Кафедра информатики, факультет математики и информатики
Таврическая академия
Крымский федеральный университет имени В.И.Вернадского

Специализированные языки программирования

Айдер Анафиев

8 апреля 2015 г.

1 Язык Лисп

1.1 Основы языка Лисп

1. Запишите последовательности вызовов `CAR` и `CDR`, выделяющие из приведенных ниже списков символ `цель`. Упростите эти вызовы с помощью комбинации селекторов:
   - `(1 2 цель 3 4)`
   - `((1) (2 цель) (3 (4)))`
   - `((1 (2 (3 4 цель))))`
2. Определите функцию, возвращающую последни q элемент списка.
3. Определите функцию, заменяющую в исходном списке все вхождения заданного значения другим.
4. Определите функцию, порождающую по заданному натуральному числу `N` список, состоящий из натуральных чисел от `1` до `N`.
5. Определите функцию, которая увеличивает элементы исходного списка на единицу.

6. Определите функцию, переводящую список чисел в список соответствующих им названий.

7. Определите функцию, удаляющую из исходного списка элементы с четными номерами.

8. Определите функцию, которая разделит исходный список из целых чисел на два списка: список положительных чисел и список отрицательных чисел.

9. Определите функцию, разделяющую исходный список на два подсписка. В первый из них должны попасть элементы с нечетными номерами, во второй — элементы с четными номерами.

10. Определите функцию, осуществляющую удаление указанного количества последних элементов исходного списка.

11. Определите функцию, осуществляющую разделение исходного списка на два подсписка. В первый из них должно попасть указанное количество элементов с начала списка, во второй — оставшиеся элементы.

12. Определите функцию, заменяющую в исходном списке два подряд идущих одинаковых элемента одним.

13. Определите функцию, удаляющую в исходном списке все повторные вхождения элементов.

14. Определите функцию, осуществляющую перестановку двух элементов списка с заданными номерами.

15. Определите функцию, вычисляющую скалярное произведение векторов, заданных списками целых чисел.

16. Определите функцию, добавляющую элементы одного списка во второй список, начиная с заданной позиции.

17. Создайте предикат, порождающий всевозможные перестановки исходного множества.

18. Определите предикат, проверяющий, является ли аргумент одноуровневым списком.

19. Определите функцию `(ЛУКОВИЦА n)`, строящую `N`-уровневый вложенный список, элементом которого на самом глубоком уровне является `N`.

20. Определите функцию `ПЕРВЫЙ-АТОМ`, результатом которой будет первый атом списка. Пример:
```
> (ПЕРВЫЙ-АТОМ '(((a b)) c d))
A
```

21. Определите функцию, удаляющую из списка первое вхождение данного элемента на верхнем уровне.

22. Определите функцию, которая обращает список `(a b c)` и разбивает его на уровни `(((c) b) a)`.

23. Определите функции, преобразующие список `(a b c)` к виду `(a (b (c)))` и наоборот.

24. Определите функции, осуществляющие преобразования между видами `(a b c)` и `(((a) b) c)`.

25. Определите функцию, удаляющую из списка каждый четный элемент.

26. Определите функцию, разбивающую список `(a b c d...)` на пары `((a b) (c d)...)`.

27. Определите функцию, которая, чередуя элементы списков `(a b...)` и `(1 2...)`, образует новый список `(a 1 b 2 ...)`.

28. Определите функцию, вычисляющую, сколько всего атомов в списке (списочной структуре).

29. Определите функцию, вычисляющую глубину списка (самой глубокой ветви).

30. Запрограммируйте интерпретатор `ВЫЧИСЛИ`, который преобразует инфиксную запись операций в префиксную и возвращает значение выражения. Пример:
```
> (ВЫЧИСЛИ '((-2 + 4) * 3))
6
```

31. Определите функцию `(ПЕРВЫЙ-СОВПАДАЮЩИЙ x y)`, которая возвращает первый элемент, входящий в оба списка `x` и `y`, в противном случае `NIL`.

32. Определите предикат `МНОЖЕСТВО-Р`, который проверяет, является ли список множеством, т.е. входит ли каждый элемент в список лишь один раз.

33. Определите функцию `МНОЖЕСТВО`, преобразующую список в множество.

34. Определите предикат `РАВЕНСТВО-МНОЖЕСТВ`, проверяющий совпадение двух множеств (независимо от порядка следования элементов). Подсказка: напишите функцию `УДАЛИТЬ`, удаляющую данный элемент из множества.

35. Определите функцию `ПОДМНОЖЕСТВО`, которая проверяет, является ли одно множество подмножеством другого. Определите также `СОБСТВЕННОЕ-ПОДМНОЖЕСТВО`.

36. Определите предикат `НЕПЕРЕСЕКАЮЩИЕСЯ`, проверяющий, что два множества не пересекаются, т.е. у них нет общих элементов.

37. Определите функцию `ПЕРЕСЕЧЕНИЕ`, формирующую пересечение двух множеств, т.е. множество из их общих элементов.

38. Определите функцию `ОБЪЕДИНЕНИЕ`, формирующую объединение двух множеств.

39. Определите функцию `СИММЕТРИЧЕСКАЯ-РАЗНОСТЬ`, формирующую множество из элементов не входящих в оба множества.

40. Определите функцию `РАЗНОСТЬ`, формирующую разность двух множеств, т.е. удаляющую из первого множества все общие со вторым множеством элементы.

41. Реализовать генератор деревьев, чтобы выдаваемые им деревья имели количество вершин, точно соответствующее числу, указанному в его первом аргументе.

42. Определите функцию, находящую максимальное из значений, находящихся в вершинах дерева.

43. Определите функцию, подсчитывающую количество всех вершин данного дерева заданной высоты.

44. Определите функцию, проверяющую, является ли одно дерево поддеревом второго.

45. Предположим, что у имени города есть свойства `x` и `y`, которые содержат координаты места нахождения города относительно некоторого начала координат. Напишите функцию `(РАССТОЯНИЕ a b)`, вычисляющую расстояние между городами `a` и `b`.

46. Предположим, что отец и мать некоторого лица, хранятся как значения соответствующих свойств у символа, обозначающего это лицо. Напишите функцию `(РОДИТЕЛИ x)`, которая возвращает в качестве значения родителей, и предикат `(СЕСТРЫ-БРАТЬЯ x1 x2)`, который истинен в случае, если `x1` и `x2` — сестры или братья, родные или с одним общим родителем.

47. Определите функцию `УДАЛИТЬ-ВСЕ-СВОЙСТВА`, которая удаляет все свойства символа.

48. Функция `GET` возвращает в качестве результата `NIL` в том случае, если у символа нет данного свойства, либо если значением этого свойства является `NIL`. Следовательно, функцией `GET` нельзя проверить, есть ли некоторое свойство в списке свойств. Напишите предикат `(ИМЕЕТ-СВОЙСТВО символ свойство)`, который проверяет, обладает ли символ данным свойством.

1.2 Функции высших порядков

1. Определите `FUNCALL` через функционал `APPLY`.

2. Определите функицонал `(MAPLIST fn список)` для одного списочного аргумента.

3. Определите функционал `(APL-APPLY f x)`, который применяет каждую функцию `fi` списка

   `(f1 f2 ... fn)`

   к соответствующему элементу списка

   `x = (x1 x2 ... xn)`

   и возвращает список, сформированный из результатов.

4. Определите функциональный предикат `(КАЖДЫЙ пред список)`, который истинен в том и только в том случае, когда, являющейся функциональным аргументом предикат `пред` истинен для всех элементов списка `список`.

5. Определите функциональный предикат `(НЕКОТОРЫЙ пред список)`, который истинен, когда, являющейся функциональным аргументом предикат `пред` истинен хотя бы для одного элемента списка `список`.

6. Определите фильтр `(УДАЛйЬ-ЕСЛИ пред список)`, удаляющий из списка `список` все элементы, которые обладают свойством, наличие которого проверяет предикат `пред`.

7. Определите фильтр `(УДАЛйЬ-ЕСЛИ-НЕ пред список)`, удаляющий из списка `список` все элементы, которые не обладают свойством, наличие которого проверяет предикат `пред`.

8. Напишите генератор натуральных чисел: `0, 1, 2, 3, 4, 5, ...`

9. Напишите генератор порождения чисел Фибоначчи: `0, 1, 1, 2, 3, 5, ...`

10. Напишите генератор, порождающий последовательность `(A), (B A), (A B A), (B A B A), ...`

11. Определите фукнционал `МНОГОФУН`, который использует функции, являющиеся аргументами, по следующей схеме:
    `(МНОГОФУН '(f g ... h) x)` ⇔ `(LIST (f x) (g x) ... (h x))`.

12. Определите функцию, которая возвращает в качестве значения свой вызов.

13. Определите функцию, которая возвращает в качестве значения свое определение (лямбда-выражение).

14. Определите функцию, которая возвращает в качестве значения форму своего определения (`DEFUN`).

1.3 Макросы

1. Определите макрос, который возвращает свой вызов.
2. Определите макрос `(POP стек)`, который читает из стека верхний элемент и меняет значение переменной стека.
3. Определите лисповскую форму `(IF условие p q)` в виде макроса.
4. Определите в виде макроса форму `(FIF тест отр нуль полож)`.
5. Определите в виде макроса форму `(REPEAT e UNTIL p)` паскалевского типа.
6. Разработать “собственный” язык программирования.