указателями и структурами из указателей и т. п. Но для такого проекта не хватило средств. Окончательно сформированные принципы, положенные в основу языка Лисп: использование единого спискового представления для программ и данных; применение выражений для определения функций; скобочный синтаксис языка.ПШПлрвщрвсщфышрывщ

Аббревиатура LISP расшифровывается как LIST PROCCESSOR или по-русски: обработчик списков. Лисп является языком низкого уровня, его можно рассматривать как ассемблер, ориентированный на работу со списковыми структурами. Поэтому на протяжении всего существования языка было много попыток его усовершенствования за счет введения дополнительных базисных примитивов и управляющих структур. Но все эти изменения, как правило, не становились самостоятельными языками. В новых своих редакциях Лисп быстро усваивал все ценные изобретения своих конкурентов.

Из-за своего экспериментального акцента на символьной обработке, Лисп был сначала очень неэффективным для цифровых вычислений, по крайней мере, по сравнению с Fortran. Лисп преодолел свою старую репутацию как безнадежно неэффективный, Лисп теперь использован во многих приложениях, где эффективность не является центральным действием. Например, Лисп стал языком выбора для оболочек операционной системы и для расширения языков, для редактирования систем и автоматизированного проектирования. Лисп, - второй старший язык в широко распространенном использовании сегодня (только Fortran более стар), который непрерывно приспосабливается, чтобы охватывать наиболее современные идеи о программировании.

После создания в начале 70-х годов мощных Лисп-систем были попытки создания языков ИИ, отличных от Лиспа, но на той же основе. Дальнейшее развитие языка идет, с одной стороны, по пути его стандартизации (Стандарт-Лисп, Франц-Лисп, Коммон Лисп), а с другой - в направлении создания концептуально новых языков для представления и манипулирования знаниями в Лисп среде. В настоящее время Лисп реализован на всех классах ЭВМ, начиная с ПЭВМ и кончая высоко производительными вычислительными системами.

**Особенности диалектов языка Лисп.**

**Маклисп.**

Кроме символьной обработки Маклисп широко использовался в традиционных числовых вычислениях, применяемых, например, в обработке речи и изображений. Кроме исследователей ИИ и разработчиков алгебраической системы Максима на Маклисп оказали влияние и работы групп в МИТ по робототехнике, обработке речи и изображений. Исходя из требований, предъявляемых этими областями, в Маклисп были включены новые математические типы данных, такие как матричная и битовая обработка, а также широкий набор арифметических функций и средств. Быть может, важнейшая из них - возможность вычислений с неограниченной точностью, основывающаяся на созданных Кнутом (1969) алгоритмах.

Маклисп был также первой Лисп-системой для которой создан хороший транслятор. Транслятор генерирует машинную программу в форме списков. Машинный код в виде списка легко обрабатывать и результирующий код для числовых задач получался эффективнее, чем у лучших фортрановских трансляторов.

Однако большую часть своих свойств Маклисп приобрел под влиянием стоящих перед исследователями ИИ проблем и накопленного опыта. Так в язык попали макросы чтения и таблицы чтения, позволяющие легко изменять и расширять структуру языка. Таким же образом из требований к программам и окружению возникли управляющие структуры, механизмы обработки прерываний и ошибок, а также использование управляющих символов, создан и интегрирован в систему экранный редактор, появились управление и взаимодействие параллельных процессов.

Основное внимание разработчики Маклиспа сосредоточили на эффективности. Этому служат указания, уточняющие способы обработки аргументов функций, а также экранирование от вмешательства программиста внутренних механизмов системы. За счет этих мер скорость работы Маклиспа в 1.5-2.5 раза выше, чем Интерлисп.

Всего в Маклиспе используется около 400 функций. Самым большим недостатком системы является то, что ее никогда не документировали должным образом. Документация по этой системе разбросана по различным отчетам и руководствам. Маклисп был исследовательской системой и не предназначался для обучения и промышленного использования.

**муЛисп.**

Интерпретатор Мулисп-85, разработанный для ПЭВМ серии IBM PC - удачный вариант реализации диалекта языка, включающий сравнительно ограниченный набор базовых функций (около 260) и оказавшийся вследствие этого более простым для изучения.

По сравнению с Коммон Лиспом диалект муЛисп не имеет такого широкого спектра доступных типов данных. В нем обеспечивается работа только с двумя типами числовой информации: целыми числами с любым основанием и рациональными. В диалекте отсутствуют средства работы со структурами, массивами, потоками и другими типами данных, указанная реализация языка Лисп имеет одно существенное преимущество - возможность пополнения базового набора функций путем подключения подпрограмм, написанных на языке ассемблера, что позволило повысить гибкость использования интерпретатора и эффективность прикладного программного обеспечения, создаваемого на его основе. Возможность такого пополнения отсутствует в большинстве других Лисп-систем, являющихся в этом смысле замкнутыми программными продуктами.

Среди других, вероятно, менее существенных, особенностей системы можно указать на реализацию специального механизма, позволяющего не заботиться о присваивании начальных значений литеральным атомам, получающих изначальное значение, равное «печатному» имени самого атома. Еще одной особенностью диалекта является возможность использования новой синтаксической конструкции «встроенный COND», существенно сокращающей тексты описаний функций пользователя и применяемой при записи тел функций и лямбда-выражений.

Набор базовых функций муЛисп-интерпретатора включает ряд функций, обеспечивающих доступ практически ко всем функциям ОС ЭВМ через соответствующие прерывания. Наконец, указанная Лисп-система обеспечивается библиотеками Лисп-функций, дополняющими базовый набор функциями, имеющимися в диалектах Коммон Лисп и Интерлисп, что облегчает решение проблемы переносимости исходных текстов программных модулей, а также библиотеками, позволяющими выполнять манипулирование окнами на экране дисплея и обрабатывать управляющие воздействия через устройство типа «мышь». В комплект дополнительного программного обеспечения к интерпретатору входят интерактивный редактор текстов и простая обучающая система, написанные на диалекте языка муЛисп.

**Интерлисп.**

Интерлисп появился в 1972 году из ББН-Лиспа. К 1978 году, когда вышло описание Интерлиспа, язык и система уже достаточно стабилизировались. Интерлисп уже не был языком в том же смысле, что и Маклисп или другие Лисп - системы или обычные традиционные системы программирования. Он представлял собой интегрированную среду программирования, в которую вошло множество различных вспомогательных средств. Интерлисп стал классическим примером хорошо развитых программных средств и средств в системах разделения времени.

Этот диалект наряду с Коммон Лиспом один из наиболее распространенных, имеет достаточно развитый аппарат представления и манипулирования различными структурами данных, включая массивы. Среди общих особенностей данного варианта языка следует отметить использование для обозначения встроенных функций нетрадиционных имен, что порой затрудняет перенос готовых программных продуктов на другие диалекты и другие ЭВМ.

В 1974 году Xerox начала разработку для Интерлиспа персональной лисповской рабочей станции под названием Alto. В реализации Лиспа для Alto впервые применили спроектированную специально для языка Лисп микропрограммируемую систему команд и мини-ЭВМ, способную с более высокой производительностью, чем универсальные ЭВМ, интерпретировать лисповские программы. Из этой машины Alto впоследствии развились Лисп-машины серии 1100 фирмы Xerox.

На основе версии Интерлиспа, работавшей в системе разделения времени, была создана совместимая снизу вверх версия Лиспа Интерлисп-де, используемая на Лисп-машинах серии 1100. В ее пользовательский интерфейс входили многооконное взаимодействие, графика с высокой разрешающей способностью, средства выбора из меню и мышь, а также ориентированный на использование экрана инспектор структур данных. Идея разделения экрана на многие независимые окна родилась в XLG. Алан Кэй уже в конце 60-х годов предложил такую идею подхода к компьютерам будущего и интерфейсу между человеком и машиной. Работа XLG привела к созданию в 70-х годах к разработке языка программирования Smolltalk и объектного программирования.

При создании Интерлиспа работа велась весьма тщательно. Система хорошо документирована и более новые версии совместимы с более ранними. Так преимуществом системы стало непрерывно пополняющееся большое количество переносимого программного обеспечения. С другой стороны, ограничение системы старым, зафиксированным уже в конце 70-х годов диалектом сделало систему отчасти устаревшей и трудно расширяемой. В Интерлиспе среди прочего отсутствуют иерархические типы данных, объекты и замыкания. К тому же он базируется на динамическом связывании, тогда как новые версии Лиспа - статические. Однако из Интерлиспа берет начало новая версия - Коммон Лисп (1986). Для программирования на более высоком уровне в Интерлисп разработаны такие средства, в которых уже присутствовали объекты.

Интерлисп - столь замкнутая система, что доступна только ее оттранслированная версия в машинных кодах. В некоторых других системах, таких как, например Зеталисп, поддерживается версия Лиспа на исходном языке, которая доступна пользователю и может модифицироваться им. Развитие закрытых систем, похожих на Интерлисп, связано с ресурсами, имеющимся у создавших их лабораторий.

Интерлисп использует свыше 500 функций и большое количество системных имен и флажков, с помощью которых можно настроить и подогнать систему. Интерлисп реализован в системе разделения времени на многих больших ЭВМ.

В Интерлиспе основное внимание было уделено удобству системы для пользователя. Главный принцип разработчиков этого диалекта: все, что может иметь место в системе, должно естественно выражаться в терминах ее входного языка. Поэтому в Интерлисп программисту доступно все. Он может переопределять любые, в том числе и встроенные, функции; задавать и переопределять реакции на ошибки; работать непосредственно с уровня входного языка с внутренними структурами интерпретатора и т. д. При этом система поддерживает свою целостность и работоспособность.

**Франс Лисп.**

Маклисп стал основой для многих новых диалектов языка Лисп, первым из которых был Франс Лисп. Эта версия Лиспа названа в честь известного венгерского композитора. Главным мотивом разработки Франс Лисп было желание получить современную Лисп-систему для новых машин VAX, чтобы на них можно было использовать систему Максима и другое лисповское программное обеспечение. Франс Лисп в довольно большой степени напоминает Маклисп, на котором первоначально была реализована Максима. Однако в диалекте отсутствуют некоторые устаревшие особенности Маклиспа и содержатся более новые системные идеи, заимствованные в то время в MIT Лисп-машин для Зеталиспа.

Новый диалект был реализован в университете в Беркли на ЭВМ VAX 780/11 на языке Си под управлением системы UNIX. Франс Лисп довольно широко используется как под управлением UNIX, так и под управлением VAX/VMS и в настоящее время является наиболее часто используемой версией Лиспа для систем разделения времени. Кроме того, он широко используется и на 32-битовых микро-ЭВМ и рабочих станциях, работающих под управлением UNIX.

Благодаря своей хорошей переносимости Франс Лисп получил распространение во многих университетах и исследовательских учреждениях. Сопровождение системы также разошлось в различных исправлениях системных ошибок, реализациях наиболее эффективных алгоритмов, а также в расширениях языка.

**Зеталисп Лисп-машин.**

Зеталисп также опирается на Маклисп. Он создан в 70-е годы в MIT в рамках проекта Лисп-машины, финансированного оборонным агентством. С самого начала его целью было изготовление коммерческого продукта. В 1979 году в связи с проектом родились два предприятия изготавливающие Лисп-машины: Symbolic Inc. и Lisp Machine Inc. (LMI). После этого в 80-е годы работа по развитию Зета Лиспа продолжалась в них независимо друг от друга на коммерческой основе. В какой-то мере системы отличаются друг от друга, но в части Зета Лиспа машины почти совместимы.[2]

Зета Лисп содержит следующие развитые механизмы и черты:

* широкий выбор типов данных;
* возможность объектно-ориентированного программирования в системе Flavor ;
* современные управляющие структуры, динамические механизмы передачи управления сопрограммы и процессы;
* гибкий механизм ключевых слов в лямбда-списке и многозначные функции;
* ввод и вывод, основывающийся на потоках;
* пространства имен;
* уже готовые функции, в том числе для сортировки, работы с линейными управлениями и матричные вычисления.

Выбор используемых в среде Зеталиспа инструментов и языков программирования зависит от поставщика, причем предлагаемый набор средств все время расширяется. Среди других языков предлагаются Фортран, Паскаль, Ада и Пролог. Для этих языков в среде Зеталиспа существуют особенно развитые программные окружения, и разработанные в них программы можно выполнять вместе с программами на Лиспе.

Готовые инструменты и прикладные разработки в большом количестве имеются для работы с ЭС, с естественным языком и речью, с реляционными базами данных, машинной графики и машинного проектирования.

**Коммон Лисп.**

Этот диалект отличается наиболее широким представлением различных структур данных и включает около 800 встроенных функций. В этом диалекте обеспечиваются средства обработки основных классов числовой информации: целых, вещественных и комплексных. Символьные данные (литеры, литеральные атомы, строки) в Коммон Лиспе соответствуют основным возможностям других Лисп-систем. Дополнительно имеются средства обработки непечатных литер в символьных именах.

Одним из существенных преимуществ диалекта Коммон Лисп является наличие средств обработки массивов и структур, по своим возможностям не уступающих соответствующим средствам традиционных языков программирования (Фортран, Паскаль). Массивы в Коммон Лиспе могут иметь любое неотрицательное число измерений и индексируются последовательностью целых чисел. Тип компонентов массива может быть произвольным. Выделяется подкласс векторов - одномерных массивов, среди которых отдельно рассматриваются строки и битовые массивы.

Структуры Коммон Лиспа являются типом многокомпонентных записей, определяемых пользователем и имеющих именованные компоненты. Встроенное макроопределение DEFSTRUCT используется для определения структур новых типов. Для создания данных нового типа в виде структуры предусмотрены средства автоматической генерации набора функций, обеспечивающих средства манипулирования объектами этого класса.

Удобным средством контроля доступа к различным переменным и функциям Лисп-программ в Коммон-Лиспе являются пакеты. Пакет - множество имен объектов, определенных и доступных в нем. Внутри пакета имена объектов подразделяются на внутренние и внешние. Первые предназначены для использования только внутри данного пакета, а вторые - для обеспечения связи с другими пакетами. Лисп-интерпретатор представляет широкий спектр средств манипулирования с пакетами. Как правило, Лисп-система имеет в своем составе четыре стандартных пакета: lisp (содержащий примитивы системы), user (умалчиваемый пакет прикладных программ и данных пользователя), keyword (содержащий ключевые слова всех встроенных функций и функций, определяемых пользователем), system (зарезервированный для системных целей).

Значительной переработке и расширению в Коммон Лиспе подверглись средства ввода-вывода и передачи информации. Для эффективной организации различных обменов с внешней средой введена концепция потоков, позволяющих осуществлять одно- и (или) двухстороннюю передачу информации. Для потоков предусмотрен набор базовых функций. В диалекте различают символьные и двоичные потоки. В первом случае передача осуществляется по байтам, а во втором - целыми числами. Кроме стандартных потоков пользователь имеет возможность создавать и использовать собственные потоки.

В дополнение к указанным типам данных Коммон Лисп имеет ряд специфических классов объектов: хэш-таблицы, обеспечивающие эффективный способ доступа к данным по ключу; READ-таблицы, обеспечивающие управление обработкой информации поступающей из входного потока Лисп-системы, и некоторые другие. Такое множество имеющихся в диалекте различных типов данных, с одной стороны, развеивает существующее ошибочное представление о языке Лисп как о средстве обработки только символьной информации, а с другой - наличие мощных средств манипулирования типами данных существенно усложняет его.

Этот диалект оставлен открыт: принципиальным является то, что осталась возможность в будущем, когда подойдет время и будет достигнуто согласие, добавить новые средства и методы. Эта идея как раз соответствует духу Лиспа.

Коммон Лисп не является готовой программной системой в том же смысле, что и Интерлисп, поскольку вопросы среды в основном оставлены открытыми. В стандарте не определено, каким должен быть редактор или другие вспомогательные средства. Сказано лишь в самом общем виде, каким образом они вызываются. Для того чтобы обеспечить быстрое развитие, среда и инструментальные средства еще не затронуты стандартизацией, и поэтому есть возможность создавать различные среды для различных целей. Коммон Лисп не определяет также и интерфейс пользователя.

В Коммон Лисп на современном этапе не включены даже средства объектного программирования, хотя и определены необходимые для этого базовые механизмы (замыкание и др.). Таким образом, объекты можно реализовать на Лиспе. Но уже ведется работа по стандартизации средств и форм объектного программирования.

В Коммон Лиспе много внимания уделено практическим требованиям, и, вероятно, поэтому не все его черты эстетичны и чисты. Несомненно, что и другие Лисп-системы будут использоваться в дальнейшем, и их также необходимо развивать.

Коммон Лисп предназначен не только для работы со списками или для символьной обработки, он является универсальным языком программирования, включающим в себя особенно хорошие средства для численных вычислений, управления данными и связи. На Коммон Лиспе можно с одинаковым успехом писать программы в традиционных операторном, процедурном, фразовом стиле, а также и в свойственных Лиспу стилях. В языке содержатся предпосылки для использования различных способов и стилей программирования, таких как операторное, функциональное, макропрограммирование, программирование, управляемое данными, и продукционное программирование, а также средства, необходимые для логического и объектного программирования и реализации других средств более высокого уровня.

Можно смело сказать, что Коммон Лисп содержит почти все, что на сегодняшний день могут дать другие известные языки программирования, и, кроме того, он предусматривает средства для расширения языка.

Современные диалекты языка Лисп можно рассматривать как мощные интерактивные системы программирования. Это объясняется двумя причинами. Во-первых, сам язык Лисп претерпевает серьезные изменения - развиваются средства языка, предназначенные для обработки нетрадиционных для Лиспа типов данных: массивов, векторов, матриц; появляются некоторые средства управления памятью (пакеты), отсутствующие в Лиспе. Серьезные изменения претерпевают и управляющие структуры. Появились несвойственные природе языка Лисп функции, заимствованные из Фортрана, Алгола, Паскаля, Си: Do, Loop, Goto , Case и прочие, позволяющие пользователю, незнакомому с принципами функциональных языков, легко переходить на Лисп. Качество программ снижается, зато возрастает популярность языка. Во-вторых, если на первом этапе развития Лисп-системам была присуща небольшая скорость обработки данных и серьезные ограничения на емкость используемой оперативной памяти, то современные Лисп-системы уже могут соперничать по этим параметрам с такими языками, как Си, Паскаль и др. Использование Лисп-машин вообще практически снимает ограничения памяти и быстродействия.