

*В.А. Макарчук, 1 курс**Научный руководитель – Л.П. Володько, к.э.н., доцент**Полесский государственный университет*

Лисп (LISP, от англ. LISt Processing language — «язык обработки списков») - семейство языков программирования, программы и данные в которых представляются системами линейных списков символов. Создатель Лиспа Джон Маккарти занимался исследованиями в области искусственного интеллекта (далее ИИ) и созданный им язык по сию пору является одним из основных средств моделирования различных аспектов ИИ [1].

В настоящее время Лисп – старейший язык после Фортрана из широко используемых языков программирования. Первый интерпретатор Лиспа появился в 1958 году. До 80-х годов Лисп был закрыт в лабораториях и исследовательских центрах. Настоящий расцвет языка, по-видимому, произойдёт в ближайшем будущем, так как растёт интерес к идее ИИ и созданию сверхбыстрых компьютеров.

Лисп – важнейший язык, используемый в символической обработке и в исследованиях по ИИ. Работы, по исследованию ИИ, начатые в США и MIT (англ. Massachusetts Institute of Technology - Массачусетский технологический институт) уже в середине 50-х годов, проводились преимущественно на языке Лисп. Большая часть программных систем ИИ написаны на Лиспе. На данный момент Лисп становится более популярным, существуют десятки диалектов Лиспа, несколько из них входят в 10 самых перспективных языков программирования (Clojure 1.6, Eginson 3.3.3).

Основные методы программирования на Лиспе – функциональное и рекурсивное программирование. Вычисление начинается с вызова некоторой функции, которая в свою очередь вызывает функции, входящие в её определение и т.д. в соответствии с иерархией определений и структурой условных предложений. Функции часто либо прямо, либо опосредованно вызывают сами себя. Каждый вызов возвращает некоторое значение в вызвавшую его функцию, вычисление которой после этого продолжается; этот процесс повторяется до тех пор, пока запустившая вычисления функция не вернёт конечный результат пользователю.

Многие практические ситуации предполагают рекурсивное или само повторяющееся поведение, возвращающееся к самому себе. Рекурсия – вызов функцией, самой себя.

В Лиспе формы представления программы и обрабатываемых ею данных одинаковы. И то и другое представляется списочной структурой, имеющей одинаковую форму. Это открывает совершенно новые возможности программирования [2, с. 49].

Пользователь не должен заботиться о распределении памяти. Система резервирует и освобождает память автоматически в соответствии с потребностью. Когда память кончается, запускается специальный мусорщик (garbage collector). Он собирает неиспользуемые символы и списки, включает их в работу путём вторичного использования. Файлы используются в основном для хранения программ и данных в промежутке между сеансами [2, с. 51].

В Лиспе имена символов, переменных, списков, функций и других объектов не закреплены предварительно за какими-нибудь типами данных. Типы в общем не связаны с именами объектов данных, а сопровождают сами объекты. Таким образом, переменные могут в различные моменты времени представлять раз-

личные объекты. В этом смысле Лисп является языком без закрепления типов.

Динамическая, осуществляемая лишь в процессе исполнения, проверка типа и позднее связывание допускают разностороннее использование символов и гибкую модификацию программ. Функции можно определять практически независимо от типов данных, к которым они применяются. Однако указанная незакреплённость типов не означает, что в Лиспе вовсе нет данных различных типов. Платой за динамические типы являются действия по проверке типа на этапе исполнения. В Лисп-машинах проверка типа эффективно осуществляется на уровне аппаратуры.

Производство Лисп-машин началось в США в середине 80-х годов. Целью проектирования Лисп-машин была разработка их в виде персональных ЭВМ. По производительности оборудования Лисп-машины очень эффективны, кроме того, они имеют большой объём основной памяти. Их аппаратура спроектирована специально для вычислений на Лиспе. С точки зрения эффективности одной из наиболее важных особенностей является проверка типов на уровне аппаратуры, используемая в системах, происходящих из MIT [3, с. 291].

Лисп пользуется не столь широким спектром применения, как другие современные языки программирования, о нём ходит миф о его «странности», из-за огромного количества скобок и многого другого. Лисп позволяет взглянуть на программирование на более глубоком уровне (более глубокое в плане использования функциональной парадигмы, а не низкоуровневым управлением памятью программы). Но всё же, на Лисп написан Emacs (Емакс, Емакс, также Ймакс — семейство многофункциональных расширяемых текстовых редакторов) и Viaweb (Программное обеспечение Viaweb, написанное на Common Lisp, позволяло пользователям создавать свои собственные интернет-магазины. Летом 1998 года Viaweb был продан Yahoo за 455 000 акций Yahoo, оценивавшихся в \$49,6 миллионов. Yahoo назвала этот продукт Yahoo Store), а так же почти все профессиональные системы компьютерной алгебры, такие как Maxima или, к примеру, Reduce.

Мощность Лиспу обеспечивают инфраструктурные технологии, формирующие подход его использования, и обеспечивающие мультипарадигмальность. Это достигается за счёт макрос-системы, которая позволяет расширять возможности языка самому. Лисп в этом смысле называют программируемым языком программирования.

Лисп широко применяется в исследовательских работах по программированию сложных методов обработки знаний. Большая часть значимых программ искусственного интеллекта запрограммирована на Лиспе или на основанном на нем языке более высокого уровня. По результатам некоторых исследований можно, скажем, в среде программирования Лисп-машины достичь повышения производительности программирования в несколько десятков раз по сравнению, например, с программированием на C++/Java [4].

В настоящее время Лисп широко не используется и главной причиной является тот факт, что процесс программирования на производстве слишком коммерциализирован. Это значит во-первых, что код, написанный разработчиком, должен быть максимально прост и легок в поддержке, т.е. если он уйдёт из компании, проекты, разрабатываемые им, без проблем должны быть “подхвачены” другими разработчиками. Во-вторых, начальнику не нужна ни философия, ни красота кода, ни оригинальные подходы к решению тех или иных задач. Главным тут является функциональность и время, потраченное на реализацию этой функциональности.

В заключение хотелось бы отметить, что в самых элитных ВУЗах планеты не перестают преподавать Лисп. Этот язык продолжает жить на компьютерах тех, кто знает, что искусство программирования, как и музыка или живопись, так же требует таланта и прозрения.

#### **Список использованных источников**

1. Материал из Википедии-свободной энциклопедии [Электрон. ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%81%D0%BF>
2. Хювёнен Э., Сеппянен Й. – Мир Лиспа. В 2-х т. Т.1: Введение в язык Лисп и функциональное программирование. Пер. с финск. – М.: Мир, 1990. – 447 с.
3. Хювёнен Э., Сеппянен Й. – Мир Лиспа. В 2-х т. Т.2: Методы и системы программирования. Пер. с финск. – М.: Мир, 1990. – 319 с.
4. Международный интернет-блог о Лисп [Электрон. ресурс]. – 03.02.2012. – Режим доступа: <https://thefreecountry.wordpress.com/2012/02/03/why-lisp/>