



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»

Лабораторная работа №2 по курсу "Функциональное и логическое программирование"

Тема Определение функций пользователя

Студент Ковалец К. Э.

Группа ИУ7-63Б

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

Москва — 2022 г.

1 Практические задания

Все диаграммы вычислений приложены к отчету.

1.1 Задание 1

Составить диаграмму вычисления следующих выражений.

Листинг 1.1 – Выражения для построения диаграмм в задании 1

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

1.2 Задание 2

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

Листинг 1.2 – Решение задания 2

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

1.3 Задание 3

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

Листинг 1.3 – Решение задания 3

```
1 (defun volume(a b c) (* a b c))
2 ;; (VOLUME 2 3 4) -> 24
```

1.4 Задание 4

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

Листинг 1.4 – Решение задания 4

```
1 (list 'a c)           ;; The variable C is unbound.
2 (cons 'a (b c))       ;; The variable C is unbound
3 (cons 'a '(b c))      ;; (A B C)
4 (caddy (1 2 3 4 5))   ;; Execution of a form compiled with errors.
5 (cons 'a 'b 'c)       ;; invalid number of arguments: 3
6 (list 'a (b c))       ;; The variable C is unbound.
7 (list a '(b c))       ;; The variable A is unbound.
8 (list (+ 1 '(length '(1 2 3)))) ;; The value (LENGTH '(1 2 3)) is not of
    type NUMBER
```

Листинг 1.5 – Варианты следующих выражений с устраненными ошибками

```
1 (list 'a 'c)           ;; (A C)
2 (cons 'a '(b c))       ;; (A B C)
3 (cons 'a '(b c))       ;; (A B C)
4 (caddr '(1 2 3 4 5))   ;; 3
5 (cons 'a 'b)           ;; (A . B)
6 (list 'a '(b c))       ;; (A (B C))
7 (list 'a '(b c))       ;; (A (B C))
8 (list (+ 1 (length '(1 2 3)))) ;; (4)
```

1.5 Задание 5

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

Листинг 1.6 – Решение задания 5

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

1.6 Задание 6

Каковы результаты вычисления следующих выражений?

В 4-ом выражении была исправлена ошибка. Изначально было так:

Листинг 1.7 – Исходное 4-ое выражение

```
1 (+ (length for 2 too)) (car '(21 22 23))) ;; The variable FOR is unbound.
```

Листинг 1.8 – Решение задания 6

```
1 (cons 3 (list 5 6))                ;; (3 5 6)
2 (cons 3 '(list 5 6))                ;; (3 LIST 5 6)
3 (list 3 'from 9 'lives (- 9 3)) ;; (3 FROM 9 LIVES 6)
4 (+ (length '(for 2 too)) (car '(21 22 23))) ;; 24
5 (cdr '(cons is short for ans))    ;; (IS SHORT FOR ANS)
6 (car (list one two))              ;; The variable ONE is unbound.
7 (car (list 'one 'two))            ;; ONE
```

1.7 Задание 7

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

Листинг 1.9 – Решение задания 7

```
1 (mystery (one two))                ;; The variable TWO is unbound.
2 (mystery one 'two))                ;; The variable ONE is unbound.
3 (mystery (last one two))           ;; The variable ONE is unbound.
4 (mystery free)                     ;; The variable FREE is unbound.
```

Листинг 1.10 – Варианты следующих выражений с устраненными ошибками

```
1 (mystery '(one two))                ;; (TWO ONE)
2 (mystery '(one 'two))                ;; ('TWO ONE)
3 (mystery (last '(one two)))          ;; (NIL TWO)
4 (mystery '(free))                    ;; (NIL FREE)
```

1.8 Задание 8

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

Формулы: $c = 5/9 * (f - 32.0)$; $f = 9/5 * c + 32.0$.

Как бы назывался роман Р. Брэдбери "+451 по Фаренгейту" в системе по Цельсию?

Листинг 1.11 – Решение задания 8

```
1 (defun f-to-c (temp) (* (/ 5 9) (- temp 32.0)))  
2 ;; (F-TO-C 451) -> 232.77779
```

Роман Р. Брэдбери назывался бы "+232.77779 по Цельсию".

1.9 Задание 9

Что получится при вычисления каждого из выражений?

В 4-ом выражении была исправлена ошибка. Изначально было так:

Листинг 1.12 – Исходное 4-ое выражение

```
1 (apply #cons "(t NIL)) ;; illegal complex number format: #CONS
```

Листинг 1.13 – Решение задания 9

```
1 (list 'cons t NIL)                ;; (CONS T NIL)  
2 (eval (list 'cons t NIL))         ;; (T)  
3 (eval (eval (list 'cons t NIL)))  ;; The function T is undefined, and its  
   name is reserved by ANSI CL.  
4 (apply #'cons '(t NIL))           ;; (T)  
5 (eval NIL)                        ;; NIL  
6 (list 'eval NIL)                  ;; (EVAL NIL)  
7 (eval (list 'eval NIL))           ;; NIL
```

1.10 Дополнительные задания

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

Листинг 1.14 – Решение дополнительного задания 1

```
1 (defun calc_cathet (hypotenuse cathet)
2   (sqrt (- (* hypotenuse hypotenuse) (* cathet cathet))))
3 ;; (CALC_CATHET 5 4) -> 3.0
```

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

Листинг 1.15 – Решение дополнительного задания 2

```
1 (defun trapezoid_area (h a b)
2   (* (/ (+ a b) 2) h))
3 ;; (TRAPEZOID_AREA 2 3 4) -> 7
```

2 Ответы на теоретические вопросы к лабораторной работе

2.1 Базис Lisp

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

Базис языка Lisp содержит:

- атомы и структуры (представляющиеся бинарными узлами);
- базовые функции и функционалы:
 - встроенные – примитивные функции (atom, eq, cons, car, cdr);
 - специальные функции и функционалы (quote, cond, lambda, eval, apply, funcall).

2.2 Классификация функций

Функции в Lisp классифицируют следующим образом:

- чистые математические функции;
- рекурсивные функции;
- специальные функции – формы (прнимают произвольное число аргументов или по разному обрабатывают аргументы);
- псевдофункции (создают эффект на внешнем устройстве);
- функции с вариативными значениями, из которых выбирается одно;
- функции высших порядков – функционалы (используются для построения синтаксически управляемых программ).

По назначению функции разделяются следующим образом:

- конструкторы — создают значение (`cons`, `list`);
- селекторы — получают доступ по адресу (`car`, `cdr`);
- предикаты — возвращают `Nil`, `T`.

2.3 Способы создания функций

Функцией называется правило, по которому каждому значению одного или нескольких аргументов ставится в соответствие конкретное значение результата.

- В Lisp можно определить функцию без имени с помощью **λ -выражений**. Lambda-определение безымянной функции:

(`lambda` <lambda-список> <форма>)

Lambda-вызов функции:

(<lambda-выражение> <формальные параметры>)

- Также в Lisp можно определить функцию с именем с помощью **`defun`**. В таких функциях `defun` связывает символьный атом с Lambda-определением:

(`defun` `f` <lambda-выражение>)

Упрощенное определение:

(`defun` `f`(`arg1`, ..., `argN`) <формы>)

2.4 Функции `Car` и `Cdr`

Функции **`car`**, **`cdr`** являются базовыми функциями доступа к данным.

- **`car`** принимает точечную пару или список в качестве аргумента и возвращает первый элемент или `Nil`, соответственно.
- **`cdr`** принимает точечную пару или список в качестве аргумента и возвращает все элементы кроме первого или `Nil`, соответственно.

2.5 Назначение и отличие в работе Cons и List

Функции **list**, **cons** являются функциями создания списков (**cons** – базовая, **list** – нет).

- **cons** создает списочную ячейку и устанавливает два указателя на аргументы.

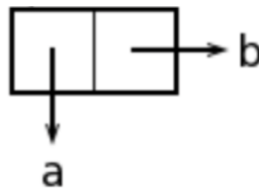


Рисунок 2.1 – Результат *cons*

- **list** принимает переменное число аргументов и возвращает список, элементы которого – переданные в функцию аргументы.

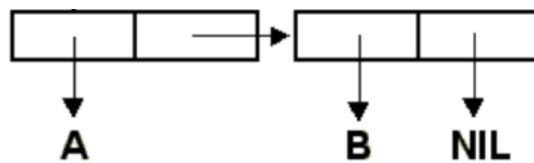


Рисунок 2.2 – Результат *list*