выражения над пользовательскими типами в естественной алгебраической форме.



Недостатки



- Отсутствие системы модулей. С++ унаследовал от Си подключение заголовочных файлов с помощью препроцессора. Это вынуждает дублировать описания объектов, порождает неочевидные требования к коду и увеличивает объём компилируемого текста, а значит и время компиляции.
- Наличие более чем одного механизма для выполнения одних и тех же задач, что усложняет язык и приводит к неоптимальному и небезопасному кодированию.
- Шаблоны порождают объёмный и не всегда оптимальный код. Частичное определение шаблонов усложняет как сам

- Сложный синтаксис и объёмная спецификация языка затрудняют его изучение.
- 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.

• Сложна



блиотека,

Программы, созданные с помощью с/с++



DOOM



unity



CRYENGINE