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.

Функция в Лиспе есть однозначное отображение множества исходных данных на множество её значений. У функции может быть произвольно много аргументов, от нуля до любого конечного числа, но обязательно должно быть хотя бы одно значение.

Классификация функций:

— **Базовые функции** – принимают фиксированное количество аргументов

— **Формы** – принимают не фиксированное количество аргументов или обрабатывают аргументы по разному

— **Функционалы (высших порядков)** – используют другие функции в качестве аргументов или вырабатывают в качестве результатов.

CAR и **CDR** являются базовыми функциями доступа к данным. CAR принимает точечную пару или пустой список в качестве аргумента и возвращает первый элемент или nil, соответственно. CDR принимает точечную пару или пустой список и возвращает список состоящий из всех элементов, кроме первого. Если в списке меньше двух элементов, то возвращается Nil.

LIST и **CONS** являются функциями создания списков (cons – базовая, list – нет). Функция cons создает списочную ячейку и устанавливает два указателя на аргументы. Функция list принимает переменное число аргументов и возвращает список, элементы которого – переданные в функцию аргументы.

2 Практическая часть

2.1 Задание №3

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 №4

Таблица 2.2 — Результаты вычисления выражений задания №2

Выражение	Результат
(list 'Fred 'and 'Wilma)	(FRED AND WILMA)
(list 'Fred '(and Wilma))	(FRED (AND WILMA))
(cons nil nil)	(NIL)
(cons t nil)	(T)
(cons nil t)	(NIL .T)
(list nil)	(NIL)
(cons '(t) nil)	((T))
(list '(one two) '(free temp))	((ONE TWO) (FREE TEMP))
(cons 'Fred '(and Wilma))	(FRED AND WILMA)
(cons 'Fred '(Wilma))	(FRED WILMA)
(list nil nil)	(NIL NIL)
(list t nil)	(T NIL)
(list nil t)	(NIL T)
(cons t (list nil))	(T NIL)
(list (t) nil)	((T) NIL)
(cons '(one two) '(free temp))	((ONE TWO) FREE TEMP)

2.3 Задание №5

В листинге 2.1 приведён текст первой функции.

Листинг 2.1 — Примеры первой функции

```
1 (defun f1 (ar1 ar2 ar3 ar4)
2   (list (list ar1 ar2) (list ar3 ar4))
3 )
4
5 (defun f1 (a1 a2 a3 a4)
6     (cons (cons a1 (cons a2 Nil)) (cons (cons a3 (cons a4 Nil)) Nil))
7 )
8
9 ; ((ar1 ar2) (ar3 ar4))
```

В листинге 2.2 приведён текст второй функции.

Листинг 2.2 — Примеры второй функци

```
1 (defun f2 (ar1 ar2)
2   (list (list ar1) (list ar2))
3 )
4
5 (defun f2 (a1 a2)
6     (cons (cons a1 Nil) (cons (cons a2 Nil) Nil))
7 )
8
9 ; ((ar1) (ar2))
```

Листинг 2.3 — Примеры второй функци

```
1 (defun f3 (a)
2     (list (list (list a)))
3 )
4
5 (defun f3 (a)
6     (cons (cons (cons a Nil) Nil) Nil)
7 )
8
9 ; (((ar1)))
```

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

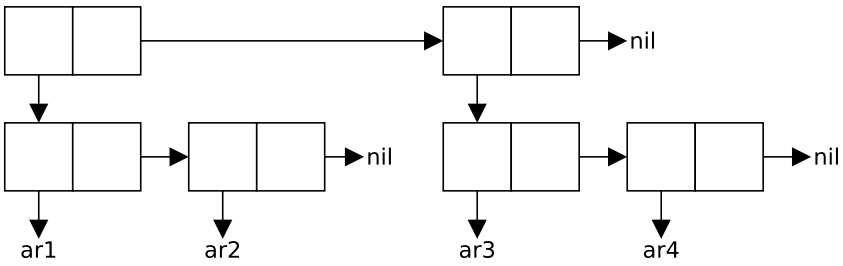

Рисунок 2.1 — Список ((ar1 ar2) (ar3 ar4))

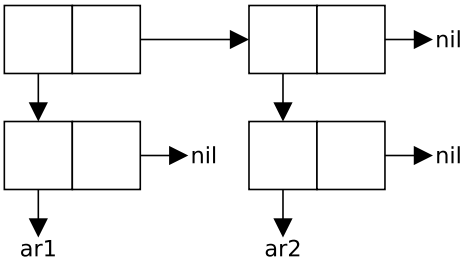

Рисунок 2.2 — Список ((ar1) (ar2))

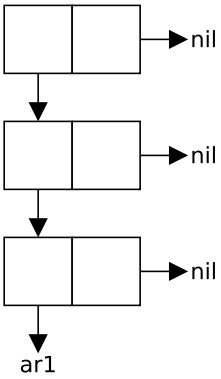

Рисунок 2.3 — Список (((ar1)))