



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 «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 2
по дисциплине «Функциональное и
логическое программирование»

Тема Определение функций пользователя

Студент Царев А.А.

Группа ИУ7-63Б

Преподаватели Толпинская Н.Б., Строганов Ю.В.

Москва, 2023

ВВЕДЕНИЕ

Целью данной работы является приобретение навыков создания и использования функций пользователя в Lisp.

Задача работы: изучить работу интерпретатора Lisp, алгоритм работы функции eval, структуру и порядок обработки программ в Lisp.

1 Теоретические вопросы

1.1 Базис Lisp

Базис — минимальный набор конструкций языка, на основе которого могут быть построены вычисления.

Базис языка составляют:

- атомы;
- структуры;
- базовые функции;
- базовые функционалы.

1.2 Классификация функций

Функции в Lisp делятся на три класса:

- базисные,
- ядра,
- пользовательские.

Базисные функции, в свою очередь, классифицируются следующим образом:

- селекторы (car, cdr и т.д.),
- конструкторы (cons),
- предикаты (atom),
- сравнения (eq).

1.3 Способы создания функций

В Lisp можно определять безымянные функции и функции с именем.

Безымянная функция:

```
1 (lambda (x1 x2 ... xn) f)
```

Функция с именем:

```
1 (defun <name> (x1 x2 ... xn) f)
```

Вызов:

```
1 (<name> a1 a2 ... an)
2 ((lambda (x1 x2 ... xn) f) a1 a2 ... an)
```

1.4 Функции car, cdr, eq, eql, equal, equalp

Функция car разыменовывает первый указатель бинарного узла:

```
1 (car '(1 2 3))      -> 1
2 (car '((1 2) 3))    -> (1 2)
3 (car '(1 . 2))      -> 1
4 (car '(1))          -> 1
```

Функция cdr разыменовывает второй указатель бинарного узла:

```
1 (cdr '(1 2 3))      -> (2 3)
2 (cdr '((1 2) 3))    -> (3)
3 (cdr '(1 . 2))      -> 2
4 (car '(1))          -> Nil
```

Функция eq осуществляет сравнение только символьных атомов:

```
1 (eq 'a 'a)          -> T
2 (eq 'a 'b)          -> Nil
3 (eq 1 1)            -> T
4 (eq 2 1)            -> Nil
5 (eq 2 2.0)          -> Nil
6 (eq 2.00 2.0)       -> Nil
```

Функция eql осуществляет сравнение между символьных атомов или чисел:

```
1 (eql 'a 'a)         -> T
2 (eql 'a 'b)         -> Nil
3 (eql 1 1)           -> T
4 (eql 2 1)           -> Nil
5 (eql 2 2.0)         -> Nil
6 (eql 2.00 2.0)      -> T
```

Функция equal осуществляет сравнение символьных атомов, чисел или списков:

```
1 (equal 'a 'a)       -> T
2 (equal 'a 'b)       -> Nil
3 (equal 1 1)         -> T
4 (equal 2 1)         -> Nil
5 (equal 2 2.0)       -> Nil
6 (equal 2.00 2.0)    -> T
7 (equal '(1 2 3) '(1 2 3)) -> T
8 (equal '(1 2 3) '(1 2 4)) -> Nil
```

Функция equalp осуществляет сравнение символьных атомов, чисел разных видов или списков:

```
1 (equalp 'a 'a)           -> T
2 (equalp 'a 'b)           -> Nil
3 (equalp 1 1)             -> T
4 (equalp 2 1)             -> Nil
5 (equalp 2 2.0)           -> T
6 (equalp 2.00 2.0)        -> T
7 (equalp '(1 2 3) '(1 2 3)) -> T
8 (equalp '(1 2 3) '(1 2 4)) -> Nil
```

1.5 Назначение и отличие в работе функций cons и list

Функция cons создает бинарный узел, первый указатель которого ссылается на значение первого переданного аргумента, а второй — на значение второго.

Функция list создает список, состоящий из переданных функции аргументов. Если функции не были переданы фактические параметры, то создается пустой список.

Основные отличия в работе функций cons и list:

- функция cons принимает фиксированное количество аргументов, функция list — произвольное;
- функция cons создает один бинарный узел, функция list — список.

2 Практические задания

2.1 Задание 1. Составить диаграмму вычисления выражений

На рисунках 2.1 – 2.6 представлены диаграммы вычисления выражений.

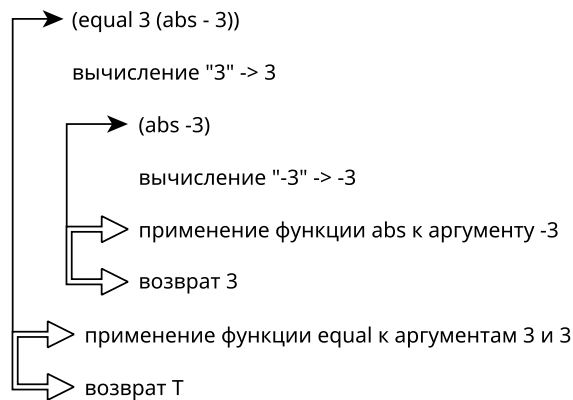


Рисунок 2.1 — Диаграмма вычисления выражения `(equal 3 (abs - 3))`

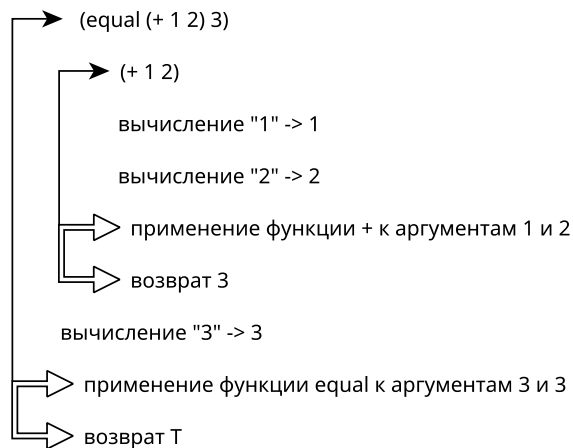


Рисунок 2.2 — Диаграмма вычисления выражения `(equal (+ 1 2) 3)`

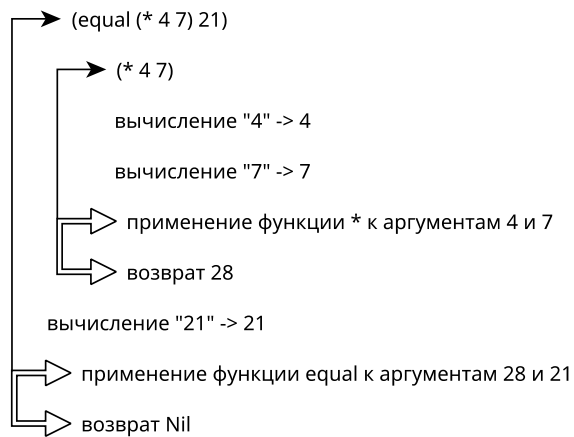


Рисунок 2.3 — Диаграмма вычисления выражения (equal (* 4 7) 21)

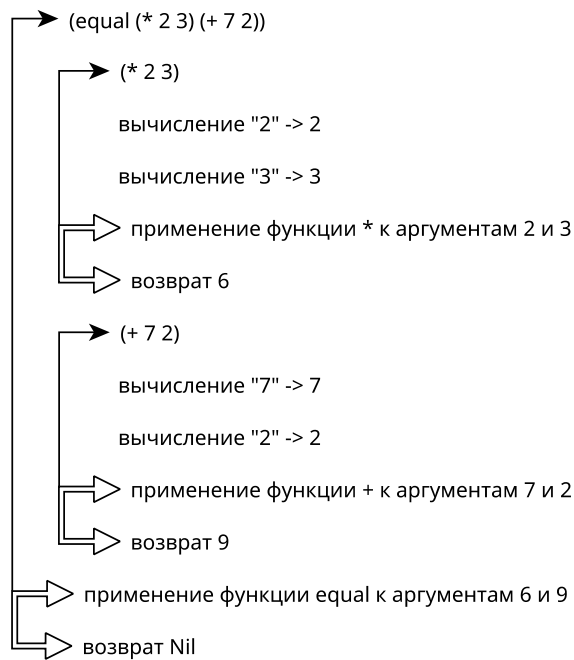


Рисунок 2.4 — Диаграмма вычисления выражения (equal (* 2 3) (+ 7 2))

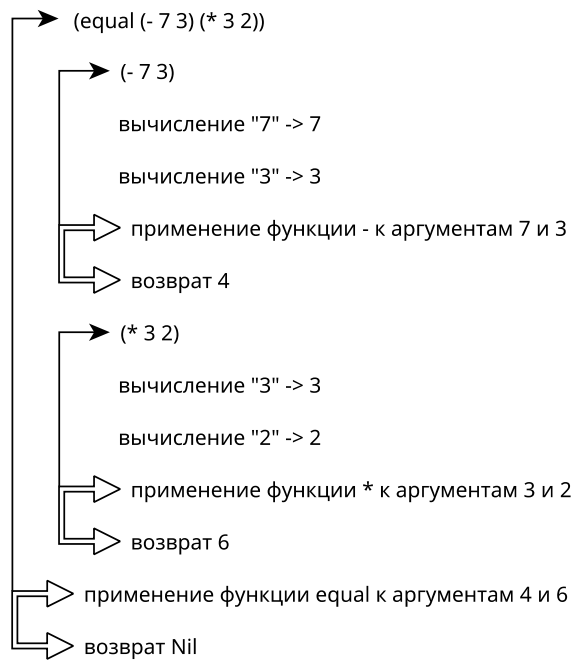


Рисунок 2.5 — Диаграмма вычисления выражения (equal (- 7 3) (* 3 2))

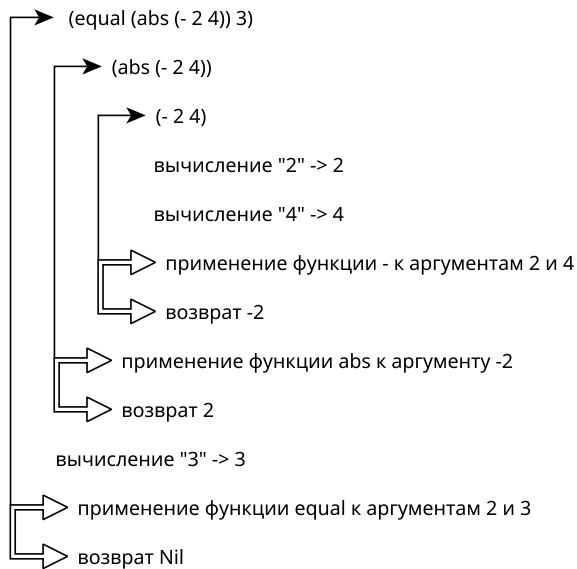


Рисунок 2.6 — Диаграмма вычисления выражения (equal (abs (- 2 4)) 3)

2.2 Задание 2. Написать функцию, вычисляющую гипотенузу прямоугольного треугольника по заданным катетам и составить диаграмму её вычисления

```

1 (defun hyp (a b)
2   (sqrt
3     (+  ( * a a)
4         ( * b b)))))

```

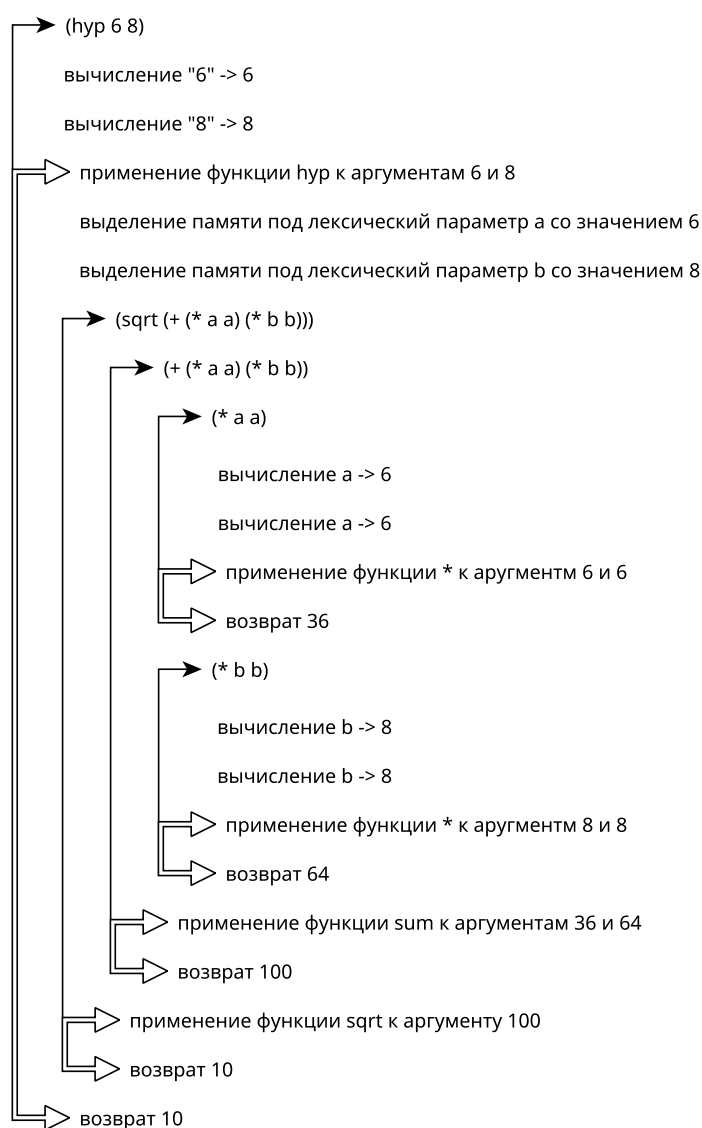


Рисунок 2.7 — Диаграмма вычисления выражения (hyp 6 8)

2.3 Задание 3. Каковы результаты вычисления следующих выражений?

Таблица 2.1 — Результаты вычисления выражений

Выражение	Результат	Исправление
(list 'a c)	variable C has no value	(list 'a 'c)
(cons 'a (b c))	undefined function B	(cons 'a '(b c))
(cons 'a '(b c))	(a b c)	—
(caddr (1 2 3 4 5))	1 is not a function name	(caddr '(1 2 3 4 5))
(cons 'a 'b 'c)	too many arguments given to CONS	(cons 'a '(b c))
(list 'a (b c))	undefined function B	(list 'a '(b c))
(list a '(b c))	variable A has no value	(list 'a '(b c))
(list (+ 1 '(length '(1 2 3))))	(length '(1 2 3)) is not a number	(list (+ 1 (length '(1 2 3))))

2.4 Задание 4. Написать функцию longer_then от двух списков-аргументов, которая возвращает Т, если первый аргумент имеет большую длину

```

1 (defun longer_then (list1 list2)
2   (> (length list1)
3      (length list2)))

```

2.5 Задание 5. Каковы результаты вычисления следующих выражений?

Таблица 2.2 — Результаты вычисления выражений

Выражение	Результат
(cons 3 (list 5 6))	(3 5 6)
(list 3 'from 9 'lives (- 9 3))	(3 from 9 lives 6)
(+ (length for 2 too)) (car '(21 22 23)))	variable FOR has no value
(cdr '(cons is short for ans))	(is short for ans)

(car (list one two))	variable ONE has no value
(cons 3 '(list 5 6))	(3 list 5 6)
(car (list 'one 'two))	one

2.6 Задание 6. Дана функция

(defun mystery (x) (list (second x) (first x))). Какие результаты вычисления следующих выражений?

Таблица 2.3 — Результаты вычисления выражений

Выражение	Результат
(mystery (one two))	undefined function ONE
(mystery (last one two))	variable ONE has no value
(mystery free)	variable FREE has no value
(mystery one 'two))	variable ONE has no value

2.7 Задание 7. Написать функцию, которая переводит температуру в системе Фаренгейта температуру по Цельсию (defun f-to-c (temp) . . .). Как бы назывался роман Р.Брэдбери «451 по Фаренгейту» в системе по Цельсию?

```

1 (defun f-to-c (temp)
2   ( * (/ 5 9)
3     (- temp 32.0)))

```

Роман «451 по Фаренгейту» назывался бы «232.77779 по Цельсию».

2.8 Что получится при вычисления каждого из выражений?

Таблица 2.4 — Результаты вычисления выражений

Выражение	Результат
(list 'cons t NIL)	(cons t NIL)
(eval (eval (list 'cons t NIL)))	undefined function T
(apply #cons "(t NIL))	bad syntax for complex number: #CONS
(list 'eval NIL)	(eval NIL)
(eval (list 'cons t NIL))	(t)
(eval NIL)	NIL
(eval (list 'eval NIL))	NIL

ЗАКЛЮЧЕНИЕ

В данной работе были приобретены навыков создания и использования функций пользователя в Lisp.

Выполнена задача работы: изучены работа интерпретатора Lisp, алгоритм работы функции eval, структура и порядок обработки программ в Lisp.