



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 и системы управления»
КАФЕДРА _____ «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ
по лабораторной работе №2
по курсу «Функциональное и Логическое программирование»
на тему: «Определение функций пользователя»

Студент	<u>ИУ7-63Б</u>	_____	<u>Лагутин Д. В.</u>
	(Группа)	(Подпись, дата)	(Фамилия И. О.)
Преподаватель		_____	<u>Толпинская Н. Б.</u>
		(Подпись, дата)	(Фамилия И. О.)

Москва, 2023 г.

Теоретические вопросы

1. Базис Lisp

Базис — минимальный набор инструментов и структур данных языка, который позволяет реализовать любую поставленную задачу.

Базис языка представлен:

- атомами;
- структурами;
- функциями

atom, eq, cons, car, cdr, cond, quote, lambda, eval, label.

2. Классификация функций

- Чистые — не зависят от внешних, глобальных данных, не создают побочных эффектов.
- Формы:
 - могут иметь переменное количество параметров;
 - к аргументам может применяться особая обработка.
- Функционалы:
 - могут принимать функцию в качестве аргумента;
 - могут возвращать функцию.

Классификация базисных функций:

- селекторы;
- конструкторы;
- предикаты;

— функции сравнения.

3. Способы создание функций

Функция может быть определена двумя способами. С помощью λ -выражения $(\text{lambda } (\lambda\text{-list}) f)$, где $\lambda\text{-list}$ — список формальных аргументов, а f - тело функции, или макро-определения $(\text{defun name } \lambda\text{-выражение})$, где name — имя определяемой функции.

4. Функции `car` и `cdr`, `eq`, `eq1`, `equal`, `equalp`

`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`.

Функции сравнения (принимают два аргумента, перечислены по мере усложнения проверки):

- `eq` корректно сравнивает два символьных атома. Так как атомы не дублируются для данного сеанса работы, то фактически сравниваются соответствующие указатели;
- `eq1` корректно сравнивает атомы и числа одинакового типа;
- `=` корректно сравнивает только числа, причем числа могут быть разных типов;
- `equal` работает идентично `eq1`, но в дополнение умеет корректно сравнивать списки;
- `equalp` корректно сравнивает любые S-выражения.

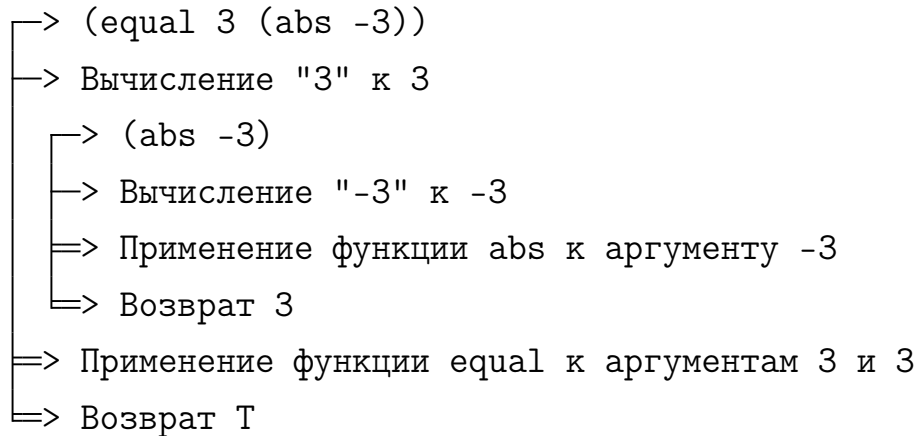
5. Назначение и отличие в работе cons и list

`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.

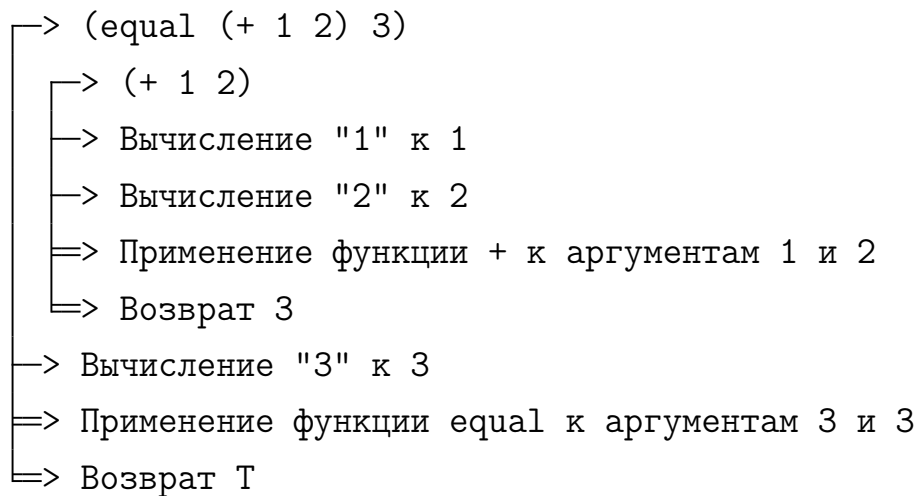
1 Задание

Составить диаграмму вычисления следующих выражений:

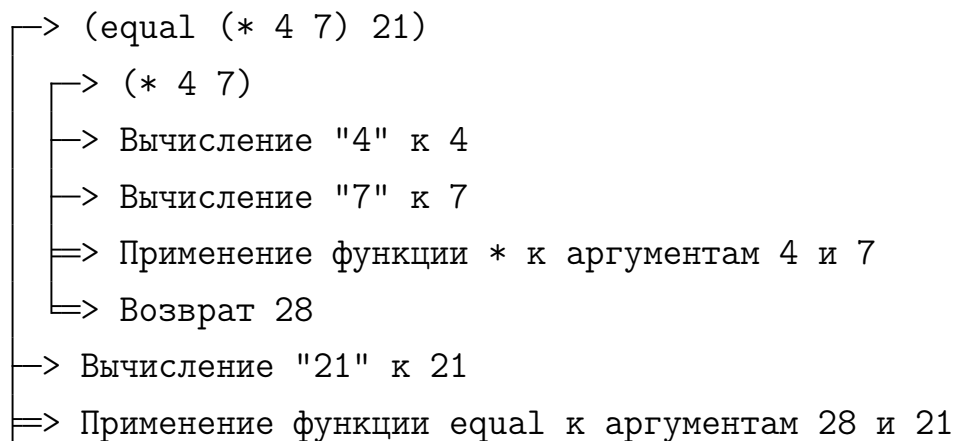
1) (equal 3 (abs -3))



2) (equal (+ 1 2) 3)



3) (equal (* 4 7) 21)



↳ Возврат nil

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

↳ (equal (* 2 3) (+ 7 2))

↳ (* 2 3)

↳ Вычисление "2" к 2

↳ Вычисление "3" к 3

⇒ Применение функции * к аргументам 2 и 3

⇒ Возврат 6

↳ (+ 7 2)

↳ Вычисление "7" к 7

↳ Вычисление "2" к 2

⇒ Применение функции + к аргументам 7 и 2

⇒ Возврат 9

⇒ Применение функции equal к аргументам 6 и 9

⇒ Возврат nil

5) (equal (- 7 3) (* 3 2))

↳ (equal (- 7 3) (* 3 2))

↳ (- 7 3)

↳ Вычисление "7" к 7

↳ Вычисление "3" к 3

⇒ Применение функции - к аргументам 7 и 3

⇒ Возврат 4

↳ (* 3 2)

↳ Вычисление "3" к 3

↳ Вычисление "2" к 2

⇒ Применение функции * к аргументам 3 и 2

⇒ Возврат 6

⇒ Применение функции equal к аргументам 4 и 6

⇒ Возврат nil

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

—> (equal (* 2 3) (+ 7 2))

—> (* 2 3)

—> Вычисление "2" к 2

—> Вычисление "3" к 3

==> Применение функции * к аргументам 2 и 3

==> Возврат 6

—> (+ 7 2)

—> Вычисление "7" к 7

—> Вычисление "2" к 2

==> Применение функции + к аргументам 7 и 2

==> Возврат 9

==> Применение функции equal к аргументам 6 и 9

==> Возврат nil

2 Задание

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

```
1 (defun hypotenuse (a b) (sqrt (+ (* a a) (* b b))))
```

(hypotenuse 3 4)

```
→ (hypotenuse 3 4)
→ Вычисление "3" к 3
→ Вычисление "4" к 4
⇒ Вызов функции hypotenuse с аргументами 3 и 4
  Создание переменной a со значением 3
  Создание переменной b со значением 4
    → (sqrt (+ (* a a) (* b b)))
      → (+ (* a a) (* b b))
        → (* a a)
          → Вычисление a к 3
          ⇒ Применение функции * к аргументам 3 и 3
          ⇒ Возврат 9
        → (* b b)
          → Вычисление b к 4
          ⇒ Применение функции * к аргументам 4 и 4
          ⇒ Возврат 16
        ⇒ Применение функции + к аргументам 9 и 16
        ⇒ Возврат 25
      ⇒ Применение функции sqrt к аргументу 25
      ⇒ Возврат 5.0
    ⇒ Возврат 5.0
```


3 Задание

Каковы результаты вычисления следующих выражений? (объяснить возможную ошибку и варианты ее устранения)

```
1 (list 'a c)
2 ;; Ошибка
3 ;; The variable C is unbound
4 ;; Ошибка возникает изза- того, что c - неизвестный для интерпретатора атом.
5 ;; Для устранения ошибки необходимо определить переменную c или выключить
6 ;; вычисление аргумента, добавив перед ним '
7 (list 'a 'c)
8 ;; (A C)
9
10 (cons 'a (b c))
11 ;; Ошибка
12 ;; The variable C is unbound.
13 ;; выражениеS- (b c) подразумевает, что b - функция, а c - ее аргумент.
14 ;; Аналогично предыдущему примеру, значение c не может быть вычислено, при
15 ;; этом b не является функцией. Вариант исправление: представить (b c) в виде
16 ;; списка
17 (cons 'a (b c))
18 ;; (A B C)
19
20 (cons 'a '(b c))
21 ;; Корректно
22 ;; (A B C)
23
24 (caddr (1 2 3 4 5))
25 ;; Ошибка
26 ;; Execution of a form compiled with errors.
27 ;; Form:
28 ;; (1 2 3 4 5)
29 ;; Compile-time error:
30 ;; illegal function call
31 ;; выражениеS- (1 2 3 4 5) было воспринято как форма, исправление - блокировка
32 ;; вычислений
33 (caddr '(1 2 3 4 5))
34 ;; 3
35
36 (cons 'a 'b 'c)
37 ;; Ошибка
38 ;; invalid number of arguments: 3
39 ;; Функция cons принимает лишь 2 аргумента. Вариант исправления - применить
40 ;; функцию list
41 (list 'a 'b 'c)
```

```

42 ;; (A B C)
43
44 (list 'a (b c))
45 ;; Ошибка
46 ;; The variable C is unbound.
47 ;; выражениеS- (b c) подразумевает, что b - функция, а c - ее аргумент.
48 ;; Аналогично первому примеру, значение c не может быть вычислено, при
49 ;; этом b не является функцией. Вариант исправление: представить (b c) в виде
50 ;; списка
51 (list 'a '(b c))
52 ;; (A (B C))
53
54 (list a '(b c))
55 ;; Ошибка
56 ;; The variable A is unbound
57 ;; Ошибка возникает изза- того, что a - неизвестный для интерпретатора атом.
58 ;; Для устранения ошибки необходимо определить переменную a или выключить
59 ;; вычисление аргумента, добавив перед ним '
60 (list 'a '(b c))
61 ;; (A (B C))
62
63 (list (+ 1 '(length '(1 2 3))))
64 ;; Ошибка
65 ;; The value
66 ;; (LENGTH '(1 2 3))
67 ;; is not of type
68 ;; NUMBER
69 ;; выражениеS- '(length '(1 2 3)) - список, хотя подразумевается вызов функции
70 ;; length. Решение - разблокировка вычислений
71 (list (+ 1 (length '(1 2 3))))
72 ;; (4)

```

4 Задание

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

```
1 (defun longer-then (a b) (> (length a) (length b)))
```

5 Задание

Каковы результаты вычисления следующих выражений?

```
1 (cons 3 (list 5 6))
2 ;; (3 5 6)
3
4 (cons 3 '(list 5 6))
5 ;; (3 LIST 5 6)
6
7 (list 3 'from 9 'lives (- 9 3))
8 ;; (3 FROM 9 LIVES 6)
9
10 (+ (length for 2 too)) (car '(21 22 23)))
11 ;; Ошибка
12 ;; The variable FOR is unbound.
13
14 (cdr '(cons is short for ans))
15 ;; (IS SHORT FOR ANS)
16
17 (car (list one two))
18 ;; Ошибка
19 ;; The variable ONE is unbound.
20
21 (car (list 'one 'two))
22 ;; ONE
```

6 Задание

Дана функция (defun mystery (x) (list (second x) (first x))).

Какие результаты вычисления следующих выражений?

```
1 (mystery (one two))
2 ;; Ошибка
3 ;; The variable TWO is unbound.
4
5 (mystery one 'two))
6 ;; Ошибка
7 ;; The variable ONE is unbound.
8
9 (mystery (last one two))
10 ;; Ошибка
11 ;; The variable ONE is unbound.
12
13 (mystery free)
14 ;; Ошибка
15 ;; The variable FREE is unbound.
```

7 Задание

Написать функцию, которая переводит температуру в системе Фаренгейта температуру по Цельсию (defun f-to-c (temp) ...).

```
1 (defun f-to-c (temp) (* (/ 5.0 9.0) (- temp 32)))
```

Как бы назывался роман Р.Брэдбери «+451 по Фаренгейту» в системе по Цельсию? «232.77779 по Цельсию»

8 Задание

Что получится при вычисления каждого из выражений?

```
1 (list 'cons t NIL)
2 ;; (CONS T NIL)
3
4 (eval (list 'cons t NIL))
5 ;; (T)
6
7 (eval (eval (list 'cons t NIL)))
8 ;; Ошибка
9 ;; The function COMMON-LISP:T is undefined.
10
11 (apply #cons "(t NIL)")
12 ;; Ошибка
13 ;; illegal complex number format: #CONS
14
15 (list 'eval NIL)
16 ;; (EVAL NIL)
17
18 (eval NIL)
19 ;; NIL
20
21 (eval (list 'eval NIL))
22 ;; NIL
```