

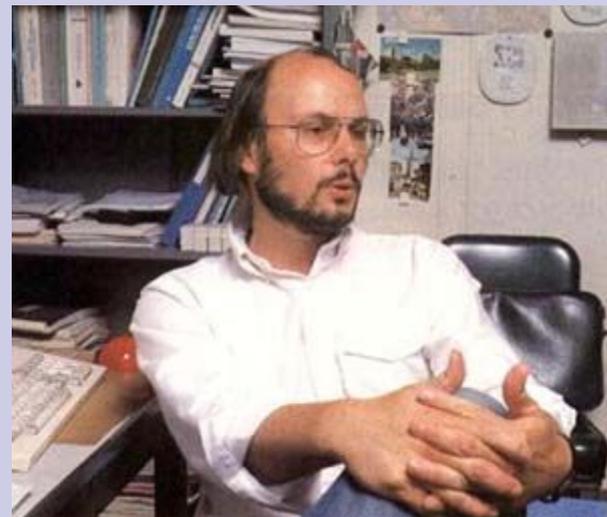
Язык программирования C++

СОЗДАНИЕ И РАЗВИТИЕ ЯЗЫКА C++

Язык возник в начале 1980-х годов, когда сотрудник фирмы Bell Laboratories Бьёрн Страуструп (1950) придумал ряд усовершенствований к языку C (Си) под собственные нужды.

До начала официальной стандартизации язык развивался в основном силами Страуструпа в ответ на запросы программистского сообщества.

В 1998 году был ратифицирован международный стандарт языка C++: ISO/IEC 14882:1998 «Standard for the C++ Programming Language».



	
ISO/IEC JTC 1/SC 22 N 4411	2008-10-09
ISO/IEC JTC 1/SC 22 Programming Languages	
Document Type:	Text for CD Ballot
Document Title:	ISO/IEC 14882, Programming language – C++
Document Source:	WG 21 Convener (P.J. Plauger)
Document Status:	This document is circulated to National Bodies for ballot. Please submit your vote via the online balloting system by the due date indicated.
Action ID:	
Due Date:	2009-01-10
No. of Pages:	1314

До недавнего времени действовала версия стандарта ISO/IEC 14882:2003

После принятия технических исправлений к стандарту в 2011 году действует нынешняя версия этого стандарта — ISO/IEC 14882:2011.

Ведется работа над C++14

СОЗДАНИЕ И РАЗВИТИЕ ЯЗЫКА C++

Принципы Страуструпа, положенные в основу концепции языка:

- Универсальный язык со статическими типами данных, эффективностью и переносимостью языка C.
- Поддержка процедурного программирования, абстракции данных, объектно-ориентированного программирования и обобщённого программирования.
- Свобода выбора программиста, даже если это даст ему возможность выбирать неправильно.
- Максимальная совместимость с C.
- Отсутствие различий между C и C++: любая конструкция, которая допустима в обоих этих языках, должна в каждом из них обозначать одно и то же и приводить к одному и тому же поведению программы.
- Отсутствие особенностей, которые зависят от платформы или не являются универсальными.
- Никакое языковое средство не должно приводить к снижению производительности программ, не использующих его.
- Не требовать слишком усложнённой среды программирования.

Сравнение языков семейства C

Характеристика	C	C++	C#	Описание
Императивный	+	+	+	Императивный язык должен описывать не столько саму задачу, сколько её решение
ООП	-	+	+	Использует три парадигмы ООП: наследование, инкапсуляцию и полиморфизм
Функциональный	-	-/+	+/-	Позволяет записывать программу как композицию функций. В чистом функциональном языке нет переменных
Рефлексивный	-	-	-/+	Возможность программы на данном языке оперировать собственным кодом как данными
Обобщенное программирование	-	+	+	Обобщенное программирование позволяет записывать алгоритмы, принимающие данные любого типа
Логический	-	-	-	Использует логику предикатов для описания баз данных и процедур логического вывода и принятия решений
Декларативный	-	-	-/+	Декларативный язык описывает не столько решение проблемы, сколько саму проблему, а решение уже должен определять компилятор
Разделительный	+/-	+/-	-/+	Содержит специальные конструкции для поддержки распараллеливания программы на несколько компьютеров или процессоров
Статическая типизация	+	+	+	Переменные и параметры методов/функций связываются с типами в момент объявления и не могут быть изменены позже

Характеристика	C	C++	C#	Описание
Динамическая типизация	-	-/+	-/+	Переменные и параметры методов/функций связываются с типами в момент присваивания значения, а не в момент объявления переменной или параметра.
Явная типизация	+	+	+	Типы переменных и параметров указываются явно
Неявная типизация	-	-/+	-/+	Типы переменных и параметров не указываются явно
Поддержка try/catch	-	+	+	Поддержка обработки исключений с помощью try/catch или эквивалентной конструкции
Динамические массивы	-	+/-	+/-	Наличие встроенных в язык массивов способных изменять свой размер
Контроль границ массивов	-	-/+	+	Определение допустимости текущего значения индекса
Множественное наследование	x	+	-	Возможность наследовать класс сразу от нескольких классов
Макросы	-/+	-/+	-	Обработка кода программы до времени её компиляции и/или выполнения
Перегрузка функций	-	+	+	Возможность перегрузки функций/методов по количеству и типам параметров
Значения параметров по умолчанию	-	+	+	Возможность при вызове функции/метода опускать некоторые параметры, чтобы при этом подставлялось значение по умолчанию, указанное при определении функции

КЛЯТВА СТРАУСТРУПА

"Я обязуюсь прилежно комментировать свой код, не использовать `goto` и следить за состоянием своих потоков и выделением памяти. Я обязуюсь не оставлять мусора в системе и избегать однобуквенных переменных.

Ресет"

