**Аннотация.**

This article discusses the major developments in the history of programming languages. In the 1950s the first three modern programming languages whose descendants are still in widespread use today were designed: FORTRAN, LISP, COBOL. Another milestone in the late 1950s was the publication the "Algol 60 Report", which featured two key language innovations: arbitrarily nested block structure and lexical scoping. The period from the late 1960s to the late 1970s brought a major flowering of programming languages: Pascal,C,SQL and ML. The 1990s many "rapid application development" languages emerged, which usually came with an IDE, garbage collection, and were descendants of older languages. All were object-oriented. These included Object Pascal, Visual Basic, and C#.

**Вопросы:**

1. Which language innovations does the "Algol 60 Report" include?
2. Which paradigms of programming languages exists nowadays?
3. When and who developed the C programming language?
4. Is C# an object-oriented programming language?
5. Does ML programming language developed on the basis of LISP?
6. Did the LISP programming language developed by John McCarthy?

**Перевод:**

**История языков программирования**

В данной статье описаны основные события в истории языков программирования. Детальный график событий смотрите в разделе хронология языков программирования.

**История языков программирования**

Первые языки программирования предшествовали современному компьютеру. С самого начала языки были кодами.

В течение девяти месяцев в 1842-1843 годах Ада Лавлейс перевела мемуары итальянского математика Луиджи Менабреа о новейшей машине Чарльза Бэббиджа - аналитической машине. К статье она добавила примечания, в которых описана методика вычисления чисел Бернулли с помощью двигателя, признанного некоторыми историками первой в мире компьютерной программой. Но некоторые биографы спорят о степени ее первоначального вклада по сравнению с вкладом ее мужа.

Жаккардовый ткацкий станок использовал отверстия в перфокартах для представления движений рук швейного ткацкого станка, чтобы автоматически генерировать Декоративные узоры.

Герман Холлерит реализовал кодирование информации на перфокартах, когда заметил, что кондукторы поездов кодируют внешний вид держателей билетов, используя положение пробитых отверстий на билетах. Затем Холлерит приступил к кодированию данных переписи 1890 года на перфокартах.

Первые компьютерные коды были специализированы для приложений. В первые десятилетия двадцатого века численные расчеты основывались на десятичных числах. В конце концов стало ясно, что логику можно представить не только словами, но и числами. Например, Алонсо Черч смог выразить лямбда-исчисление формальным способом. Машина Тьюринга была абстракцией работы машины для разметки ленты, например, используемой в телефонных компаниях. Однако, в отличие от лямбда-исчисления, код Тьюринга не очень хорошо служит основой для языков более высокого уровня.; его основное применение заключается в строгом анализе алгоритмической сложности.

Как и многие "первые" в истории, первый современный язык программирования трудно определить. С самого начала язык определялся аппаратными ограничениями. Перфокарты допускали 80 столбцов, но некоторые столбцы приходилось использовать для сортировки числа на каждой карте. Fortran включал некоторые ключевые слова, которые были такими же, как английские слова, такие как "IF", "GOTO" (go to) и "CONTINUE". Использование магнитного барабана для памяти означало, что компьютерные программы также должны были чередоваться с вращениями барабана. Таким образом, программы были более зависимы от оборудования, чем сегодня.

Для некоторых людей ответ зависит от того, сколько мощности и удобочитаемости требуется для получения статуса "языка программирования". Жаккардовые станки и разностная машина Чарльза Бэббиджа имели простые, крайне ограниченные языки для описания действий, которые должны были выполнять эти машины. Можно даже рассматривать отверстия на свитке пианино плеера как ограниченный предметно-специфический язык программирования, хотя и не предназначенный для потребления человеком.

**1940-е годы**

В 1940-х годах были созданы первые современные компьютеры с электрическим приводом. Ограниченная скорость и объем памяти вынуждали программистов писать программы на ассемблере, настроенные вручную. Вскоре выяснилось, что программирование на ассемблере требует больших интеллектуальных усилий и подвержено ошибкам.

В 1948 году Конрадзузе опубликовал статью о своем языке программирования Plankalkül. Однако она не была реализована в его время, и его первоначальный вклад был изолирован от других разработок.

Некоторые важные языки, которые были разработаны в этот период, включают в себя:

- 1943 - Plankalkül (KonradZuse)

- 1943 - ENIAC coding system

- 1949 - C-10

**1950-е и 1960-е годы**

В 1950-х годах были разработаны первые три современных языка программирования, потомки которых до сих пор широко используются сегодня:

- FORTRAN (1955), "FORmulaTRANslator", изобретенный Джоном У. Бэкусом и др.;

- LISP, "LIStProcessor", изобретенный Джоном Маккарти и др.;

- COBOL, в COmmonBusiness ориентированный язык, созданный в короткий комитета диапазона, в значительной степени под влиянием Грейс Хоппер.

Еще одной важной вехой в конце 1950-х годов стала публикация комитетом американских и европейских ученых-компьютерщиков "нового языка для алгоритмов"; "доклад Алгола 60" ("ALGOrithmicLanguage"). Этот отчет объединил многие идеи, циркулировавшие в то время, и показал два ключевых языковых новшества:

- произвольно вложенная блочная структура: значимые фрагменты кода могут быть сгруппированы в блоки операторов без необходимости превращения в отдельные явно именованные процедуры.;

- лексическая область действия: блок может иметь свои собственные переменные, которые код вне блока не может видеть, не говоря уже о том, чтобы возиться с ними.Еще одно новшество, связанное с этим, заключалось в том, как был описан язык:

- для описания синтаксиса языка использовалась математически точная нотация-форма Бэкуса-Наура (BNF). Почти все последующие языки программирования использовали вариант BNF для описания контекстно-свободной части своего синтаксиса.

Алгол 60 особенно повлиял на дизайн более поздних языков, некоторые из которых вскоре стали более популярными. Большие системы Берроуза были разработаны для программирования в расширенном подмножестве Алгола.

Ключевые идеи Алгола были продолжены, производя Алгол 68:

- синтаксис и семантика стали еще более ортогональными, с анонимными подпрограммами, рекурсивной системой типизации с функциями более высокого порядка и т. д.;

- не только контекстно-свободная часть, но и весь синтаксис и семантика языка были определены формально, в терминах грамматики Ван Вейнгардена, формализма, разработанного специально для этой цели.Многие малоиспользуемые языковые функции Algol 68 (например, параллельные и параллельные блоки) и его сложная система синтаксических сокращений и автоматического принуждения типов сделали его непопулярным среди разработчиков и приобрели ему репутацию "трудного". Никлаус Вирт фактически вышел из проектного комитета, чтобы создать более простой язык Pascal.

Обзор:

- 1951 - Regional Assembly Language

- 1952 - Autocode

- 1954 - FORTRAN

- 1955 - FLOW-MATIC (forerunner to COBOL)

- 1957 - COMTRAN (forerunner to COBOL)

- 1958 - LISP

- 1958 - ALGOL 58

- 1959 - COBOL

- 1962 - APL

- 1962 - Simula

- 1964 - BASIC

- 1964 - PL/I"'

**1967-1978: установление фундаментальных парадигм**

Период с конца 1960-х до конца 1970-х годов принес значительный расцвет языков программирования. Большинство основных языковых парадигм, используемых в настоящее время, были изобретены в этот период:

- Simula, изобретенный в конце 1960-х годов Найгардом и Далем как надмножество Algol 60, был первым языком, предназначенным для поддержки объектно-ориентированного программирования.

- Smalltalk (середина 1970-х годов) обеспечил полную основу дизайна объектно-ориентированного языка.

- C, ранний язык системного программирования, был разработан Деннисом Ричи и Кеном Томпсоном в Bell Labs между 1969 и 1973 годами.

- Prolog, разработанный в 1972 году Кольмерауэром, Русселем и Ковальски, был первым языком логического программирования.

- ML построил полиморфную систему типов (изобретенную Робином Милнером в 1973 году) поверх Lisp, став пионером статически типизированных функциональных языков программирования.Каждый из этих языков породил целую семью потомков, и большинство современных языков насчитывают по крайней мере одного из них в своей родословной.

В 1960-х и 1970-х годах также велись значительные дебаты по поводу достоинств "структурированного программирования", которое по существу означало Программирование без использования GOTO. Эта дискуссия была тесно связана с языковым дизайном: некоторые языки не включали GOTO, что вынуждало программиста выполнять структурированное программирование. Хотя в то время споры бушевали жарко, почти все программисты теперь согласны с тем, что даже в языках, которые предоставляют GOTO, это плохой стиль, чтобы использовать его, за исключением редких случаев. В результате более поздние поколения языковых дизайнеров нашли дебаты о структурированном программировании утомительными и даже сбивающими с толку.

Некоторые важные языки, которые были разработаны в этот период, включают в себя:

- 1970 - Pascal

- 1970 - Forth

- 1972 - C

- 1972 - Smalltalk

- 1972 - Prolog

- 1973 - ML

- 1978 - SQL

**1980-е годы: консолидация, модули, производительность**

1980-е годы были годами относительной консолидации. C++ объединил объектно-ориентированное и системное программирование. Правительство Соединенных Штатов стандартизировало Ada, язык системного программирования, предназначенный для использования оборонными подрядчиками. В Японии и других странах огромные суммы были потрачены на изучение так называемых языков "пятого поколения", которые включали логические конструкции программирования. Сообщество функциональных языков перешло к стандартизации ML и Lisp. Вместо того чтобы изобретать новые парадигмы, все эти движения развивали идеи, изобретенные в предыдущем десятилетии.

Однако одной из важных новых тенденций в языковом дизайне стало усиление внимания к программированию для крупномасштабных систем за счет использования "модулей", или крупномасштабных организационных единиц кода. Modula, Ada и ML разработали известные модульные системы в 1980-х годах. Модульные системы часто были связаны с общими программными конструкциями-дженерики, по сути, были параметризованными модулями (см. Также параметрический полиморфизм).

Хотя основные новые парадигмы для языков программирования так и не появились, многие исследователи расширили идеи предыдущих языков и адаптировали их к новым контекстам. Например, языки систем Argus и Emerald адаптировали объектно-ориентированное программирование к распределенным системам.

1980 - е годы также принесли прогресс в реализации языка программирования. Движение RISC в компьютерной архитектуре постулировало, что аппаратное обеспечение должно быть разработано для компиляторов, а не для программистов-ассемблеров. Благодаря повышению скорости процессора, что позволило использовать все более агрессивные методы компиляции, движение RISC вызвало повышенный интерес к технологии компиляции для языков высокого уровня.

Языковые технологии продолжали развиваться в этом направлении и в 1990-е годы.

Некоторые важные языки, которые были разработаны в этот период, включают в себя:

- 1983 - Ada

- 1983 - C++

-1985 - Eiffel

- 1987 - Perl

- 1989 - FL (Backus)

**1990-е годы: эпоха Интернета**

В 1990-е годы не было фундаментальной новизны, но было много рекомбинации и созревания старых идей. Большой движущей философией была продуктивность программистов. Появилось много языков "быстрой разработки приложений", которые обычно шли с IDE, сборкой мусора и были потомками более старых языков. Все они были объектно-ориентированными. К ним относятся Object Pascal, Visual Basic и C#. Java был более консервативным языком, который также включал сборку мусора и получал большое внимание. Более радикальными и инновационными, чем языки RAD, были новые скриптовые языки. Они не были прямыми потомками других языков и отличались новыми синтаксисами и более либеральным включением признаков. Многие считают эти скриптовые языки более продуктивными, чем даже языки RAD, но часто из-за выбора, который делает небольшие программы более простыми, но большие программы более трудными для написания и обслуживания. Тем не менее, скриптовые языки стали самыми известными из тех, которые используются в связи с Интернетом.

Некоторые важные языки, которые были разработаны в этот период, включают в себя:

- 1990 - Haskell

- 1991 - Python

- 1991 - Java

- 1993 - Ruby

- 1995 - PHP

- 2000 - C#

**Текущие тренды**

Эволюция языка программирования продолжается, как в промышленности, так и в научных исследованиях. Некоторые современные направления:

- Механизмы добавления в язык проверки безопасности и надежности: расширенная статическая проверка, Управление информационными потоками, статическая потокобезопасность.

- Альтернативные механизмы модульности: миксины, делегаты, аспекты.

- Разработка компонентно-ориентированного программного обеспечения.

- Метапрограммирование, рефлексия или доступ к абстрактному синтаксическому дереву

- Уделение повышенного внимания распределению и мобильности.

- Интеграция с базами данных, включая XML и реляционные базы данных.

- Открытый исходный код как философия развития для языков, включая коллекцию компиляторов GNU и последние языки, такие как Python, Ruby и Squeak.

- Поддержка Unicode, так что исходный код (текст программы) не ограничивается теми символами, которые содержатся в наборе символов ASCII; позволяя, например, использовать нелатинские сценарии или расширенную пунктуацию.