

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 №1

Составить диаграмму вычисления следующих выражений

1 (equal 3 (abs (- 3)))

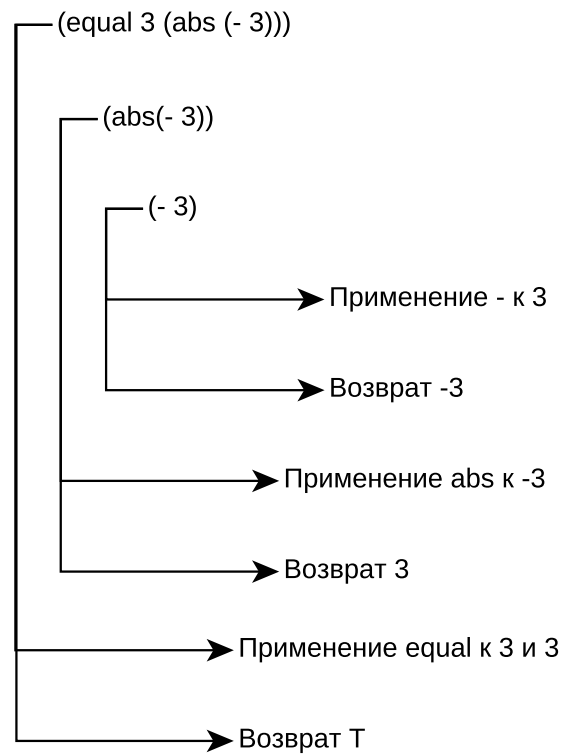


Рисунок 2.1 — Диаграмма для первого пункта

1 (equal (+ 1 2) 3)

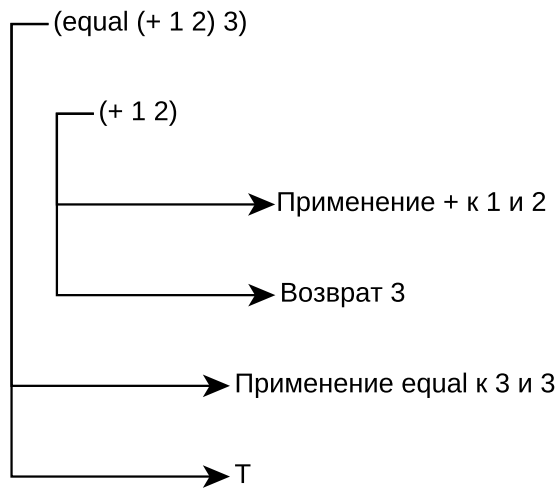


Рисунок 2.2 — Диаграмма для второго пункта

1 (equal (* 4 7) 21)

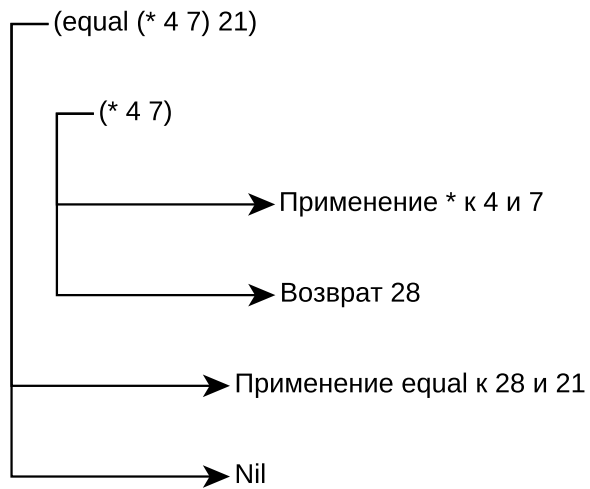


Рисунок 2.3 — Диаграмма для третьего пункта

1 (equal (* 2 3) (+ 7 2))

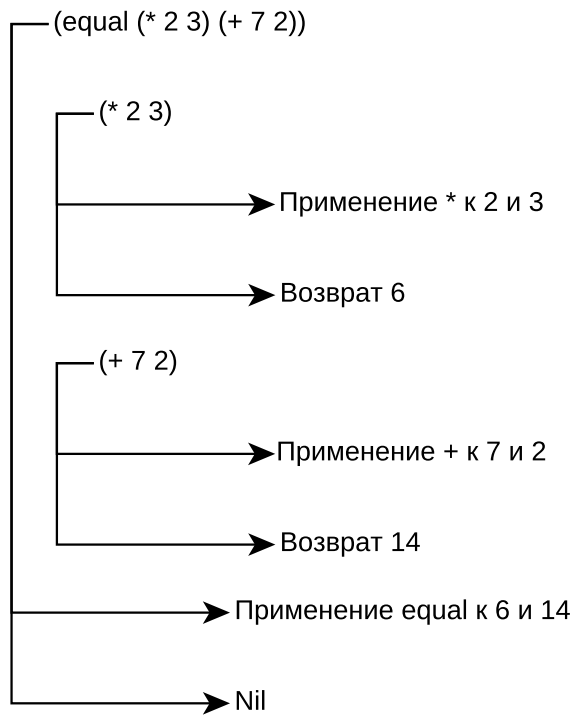


Рисунок 2.4 — Диаграмма для четвертого пункта

```
1 (equal (- 7 3) (* 3 2))
```

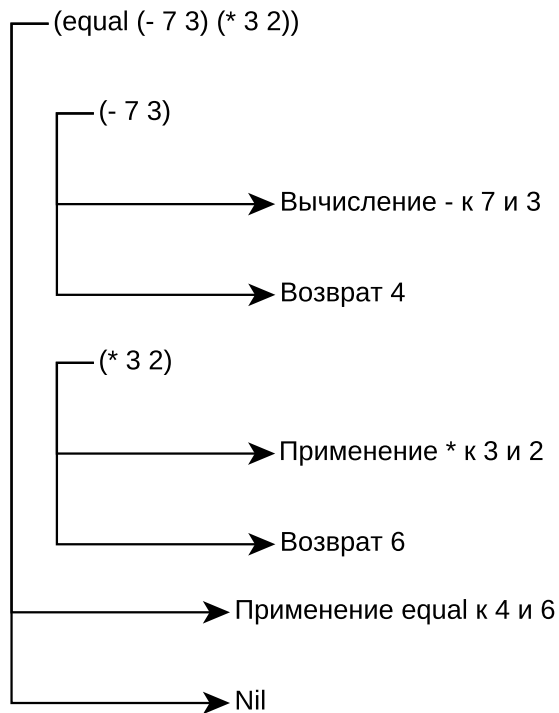


Рисунок 2.5 — Диаграмма для пятого пункта

```
1 (equal (abs (- 2 4)) 3))
```

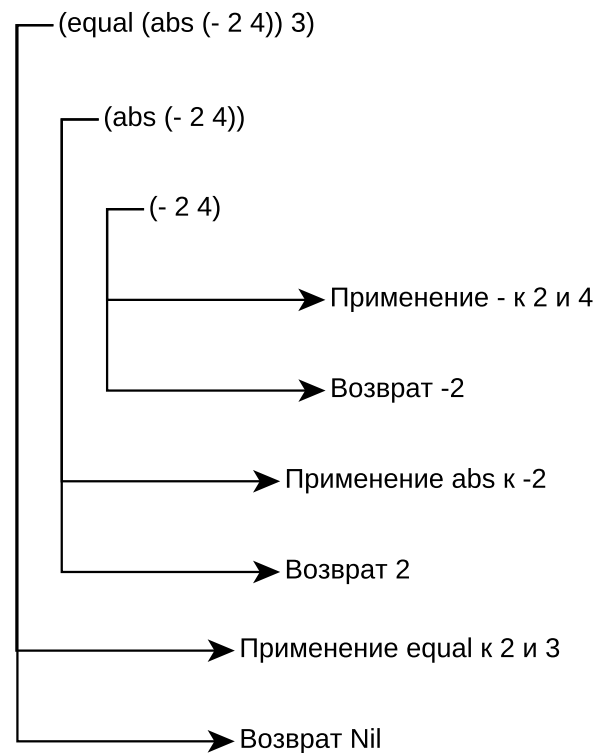


Рисунок 2.6 — Диаграмма для пятого пункта

2.2 Задание №2

Написать функцию, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму её вычисления.

```
1 (defun hypotenuse (a b)
2   (sqrt (+ (* a a) (* b b)))
3 )
4
5 (hypotenuse 3 4) ;;; 5.0
```

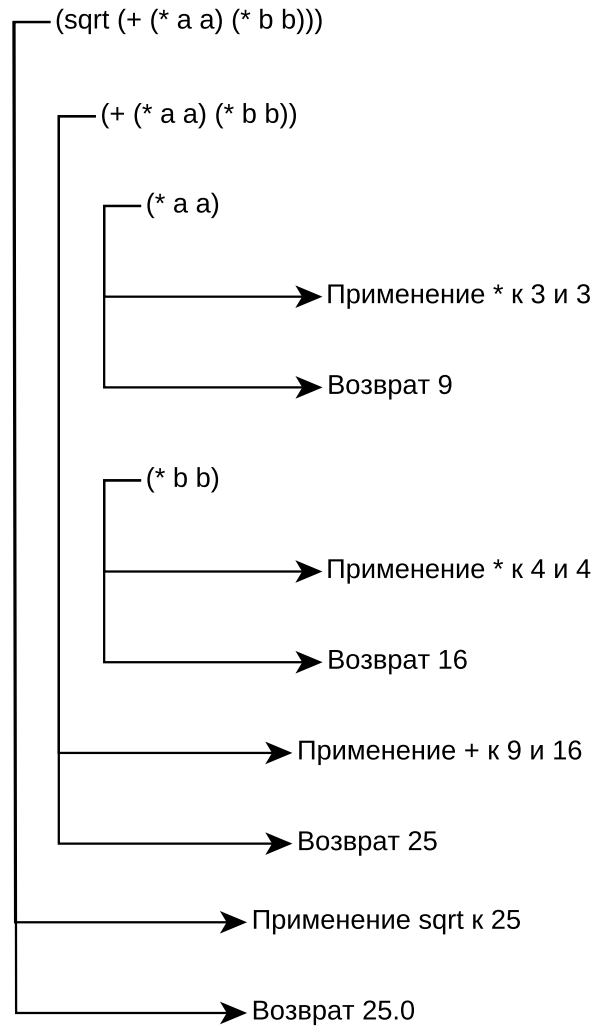


Рисунок 2.7 — Диаграмма для второго задания

2.3 Задание №3

Написать функцию, вычисляющую объем параллелепипеда по 3-м его сторонам, и составить диаграмму ее вычисления.

```

1 (defun volume (a b c)
2   (* a b c)
3 )
4
5 (volume 3 4 5) ;;; 60

```

Ниже представлена диаграмма вычисления данной программы.

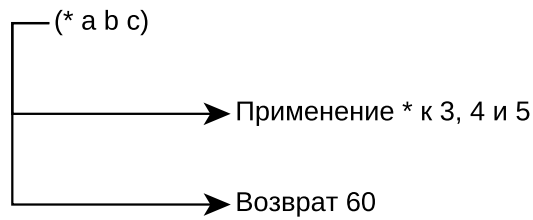


Рисунок 2.8 — Диаграмма для третьего задания

2.4 Задание №4

Каковы результаты вычисления следующих выражений:

```
1 (list 'a c)
```

Результат: ошибка - нет переменной C

```
1 (cons 'a (b c))
```

Результат: ошибка - нет переменной C

```
1 (cons 'a '(b c))
```

Результат: (A B C)

```
1 (caddy (1 2 3 4 5))
```

Результат: ошибка - нет функции CADDY

```
1 (cons 'a 'b 'c)
```

Результат: ошибка - неправильное количество аргументов должно быть 2

```
1 (list 'a (b c))
```

Результат: ошибка - нет переменной C

```
1 (list a '(b c))
```

Результат: ошибка - нет переменной A

```
1 (list (+ 1 '(length '(1 2 3))))
```

Результат: ошибка - значение (LENGTH '(1 2 3)) не числовое

2.5 Задание №5

Написать функцию `longer_then` от двух списков-аргументов, которая возвращает `T`, если первый аргумент имеет большую длину.

```
1 (defun longer_then (list1 list2)
2   (> (length list1) (length list2))
3 )
4
5 (longer_then '(1 2 3) '(1 2)) ;;; T
6 (longer_then '(1 2 3) '(1 2 3 4)) ;;; NIL
```

2.6 Задание №6

Каковы результаты вычисления следующих выражений?

```
1 (cons 3 (list 5 6))
```

Результат: (3 5 6)

```
1 (list 3 'from 9 'lives (- 9 3))
```

Результат: (3 FROM 9 LIVES 6)

```
1 (+ (length for 2 too) (car '(21 22 23)))
```

Результат: ошибка - нет переменной FOR

```
1 (cdr ' (cons is short for ans))
```

Результат: (IS SHORT FOR ANS)

```
1 (car (list one two))
```

Результат: ошибка - нет переменной ONE

```
1 (cons 3 '(list 5 6))
```

Результат: (3 LIST 5 6)

```
1 (car (list 'one 'two))
```

Результат: (ONE)