История развития языков программирования​

Прогресс компьютерных технологий определил процесс появления новых разнообразных знаковых систем для записи алгоритмов языков программирования. Смысл появления такого языка — упрощение программного кода.

С каждым днём наш мир становится более мобильным и информационным. Всё больше и больше компьютеры вступают в нашу повседневную жизнь, и чтобы облегчить наше общение с ними создаётся новое ПО с помощью различных языков программирования.

Языки программирования принято делить на пять поколений. В первое поколение входят языки, созданные в начале 50-х годов, когда первые компьютеры только появились на свет. Это был первый язык ассемблера, созданный по принципу «одна инструкция — одна строка». Расцвет второго поколения языков программирования пришелся на конец 50-х — начало 60-х годов. Тогда был разработан символический ассемблер, в котором появилось понятие переменной. Он стал первым полноценным языком программирования. Благодаря его возникновению заметно возросли скорость разработки и надежность программ. Появление третьего поколения языков программирования принято относить к 60-м годам. В это время родились универсальные языки высокого уровня, с их помощью удается решать задачи из любых областей. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера и возможность использования мощных синтаксических конструкций, позволили резко повысить производительность труда программистов. Понятная большинству пользователей структура этих языков привлекла к написанию небольших программ (как правило, инженерного или экономического характера) значительное число специалистов из некомпьютерных областей. Подавляющее большинство языков этого поколения успешно применяется и сегодня.

С начала 70-х годов по настоящее время продолжается период языков четвертого поколения. Эти языки предназначены для реализации крупных проектов, повышения их надежности и скорости создания. Они обычно ориентированы на специализированные области применения, где хороших результатов можно добиться, используя не универсальные, а проблемно-ориентированные языки, оперирующие конкретными понятиями узкой предметной области. Как правило, в эти языки встраиваются мощные операторы, позволяющие одной строкой описать такую функциональность, для реализации которой на языках младших поколений потребовались бы тысячи строк исходного кода.

Рождение языков пятого поколения произошло в середине 90-х годов. К ним относятся также системы автоматического создания прикладных программ с помощью визуальных средств разработки, без знания программирования. Главная идея, которая закладывается в эти языки, — возможность автоматического формирования результирующего текста на универсальных языках программирования (который потом требуется откомпилировать). Инструкции же вводятся в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием.

**ENIAC** — первый электронный цифровой компьютер общего назначения, который можно было перепрограммировать. 14 февраля ENIAC был впервые показан публике.

В конце 1945 г., когда компьютер был наконец собран и готов к проведению первого испытания, война, нуждам которой он был призван служить, окончилась. Поэтому в качестве испытания первой была поставлена задача по моделированию термоядерного взрыва супербомбы. В дальнейшем компьютер использовали в основном для прогнозирования погоды, в том числе предсказания направления выпадения ядерных осадков.

Поражают характеристики компьютера: он весил 27 тонн, потреблял 174 кВт энергии, а на его создание ушло более чем 200 000 человеко-часов и 500 000 долларов США. Всего комплекс включал в себя 17 468 ламп 16 различных типов, 7200 кремниевых диодов, 1500 реле, 70 000 резисторов и 10 000 конденсаторов.

Всего существовало 1,8 миллиарда вариантов отказа в каждую секунду — до этого человечество не создавало ни один прибор такой сложности и с таким требованием к надёжности.

**Assembler** — язык программирования низкого уровня, представляющий собой формат записи машинных команд, удобный для восприятия человеком.

Язык ассемблера люди начали использовать с 1950 года. Этот язык первым позволил отображать двоичный код в более удобной для человека форме: в виде букв или укороченных слов, которые примерно обозначали сущность команды. Умея разбираться в распечатке ассемблера, можно находить ошибки в программах, созданных даже при помощи других языков. В данный момент у каждого из нас дома стоят много «интерпретаторов ассемблера» — меню в телевизорах, программы стиральных машин разработаны зачастую именно на нем. (хотя ленивые программисты пишут программы для микроконтроллеров и на C). Кроме того, на ассемблере написаны драйверы устройств, библиотеки и функции программ, написанных даже на других языках. Кроме того, на ассемблере создаются целые операционные системы.

Данный тип языков получил свое название от названия транслятора (компилятора) с этих языков — ассемблера (англ. assembler — сборщик). Название обусловлено тем, что на первых компьютерах не существовало языков более высокого уровня, и единственной альтернативой созданию программ с помощью ассемблера было программирование непосредственно в кодах.

Директивы ассемблера позволяют включать в программу блоки данных (описанные явно или считанные из файла); повторить определённый фрагмент указанное число раз; компилировать фрагмент по условию; задавать адрес исполнения фрагмента, менять значения меток в процессе компиляции; использовать макроопределения с параметрами и др.

**PASCAL**

Pascal — один из наиболее известных языков программирования разработанный Никлаусом Виртом в 1967-1971, изначально стал популярным в основном благодаря относительно простому синтаксису. Язык назван в честь французского математика, физика, литератора и философа Блеза Паскаля, который создал одну из первых в мире механических машин, складывающую два числа.

Выпуск компанией Borland удобного и простого компилятора Turbo Pascal позволил ему занять лидирующие позиции и до сих пор изучаться в школах и институтах, хотя и как основы программирования, а не идеального языка для создания программ.

Особенностями языка являются строгая типизация и наличие средств структурного (процедурного) программирования. Паскаль был одним из первых таких языков. По мнению Вирта, язык должен способствовать дисциплинированному программированию, поэтому, наряду со строгой типизацией, в Паскале сведены к минимуму возможные синтаксические неоднозначности, а сам синтаксис автор постарался сделать интуитивно понятным даже при первом знакомстве с языком.

**C**

Язык программирования Си был разработан в 1972 году Деннисом Ритчи, в компании Bell Laboratories. Название происходит от номера проекта лаборатории. Изначально этот язык задумывался как промежуточный между языками высокого и низкого уровней, но продуктивность и компактность кода + преимущества структурного языка, несмотря на сложность обучения сделали Си самым популярным языком.

После появления язык был хорошо принят, потому что он позволял быстро создавать компиляторы для новых платформ, а также позволял программистам довольно точно представлять, как выполняются их программы. Благодаря близости к языкам низкого уровня программы на Си работали эффективнее написанных на многих других языках высокого уровня, и лишь оптимизированный вручную код на ассемблере мог работать ещё быстрее, потому что давал полный контроль над машиной. На сегодняшний день развитие компиляторов и усложнение процессоров привело к тому, что вручную написанный ассемблерный код (кроме разве что очень коротких программ) практически не выигрывает по сравнению с кодом, генерируемым компиляторами, при этом Си продолжает оставаться одним из наиболее эффективных языков высокого уровня.

В наше время этот язык наиболее популярен и универсален, что доказывает создание именно на ней всемирной ОС — Windows.

Сейчас Си считается основой для разработки современных больших и сложных проектов. Конечно, как и у всего существующего в природе, у него имеются и слабые стороны, вытекающие из требований эффективности.

Помимо «Окон» на Си написаны такие известные программы как Outlook и Opera.

**C++**

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

C++ компилируемый язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования. В сравнении с его предшественником, языком программирования Cи, наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования. Название «язык программирования C++» происходит от языка программирования C, в котором унарный оператор ++ обозначает инкремент переменной.

Язык программирования C++ широко используется для разработки программного обеспечения. А именно, создание разнообразных прикладных программ, разработка операционных систем, драйверов устройств, а также видео игр и многое другое.

По словам Бьерна Страуструпа, он должен «укорить написание хороших программ и сделать этот процесс наиболее приятным для каждого отдельно взятого программиста».

**Python**

Python (в русском языке распространено название питон) — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой набор полезных функций.

Создание языка Python начиналось довольно медленно и неуверенно. Главным энтузиастом, который в 1990 году пытался воплотить Python в реальность, стал Гвидо Ван Россум. Именно этот человек, работая над разработкой языка ABC в Голландском институте CWI, понял, что хотел бы создать нечто новое. Это послужило стартом для написания нового интерпретатора; конечно, не без использования некоторых идей, взятых с ABC.

В 1996 году, когда данный проект набирал критическую массу, к разработке подключился Стив Маевский, который был довольно известным в сети, так как вел свой блог «Сравнительная критика языков программирования».

Уже в 1991 году стали появляться первые средства ООП разработки.

Сейчас Python поддерживает структурное, обобщенное, объектно-ориентированное, функциональное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Python — активно развивающийся язык программирования, новые версии с добавлением/изменением языковых свойств выходят примерно раз в два с половиной года. Язык не подвергался официальной стандартизации, роль стандарта де-факто выполняет CPython, разрабатываемый под контролем автора языка.

**Java**

Созданная компанией Sun Microsystems система разработки Java безопасна и высокопроизводительна. Java — объектно-ориентированный язык, удобный и надёжный в эксплуатации благодаря таким своим достоинствам, как многозадачность, поддержка протоколов Internet и многоплатформенность. Java — это интерпретируемый язык, и каждая Java-программа компилируется для гипотетической машины, называемой Виртуальная Машина Java. Результатом такой компиляции является байт-код Java, который в свою очередь может выполняться на любой операционной системе при условии наличия там системы времени выполнения Java, которая интерпретирует байт-код в реальный машинный код конкретной системы.

Язык Java является объектно-ориентированным и поставляется с достаточно объемной библиотекой классов. Библиотеки классов Java значительно упрощают разработку приложений, предоставляя в распоряжение программиста мощные средства решения распространенных задач.

На Java написан движок известного эмулятора «Ил-2: Штурмовик».

**C#**

Язык программирования С# был разработан в 1993—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft.

C# — это язык с C-подобным синтаксисом. Здесь он близок в этом отношении к C++ и Java.

Будучи объектно-ориентированным языком, он много перенял у Java и С++. Как и Java, C# изначально предназначался для веб-разработки, и примерно 75% его синтаксических возможностей такие же, как у Java. C# также называют «очищенной версией Java». Ещё 10% наш герой позаимствовал из C++ и 5% – из Visual Basic. Оставшиеся 10% C# — это реализация собственных идей разработчиков. Объектно-ориентированный подход позволяет строить с помощью C# крупные, но в то же время гибкие, масштабируемые и расширяемые приложения.

При этом он всё ещё активно развивается, и с каждой новой версией появляется всё больше интересного — например лямбды, динамическое связывание, асинхронные методы и т.д.

Инструментарий C# позволяет решать широкий круг задач, язык действительно очень мощный и универсальный. На нём часто разрабатывают: веб-приложения, игры, мобильные приложения для Android или iOS, программы под Windows.

Перечень возможностей разработки практически не имеет ограничений благодаря широчайшему набору инструментов и средств. Конечно, всё это можно реализовать при помощи других языков. Но некоторые из них узкоспециализированные, а в некоторых придётся использовать дополнительные инструменты сторонних разработчиков. В C# решить широкий круг задач возможно быстрее, проще и с меньшими затратами времени и ресурсов.

**Swift**

Swift — открытый мультипарадигмальный компилируемый язык программирования общего назначения. Создан компанией Apple в первую очередь для разработчиков iOS и macOS. Swift работает с фреймворками Cocoa и Cocoa Touch и совместим с основной кодовой базой Apple, написанной на Objective-C. Swift задумывался как более лёгкий для чтения и устойчивый к ошибкам программиста язык, нежели предшествовавший ему Objective-C. Программы на Swift компилируются при помощи LLVM, входящей в интегрированную среду разработки Xcode 6 и выше. Swift может использовать рантайм Objective-C, что делает возможным использование обоих языков (а также С) в рамках одной программы.

Старший вице-президент по разработке программного обеспечения Apple Крейг Федериги во время анонса этого продукта заявил, что язык программирования Swift был заложен ещё в платформе NeXT (ОС NeXTSTEP выпускалась в 1989—1995 годах), которая стала основой для современной macOS, а затем и iOS.

Разработка текущего варианта языка Swift была начата в 2010 году Крисом Латтнером, руководителем отдела разработки инструментов для создания программного обеспечения Apple и одним из основных разработчиков LLVM. Swift заимствовал идеи из «Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, и ещё из стольких многих языков, что сложно перечислить»

2 июня 2014 года на конференции WWDC Swift был официально представлен вместе с бесплатным руководством по использованию языка объёмом в 500 страниц, доступным на сервисе «iBook Store».

8 июня 2015 года компания Apple объявила о выпуске новой версии Swift 2.0, которая получила более высокую производительность, новое API обработки ошибок, улучшения синтаксиса языка, а также функцию проверки доступности функций Swift для целевых ОС.

3 декабря 2015 года была выпущена бета версия Swift 3.0 с поддержкой операционных систем OS X, iOS и Linux

19 сентября 2017 года была выпущена версия Swift 4.0.

20 сентября 2019 Swift 5.1 — это следующий крупный релиз Swift, и его задача наконец обеспечить стабильность ABI.

Источники:

* progopedia.ru
* habr.com
* ru.wikipedia.org
* bourabai.ru
* cppstudio.com
* tproger.ru
* geekbrains.ru
* ru.bmstu.wiki
* studfile.net