ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ МЕЖДУНАРОДНОГО УНИВЕРСИТЕТА КЫРГЫЗСКОЙ РЕСПУБЛИКИ

РЕФЕРАТ

По предмету СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Тема: **“История развития языка C++”**

Группа: ПКС 2-19

Студентка: Бактыгулова Мириада

Преподаватель: Тихонов Иван Георгиевич

БИШКЕК 2022

План

1. Язык С и UNIX
2. C мобильный язык ассемблера
3. С - структурный язык программирования
4. Стандартизация языка С
5. Рождение С++
6. Откровения Бьерна Страуструп
7. Почему С++
8. Зачем был нужен С++
9. Настоящий объектно-ориентированный C++
10. Литература

Язык С и UNIX

Язык программирования *С* (читается "Си") создан в начале 70-х годов, когда Кен Томпсон и Дэннис Ритчи из Bell Labs разрабатывали операционную систему UNDC. Сначала они создали часть компилятора *С*, затем использовали ее для компиляции остальной части компилятора *С* и, наконец, применили полученный в результате компилятор для компиляции UNIX. Операционная система UNIX первоначально распространялась в исходных кодах на *С* среди университетов и лабораторий, а получатель мог откомпилировать исходный код на языке *С* в машинный код с помощью подходящего компилятора *С*.

Распространение исходного кода сделало операционную систему UNIX уникальной; программист мог изменить операционную систему, а исходный код мог быть перенесен с одной аппаратной платформы на другую. Сегодня стандарт POSIX определяет стандартный набор системных вызовов UNIX, доступных в *С*, которые должны быть реализованы в версиях UNIX, являющихся POSIX-совместимыми. С был третьим языком, который разработали Томсон и Ритчи в процессе создания UNIX; первыми двумя были, разумеется, *А* и *В*.

C мобильный язык ассемблера

По сравнению с более ранним языком — BCPL, язык *С* был улучшен путем добавления типов данных определенной длины. Например, тип данных int мог применяться для создания переменной с определенным числом битов (обычно 16), в то время как тип данных long мог использоваться для создания целой переменной с большим числом битов (обычно 32). В отличие от других языков высокого уровня, *С* мог работать с адресами памяти напрямую с помощью указателей и ссылок. Поскольку *С* сохранил способность прямого доступа к аппаратному обеспечению, его часто относят к языкам среднего уровня или в шутку называют "мобильным языком ассемблера".

С - структурный язык программирования

Что касается грамматики и синтаксиса, то *С* является структурным языком программирования. В то время как многие современные программисты мыслят в категориях классов и объектов, программисты на *С* думают в категориях процедур и функций. В *С* можно определить собственные абстрактные типы данных, используя ключевое слово struct. Аналогично можно описывать собственные целые типы (перечисления) и давать другие названия существующим типам данных при помощи ключевого слова typedef. В этом смысле *С* является структурным языком с зародышами объектно-ориентированного программирования.

Стандартизация языка С

Широкое распространение языка *C* на различных типах компьютеров (иногда называемых аппаратными платформами) привело, к сожалению, ко многим вариациям языка. Они были похожи, но несовместимы друг с другом. Это было серьезной проблемой для разработчиков программ, нуждавшихся в написании совместимых программ, которые можно было бы выполнять на нескольких платформах. Стало ясно, что необходима стандартная версия *C*. В 1983г. ANSI (Американский Национальный Комитет Стандартов) сформировал технический комитет X3J11 для создания стандарта языка *C* (чтобы "обеспечить недвусмысленное и машинно-независимое определение языка"). В 1989 стандарт был утвержден. ANSI скооперировался с ISO (Международной Организацией Стандартов), чтобы стандартизовать *C* в международном масштабе; совместный стандарт был опубликован в 1990 году и назван ANSI/ISO 9899:1990. Этот стандарт совершенствуется до сих пор и поддерживается большинством фирм разработчиков компиляторов.

Рождение С++

Бьерн Страуструп высвободил объектно-ориентированный потенциал *С* путем перенесения возможностей классов *Simula 67* в *С*. Первоначально новый язык носил имя "С с классами" и только потом стал называться *C++*. Язык *C++* достиг популярности, будучи разработанным в Bell Labs, позже он был перенесен в другие индустрии и корпорации. Сегодня это один из наиболее популярных (и авторитетных) языков программирования в мире. *C++* наследует как хорошие, так и плохие стороны *С*.

Откровения Бьерна Страуструп

Бьерн Страуструп: "Я придумал *C++*, записал его первоначальное определение и выполнил первую реализацию. Я выбрал и сформулировал критерии проектирования *C++*, разработал его основные возможности и отвечал за судьбу предложений по расширению языка в комитете по стандартизации *C++*", - пишет автор самого популярного языка программирования. - "Язык *C++* многим обязан языку *C*, и язык *C* остается подмножеством *языка C++* (но в C++ устранены несколько серьезных брешей системы типов C). Я также сохранил средства *C*, которые являются достаточно низкоуровневыми, чтобы справляться с самыми критическими системными задачами. Язык *C*, в свою очередь многим обязан своему предшественнику, BCPL; кстати, стиль комментариев // был взят в *C++* из BCPL. Другим основным источником вдохновения был язык *Simula67*. Концепция классов (с производными классами и виртуальными функциями) была позаимствована из него. Средства перегрузки операторов и возможность помещения объявлений в любом месте, где может быть записана инструкция, напоминает *Algol68*. "

Почему С++

Название *C++* выдумал Рик Масситти. Название указывает на эволюционную природу перехода к нему от *C*. "++" - это операция приращения в *C*. Чуть более короткое имя *C+* является синтаксической ошибкой; кроме того, оно уже было использовано как имя совсем другого языка. Знатоки семантики *C* находят, что *C++* хуже, чем *++C*. Названия *D* язык не получил, поскольку он является расширением C и в нем не делается попыток исцеляться от проблем путем выбрасывания различных особенностей.

Зачем был нужен С++

Изначально *язык программирования C++* был разработан, чтобы автору и его друзьям не приходилось программировать на ассемблере, C или других современных языках высокого уровня. Основным его предназначением было сделать написание хороших программ более простым и приятным для отдельного программиста. Плана разработки *C++* на бумаге никогда не было; проект, документация и реализация двигались одновременно. Разумеется, внешний интерфейс *C++* был написан на *C++*. Никогда не существовало "Проекта C++" и "Комитета по разработке C++". Поэтому *C++* развивался и продолжает развиваться во всех направлениях, чтобы справляться со сложностями, с которыми сталкиваются пользователи, а также в процессе дискуссий автора с его друзьями и коллегами.

Настоящий объектно-ориентированный C++

В *языке С++* полностью поддерживаются принципы объектно-ориентированного программирования, включая три кита, на которых оно стоит: инкапсуляцию, наследование и полиморфизм. Инкапсуляция в *С++* поддерживается посредством создания нестандартных (пользовательских) типов данных, называемых классами. Язык *С++* поддерживает наследование. Это значит, что можно объявить новый тип данных (класс), который является расширением существующего.

Хотя *язык программирования С++* справедливо называют продолжением *С* и любая работоспособная программа на языке *С* будет поддерживаться компилятором *С++*, при переходе от *С* к *С++* был сделан весьма существенный скачок. Язык *С++* выигрывал от своего родства с языком *С* в течение многих лет, поскольку многие программисты обнаружили, что для того, чтобы в полной мере воспользоваться преимуществами *языка С++*, им нужно отказаться от некоторых своих прежних знаний и приобрести новые, а именно: изучить новый способ концептуальности и решения проблем программирования. Перед тем как начинать осваивать *С++*, Страуструп и большинство других программистов, использующих *С++* считают изучение языка *С* необязательным.

Язык программирования *C++* в настоящее время считается господствующим языком, используемым для разработки коммерческих продуктов, 90% игр пишутся на *С++* с применением DirectX.

Литература

Источники :

* Х.М.Дейтел, П.Дж.Дейтел "Как программировать на C++"
* Бьерн Страуструп "Язык программирования C++. 3-е издание."
* Симон Робинсон, Олли Корнес, Джей Глинн и др. "C# для профессионалов"
* Джесс Либерти "Освой самостоятельно С++ за 21 день"
* Станислав Горнаков "DirectX, уроки программирования на С++"

