



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ
КАФЕДРА

Инженерный бизнес и менеджмент
Промышленная логистика

ДОКЛАД
по дисциплине
«Парадигмы и конструкции языков
программирования»

НА ТЕМУ:

«Lips – первый древнейший»

Студент

ИБМЗ-34Б
(группа)

(подпись, дата)

А.М.Гайдуков
(И.О. Фамилия)

Преподаватель

(подпись, дата)

(И.О. Фамилия)

Москва, 2024 г.

Введение

Лисп, или LISP (от англ. LISt Processing language — «язык обработки списков», современное написание: Lisp) — семейство языков программирования, программы и данные в которых представляются в виде списков.

Существует альтернативная расшифровка названия LISP: Lots of Irritating Superfluous Parentheses («Много раздражающих лишних скобок») — намёк на особенности синтаксиса языка.

Создан был этот язык в конце 1950-х годов математиком Джоном Маккарти. Принято считать, что именно он положил начало понятию «искусственного интеллекта».

Во время работы над одним из своих исследовательских проектов ему пришлось разработать совершенно новый язык программирования, содержащий ряд концепций, до этого не применявшихся.

Основные особенности языка

1. **Условные конструкции** Концепция if/then/else, знакомая многим программистам, впервые была реализована именно в Lisp. Впоследствии этот подход был заимствован другими языками программирования.
2. **Функции как первоклассные объекты** В Lisp функции рассматриваются как объекты первого класса, находящиеся на одном уровне с числами и строками. Их можно передавать в другие функции или возвращать в качестве результата.
3. **Рекурсия** Хотя рекурсия известна с древних времён как математическое понятие, её программная реализация впервые появилась в Lisp. Рекурсия заменяет традиционные циклы в языке.
4. **Переосмысление переменных** Все переменные в Lisp представляют собой указатели, что отличает их от традиционного подхода в других языках.
5. **Сборка мусора** Lisp стал первым языком, в котором был реализован механизм автоматической очистки памяти, удаляющий неиспользуемые объекты. Это нововведение позже стало стандартом для многих современных языков.
6. **Программа как выражения** Программы в Lisp представляют собой наборы выражений, способных возвращать значения. Это позволяет использовать гибкие конструкции, например:

```
(if foo (= x 1) (= x 2))
```

или

```
(= x (if foo 1 2))
```

7. **Символы как отдельный тип данных** Lisp поддерживает символы, которые отличаются от строк. Это позволяет проверять равенство символов через сравнение указателей, что делает операции более эффективными.
8. **Код как данные** Lisp позволяет представлять код в виде деревьев символов, что делает язык уникальным в плане метапрограммирования.

9. **Единое пространство языка** В Lisp отсутствует строгая граница между стадиями чтения, компиляции и выполнения. Программы можно компилировать, читать или выполнять в любой момент времени, что делает язык чрезвычайно гибким.
10. **Макросы** Одной из уникальных черт Lisp являются макросы, которые позволяют программистам расширять синтаксис языка. Макросы представляют собой программы, пишущие другие программы. Пол Грэм называет их "инструментами, которые создают инструменты".

Пример минимальной программы

Минимальная программа в Lisp удивительно проста по сравнению с аналогами в других языках.

- **Пример на C:**

- ```
int main()
{
 return 0;
}
```

- **Пример на Java:**

- ```
public class Minimal {
    public static void main(String[] args) {}
}
```

- **Пример на Lisp:**

`t`

В Lisp достаточно одного символа `t`, обозначающего истину. Эта программа возвращает значение `t`, не выполняя дополнительных операций.

Основные преимущества

1. **Гибкость определения программных элементов** Lisp предоставляет возможность создавать элементы программы (например, функции или классы) только тогда, когда это действительно необходимо, избегая избыточных определений.
2. **Любая сущность может быть выполнена** В Lisp любая сущность, будь то выражение, функция или значение, может быть запущена, и она всегда возвращает осмысленный результат. Это означает, что в языке отсутствует разделение между выражениями (expression) и инструкциями (statement). Таким образом, нет необходимости использовать символы завершения инструкций, как `;` в C/C++/Java, или специальные разделители в Pascal.

Макросы

Макросы являются одной из самых мощных особенностей Lisp. Они позволяют обрабатывать формы и объекты, генерируя код, который затем компилируется и выполняется. Этот процесс, называемый развёрткой макросов (macroexpansion), происходит до выполнения программы. Благодаря макросам, Lisp может имитировать синтаксис и возможности других языков программирования, что делает его невероятно гибким инструментом.

Популярность среди хакеров

Эта уникальная гибкость сделала Lisp популярным среди хакеров, особенно в прошлом. Программисты, которые осваивали Lisp, часто отмечали, что он не только предоставляет новые подходы к решению задач, но и помогает переосмыслить само программирование и информатику. Работа с Lisp расширяет горизонты понимания языка и архитектуры программного обеспечения.

Основы структуры

Lisp основан на списках, где каждое выражение является списком элементов. Эти элементы могут быть как конечными атомами (например, числа или символы), так и другими списками. Такой подход делает Lisp удобным для обработки данных и построения деревьев выражений.

Лисп-машины

Lisp также стал базовым языком для специализированных лисп-машин, производившихся в 1980-х годах, например, компанией Symbolics. Лисп-машина — это вычислительная система, оптимизированная для эффективного выполнения программ на Lisp. Эти устройства сыграли важную роль в развитии технологий, таких как:

- сборка мусора;
- лазерная печать;
- многооконные графические интерфейсы;
- растровая графика высокого разрешения;
- рендеринг;
- инновации в сетевых технологиях.

Хотя лисп-машины не получили широкого распространения, многие их разработки легли в основу современных инженерных рабочих станций.

Диалекты и стандартизация

К 1980-м годам существовало множество диалектов Lisp, таких как MacLisp, Scheme, Zetalisp и Common Lisp. Это многообразие затрудняло промышленное использование языка. Для решения проблемы Министерство обороны США поддержало создание стандарта Common Lisp, что позволило объединить основные возможности различных диалектов в один язык. Этот процесс завершился в 1984 году с публикацией руководства "Common Lisp: the Language" под редакцией Гая Стила.

Сферы применения

Lisp используется в различных областях, включая науку, промышленность, образование и медицину. Например, он применялся для:

- декодирования генома человека;
- проектирования самолётов;
- разработки интернет-серверов и приложений;
- научных расчётов и создания игр.

Некоторые специализированные диалекты Lisp включают:

- **AutoLISP** для автоматизации работы в AutoCAD;

- **Emacs Lisp** для расширения функционала текстового редактора Emacs;
- **Nyquist** для работы с аудиофайлами в Audacity;
- **SKILL** для проектирования электронных схем в Cadence Design Systems.

Заключение

Lisp — это не просто язык программирования, а мощный инструмент, который значительно опережает своё время. Его способность адаптироваться к различным задачам, поддерживать экспериментальные разработки и раскрывать новые подходы в информатике делает его актуальным даже спустя десятилетия. Хотя Lisp может показаться сложным для новичков, его изучение предоставляет уникальные возможности для профессионального роста и переосмысления основ программирования. Его наследие и вклад в развитие компьютерных наук остаются неоценимыми.