



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»

Отчёт

по лабораторной работе №1

Название: Определение функций пользователя.

Дисциплина: Функциональное и логическое программирование

Студент ИУ7-64Б
(Группа)

Л.Е.Тартыков
(Подпись, дата) (И.О. Фамилия)

Преподаватель

Н.Б.Толпинская
(Подпись, дата) (И.О. Фамилия)

Преподаватель

Ю.В.Строганов
(Подпись, дата) (И.О. Фамилия)

Москва, 2022

1 Теоретические вопросы

1.1 Базис Lisp

Базис - минимальный набор конструкций языка и структур данных, с помощью которых можно решить любую задачу.

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

Базисные функции – минимальный набор функций, которые позволяют решить любую задачу.

1.2 Классификация функций

1. Чистые математические функции.

Имеют фиксированное число аргументов и возвращают один результат. Сначала вычисляются все аргументы, затем к ним применяется исходная функция.

2. Специальные функции (формы).

Специальные функции – функции, у которых переменное число аргументов или они обрабатываются по-разному (один вычисляется, другой - нет).

3. Псевдофункции.

Псевдофункции – функции, которые создают "спецэффекты"; например, вывод на экран.

4. Функции с вариантами значений – выбирают какое-то одно значение.

5. Функционалы.

Функционалы – функции, которые в качестве аргументов используют функции или возвращают в качестве результата функцию. Также они называются *функциями более высокого порядка*. Позволяют создавать

синтаксически управляемые программы (программы, которые сами создают какие-то функции; эти функции затем выполняются).

6. Рекурсивные.

1.3 Способы создания функций

1. *lambda*-выражение. Данный способ представлен с помощью формулы (1.1).

$$(lambda \lambda\text{-список форма}), \quad (1.1)$$

где λ -список – список формальных параметров, форма – тело функции. *lambda*-выражение не хранится в памяти и не имеет имени. Вычисляется сразу же. Используется для повторных вычислений.

Вызов *lambda*-функции выполняется по формуле (1.2).

$$(\lambda\text{-выражение последовательность форм}) \quad (1.2)$$

2. С помощью *defun* по формуле (1.3).

$$(defun f \lambda\text{-выражение}) \quad (1.3)$$

Система по имени символьного атома находит его определение.

1.4 Функции Car и Cdr

car и *cdr* являются базовыми функциями доступа к данным.

car – принимает на вход один аргумент и возвращает первый элемент списка.

Пример использования *car* и списка представлен на листинге 1.1.

Листинг 1.1 – Использование *car* и списка.

1

<pre>(car '(A B C D)) ;; (A)</pre>

cdr – принимает на вход один аргумент и возвращает хвост списка.

Пример использования *cdr* и списка представлен на листинге 1.2.

Листинг 1.2 – Использование *cdr* и списка.

```
1 (cdr '(A B C D))      ;; (B C D)
```

1.5 Назначение и отличие в работе Cons и List

cons – имеет два аргумента и возвращает бинарный узел. Если вторым аргументом является атом, то возвращается точечная пара; если список – список.

list – имеет произвольное число аргументов и возвращает список.

Пример использования *cons* представлен на листинге 1.3.

Листинг 1.3 – Использование *cons*.

```
1 (cons 'A 'B)          ;; (A.B)
2 (cons 'A '(B C D))    ;; (A B C D)
3 (cons '(A B) '(C D))  ;; ((A B) C D)
```

Пример использования *list* представлен на листинге 1.4.

Листинг 1.4 – Использование *list*.

```
1 (list 'A 'B)          ;; (A B)
2 (list 'A '(B C) 'D)   ;; (A (B C) D)
```

функция *list* может быть представлена с помощью *cons* (листинг 1.5).

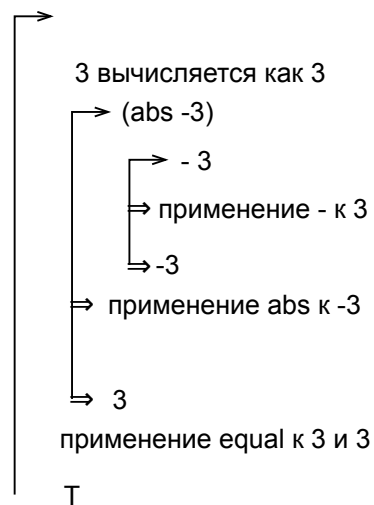
Листинг 1.5 – Представление *list* с помощью *cons*.

```
1 (defun list2 (ar1 ar2) (cons ar1 (cons ar2 ())))
2 (defun list3 (ar1 ar2 ar3) (cons ar1 (cons ar2 (cons ar3 ())))
3 ...
```

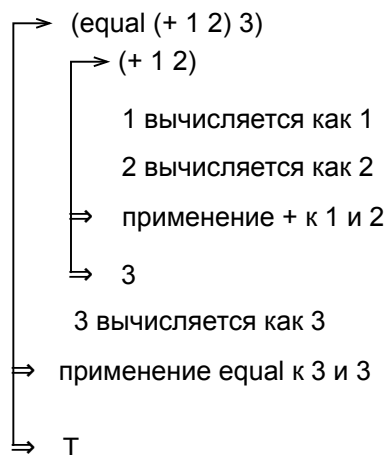
2 Практические задания

2.1 Составить диаграмму вычисления следующих выражений:

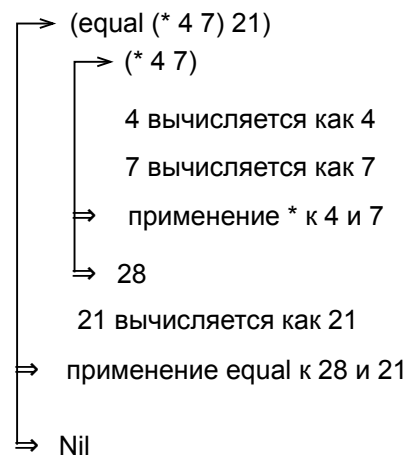
1. $(\text{equal } 3 (\text{abs } -3))$



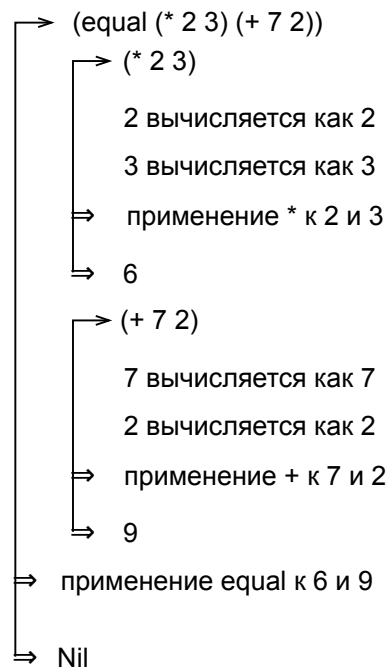
2. $(\text{equal } (+ 1 2) 3)$



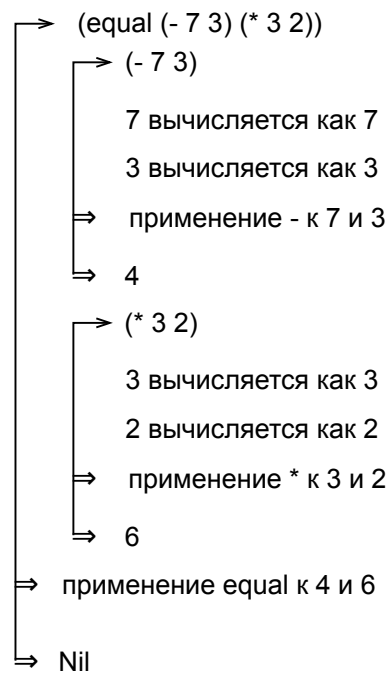
3. (equal (* 4 7) 21)



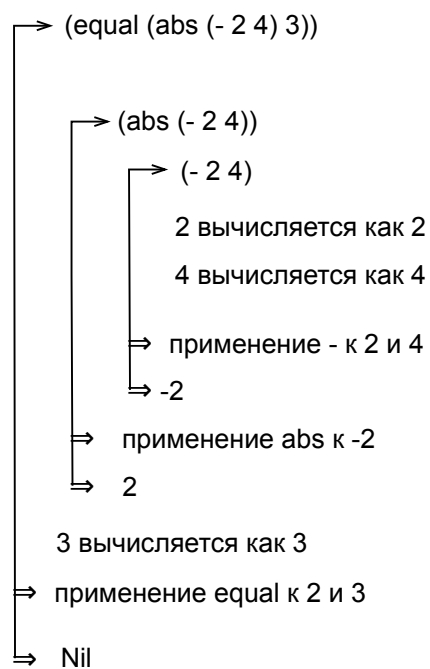
4. (equal (* 2 3) (+ 7 2))



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



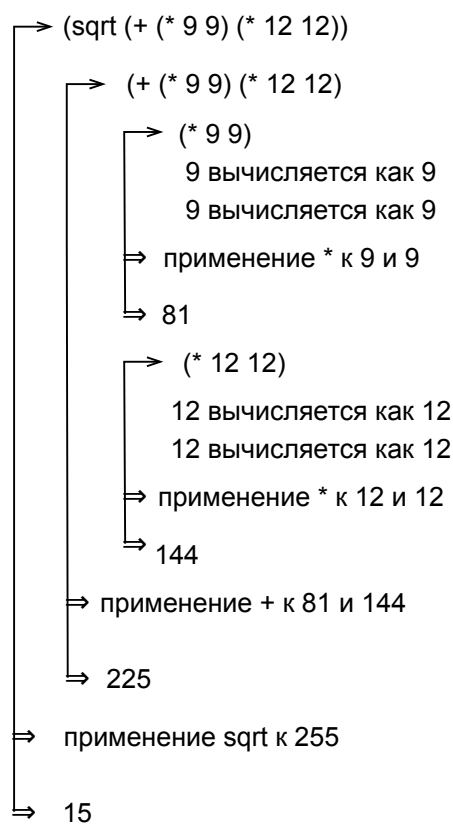
6. (equal (abs (- 2 4) 3))



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

Листинг 2.1 – Задание 2.

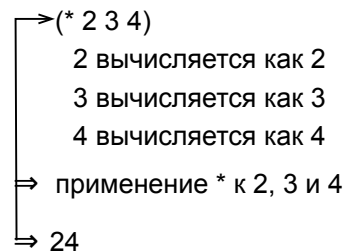
```
1 (defun hypotenuse (cath1 cath2)
2   (sqrt (+ (* cath1 cath1) (* cath2 cath2))))
```



2.3 Написать функцию, вычисляющую объем параллелепипеда по трем его сторонам, и составить диаграмму ее вычисления.

Листинг 2.2 – Задание 3.

```
1 (defun volume (side1 side2 side3) (* side1 side2 side3))
```



2.4 Каковы результаты вычисления следующих выражений? Объяснить возможную ошибку и варианты ее устранения

1. (list 'a c) ;ошибка \Rightarrow (list 'a c)
2. (cons 'a (b c)) ;ошибка \Rightarrow (cons 'a '(b c))
3. (cons 'a '(b c)) \Rightarrow (a b c)
4. (caddy (1 2 3 4 5)) ;ошибка \Rightarrow (caddr (1 2 3 4 5)) \Rightarrow ошибка \Rightarrow
;(caddr '(1 2 3 4 5))
5. (cons 'a 'b 'c) ;ошибка \Rightarrow (cons 'a ' ('b 'c))
6. (list 'a (b c)) ;ошибка \Rightarrow (list 'a '(b c))
7. (list a '(b c)) ;ошибка \Rightarrow (list 'a (b c))
8. (list (+ 1 '(length '(1 2 3)))) ;ошибка \Rightarrow (list (+ 1 (length '(1 2 3))))

2.5 Написать функцию `longer_then` от двух списковых аргументов, которая возвращает Т, если первый аргумент имеет большую длину.

Листинг 2.3 – Задание 5.

```
1 (defun longer_then (arg1 arg2) (> (length arg1) (length arg2)))
```

2.6 Каковы результаты вычисления следующих выражений?

1. `(cons 3 (list 5 6))` \Rightarrow `(3 5 6)`
2. `(cons 3 '(list 5 6))` \Rightarrow `(3 list 5 6)`
3. `(list 3 'from 9 'lives (- 9 3))` \Rightarrow `(3 from 9 lives 6)`
4. `(+ (length for 2 too))` \Rightarrow ошибка
5. `(car '(21 22 23)))` \Rightarrow ошибка
6. `(cdr '(cons is short for ans))` \Rightarrow `(is short for ans)`
7. `(car (list one two))` \Rightarrow ошибка
8. `(car (list 'one 'two))` \Rightarrow `one`

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

1. (mystery (one two)) \Rightarrow ошибка
2. (mystery (last one two)) \Rightarrow ошибка
3. (mystery free) \Rightarrow ошибка
4. (mystery one 'two)) \Rightarrow ошибка

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

Формулы:

1. $c = 5/9 * (f - 320);$

Листинг 2.4 – Перевод из Фаренгейт в Цельсий.

```
1 (defun f_to_c (temp) (* (/ 5 9) (- temp 32.0)))
```

2. $f = 9/5 * c + 32.0;$

Листинг 2.5 – Перевод из Цельсия в Фаренгейт.

```
1 (defun c_to_f (temp) (+ (* (/ 9 5) temp) 32.0))
```

Как бы назывался роман Р.Брэдбери «+451 по Фаренгейту» в системе по Цельсию? «+232.778 по Цельсию»

2.9 Что получится при вычислении каждого из выражений?

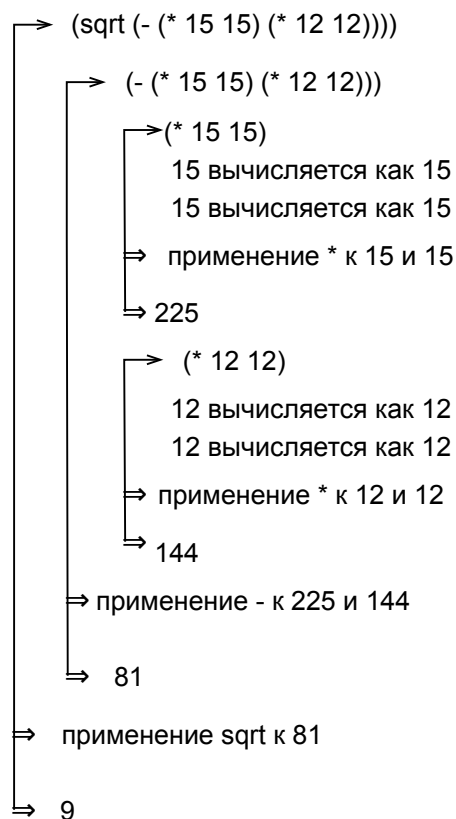
1. $(\text{list } 'cons \ t \ NIL) \Rightarrow (\text{cons } t \ Nil)$
2. $(\text{eval } (\text{list } 'cons \ t \ NIL))$
 $(\text{eval } (\text{cons } t \ Nil))$
 $(\text{eval } (t)) \Rightarrow \text{ошибка}$
3. $(\text{eval } (\text{eval } (\text{list } 'cons \ t \ NIL)))$
 $(\text{eval } (\text{eval } (\text{cons } t \ Nil)))$
 $(\text{eval } (\text{eval } (t))) \Rightarrow \text{ошибка}$
4. $(\text{apply } \#cons \ '(t \ NIL)) \Rightarrow \text{ошибка}$
 $(\text{apply } \#'cons \ '(t \ NIL)) \Rightarrow (t)$
5. $(\text{list } 'eval \ NIL) \Rightarrow (\text{eval } NIL)$
6. $(\text{eval } NIL) \Rightarrow Nil$
7. $(\text{eval } (\text{list } 'eval \ NIL)) \Rightarrow Nil$

Дополнительные задания

2.10 Написать функцию, вычисляющую катет по заданной гипотенузе и другому катету прямоугольного треугольника, составить диаграмму ее вычисления.

Листинг 2.6 – Дополнительное задание 1.

```
1 (defun hypo_cath (hypo cath)
2   (sqrt (- (* hypo hypo) (* cath cath))))
```



2.11 Написать функцию, вычисляющую площадь трапеции по ее основаниям и высоте, составить диаграмму ее вычисления.

Листинг 2.7 – Дополнительное задание 2.

```
1 (defun trapezoid_square (base1 base2 height)
2   (* 0.5 (+ base1 base2) height))
```

