

ВВЕДЕНИЕ

Целью работы является приобретение навыков использования списков и стандартных функций Lisp.

Задачи работы: изучить способ использования списков для фиксации информации, внутреннее представление одноуровневых и структурированных списков, методы их обработки с использованием базовых функций Lisp.

1 Теоретические сведения

Базис в Lisp образуют атомы, структуры, базовые функции, базовые функционалы.

Классификация **функций**:

- **классические функции** - математические функции, возвращающие один результат;
- **формы** - функции, принимающие произвольное количество аргументов, которые могут по-разному обрабатываться;
- **функционалы** - функции, принимающие на вход описание другой функции.

Так же существует классификация **базовых функций**:

- **конструкторы** - cons, list и т.д.;
- **селекторы** - функции доступа (например, car, cdr);
- **функции-предикаты** - atom, null и т.д.;
- **функции сравнения** - eq, eql, = и т.д.

2 Практическая часть

2.1 Задание 1

В таблице 2.1 приведены выражения и результат их вычисления.

Таблица 2.1 — Результаты вычисления выражений задания №1

Выражение	Результат
(caadr '((blue cube) (red pyramid)))	red
(cdar '((abc) (def) (ghi)))	nil
(cadr '((abc) (def) (ghi)))	(def)
(caddr '((abc) (def) (ghi)))	(ghi)

2.2 Задание 2

Таблица 2.2 — Результаты вычисления выражений задания №2

№	Выражение	Результат
1	(list 'Fred 'and 'Wilma)	(FRED AND WILMA)
2	(list 'Fred '(and Wilma))	(FRED (AND WILMA))
3	(cons nil nil)	(NIL)
4	(cons t nil)	(T)
5	(cons nil t)	(NIL .T)
6	(list nil)	(NIL)
7	(cons '(t) nil)	((T))
8	(list '(one two) '(free temp))	((ONE TWO) (FREE TEMP))
9	(cons 'Fred '(and Wilma))	(FRED AND WILMA)
10	(cons 'Fred '(Wilma))	(FRED WILMA)
11	(list nil nil)	(NIL NIL)
12	(list t nil)	(T NIL)
13	(list nil t)	(NIL T)
14	(cons t (list nil))	(T NIL)
15	(list (t) nil)	((T) NIL)
16	(cons '(one two) '(free temp))	((ONE TWO) FREE TEMP)

В таблице 2.2 приведены выражения и результат их вычисления.

2.3 Задание 3

В листинге 2.1 приведён текст трёх функций, при создании которых использовалась функция `list`, а в комментариях под каждой функцией - значение, которое она возвращает.

Листинг 2.1 — Примеры функций с использованием функции `list`

```
1 (defun f1 (ar1 ar2 ar3 ar4)
2   (list (list ar1 ar2) (list ar3 ar4)))
3 ; ((ar1 ar2) (ar3 ar4))
4
5 (defun f2 (ar1 ar2)
6   (list (list ar1) (list ar2)))
7 ; ((ar1) (ar2))
8
9 (defun f3 (ar1)
10  (list (list (list ar1))))
11 ; (((ar1)))
```

В листинге 2.2 приведён текст функций аналогичных функциям из листинга 2.3, но для их создания использовалась функция `cons`.

Листинг 2.2 — Примеры функций с использованием функции `list`

```
1 (defun f1 (ar1 ar2 ar3 ar4)
2   (cons (cons ar1 (cons ar2 nil)) (cons (cons ar3 (cons ar4 nil)) nil)))
3 ; ((ar1 ar2) (ar3 ar4))
4
5 (defun f2 (ar1 ar2)
6   (cons (cons ar1 nil) (cons (cons ar2 nil) nil)))
7 ; ((ar1) (ar2))
8
9 (defun f3 (ar1)
10  (cons (cons (cons ar1 nil) nil) nil))
11 ; (((ar1)))
```

На рисунках 2.1-2.3 изображены списочные ячейки результатов рассмотренных выше функций.

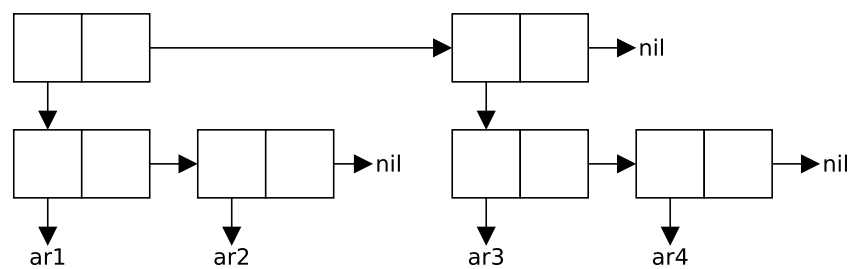


Рисунок 2.1 — Список ((ar1 ar2) (ar3 ar4))

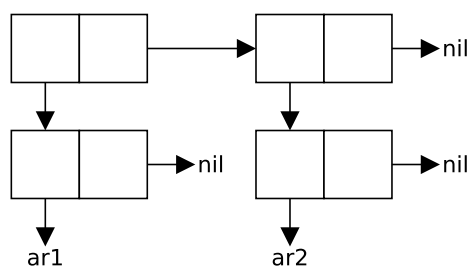


Рисунок 2.2 — Список ((ar1) (ar2))

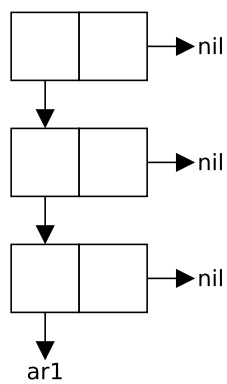


Рисунок 2.3 — Список (((ar1)))