



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 ИУ7-65Б
(Группа)

(Подпись, дата)

Турчанинов А. М.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Толшинская Н. Б.
(И. О. Фамилия)

2024 г.

1 Практическая часть

1.1 Составить диаграмму вычисления следующих выражений

`(equal 3 (abs -3))`

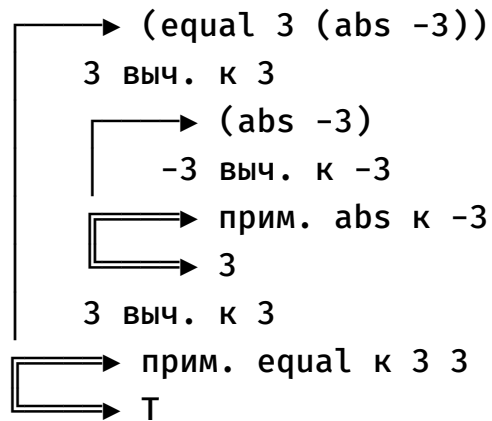


Рисунок 1.1 – Диаграмма выполнения для `(equal 3 (abs -3))`

`(equal (+ 1 2) 3)`

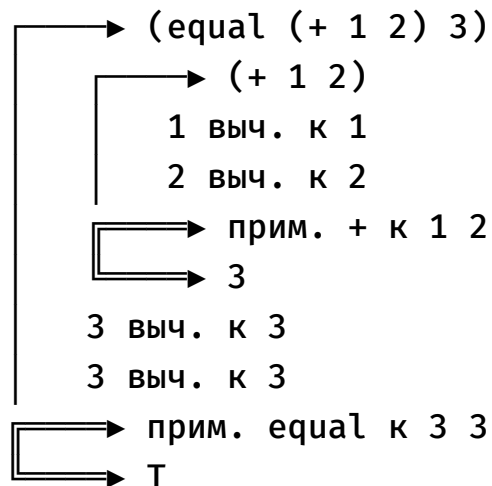


Рисунок 1.2 – Диаграмма выполнения для `(equal (+ 1 2) 3)`

(equal (* 4 7) 21)

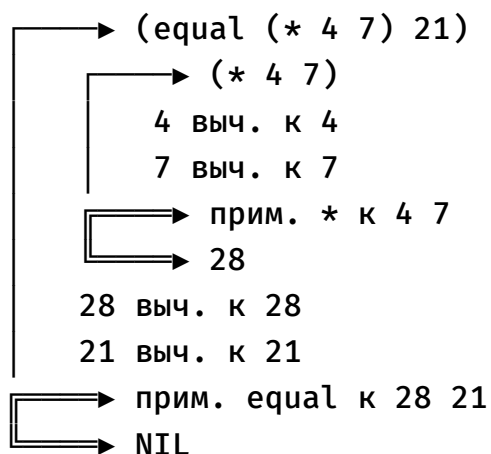


Рисунок 1.3 – Диаграмма выполнения для (equal (* 4 7) 21)

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

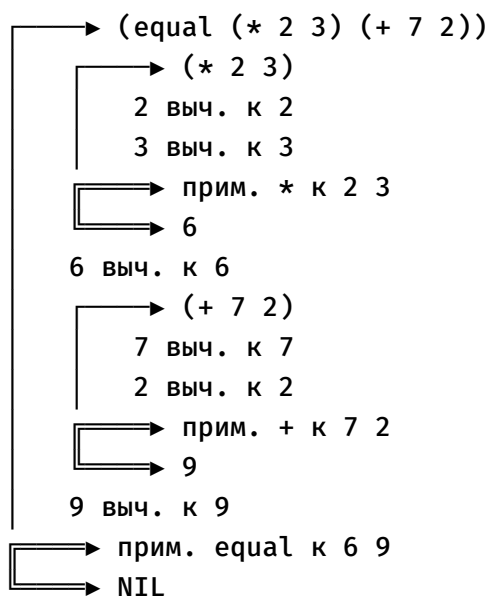


Рисунок 1.4 – Диаграмма выполнения для (equal (* 2 3) (+ 7 2))

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

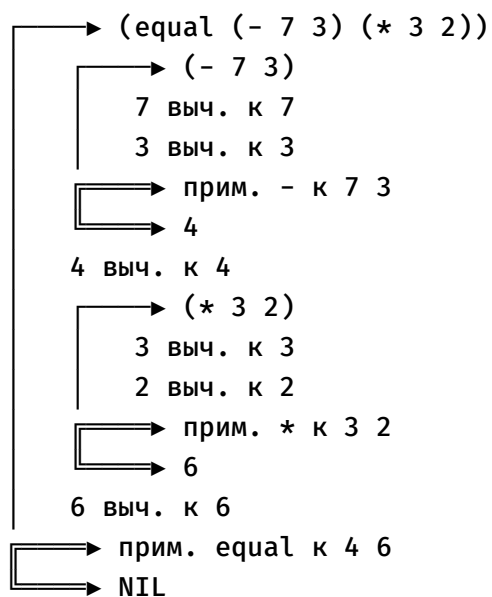


Рисунок 1.5 – Диаграмма выполнения для (equal (- 7 3) (* 3 2))

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

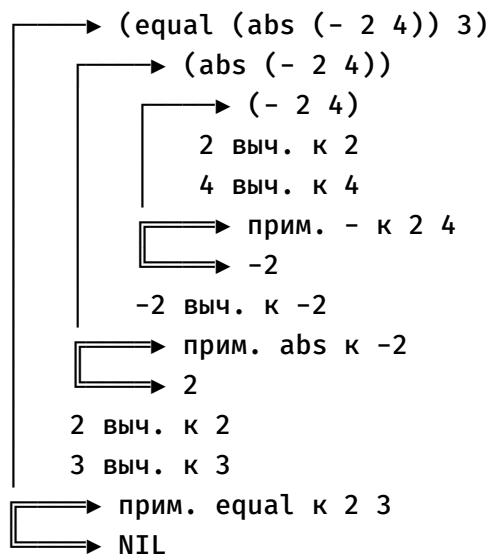


Рисунок 1.6 – Диаграмма выполнения для (equal (abs (- 2 4)) 3))

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

Листинг 1.1 – Lambda-функция, вычисляющая гипотенузу прямоугольного треугольника

```
1 (lambda (a b) (sqrt (+ (* a a) (* b b))))
```

На Рисунке 1.7 приведена диаграмма выполнения данной функции с фактическими параметрами 3 и 4.

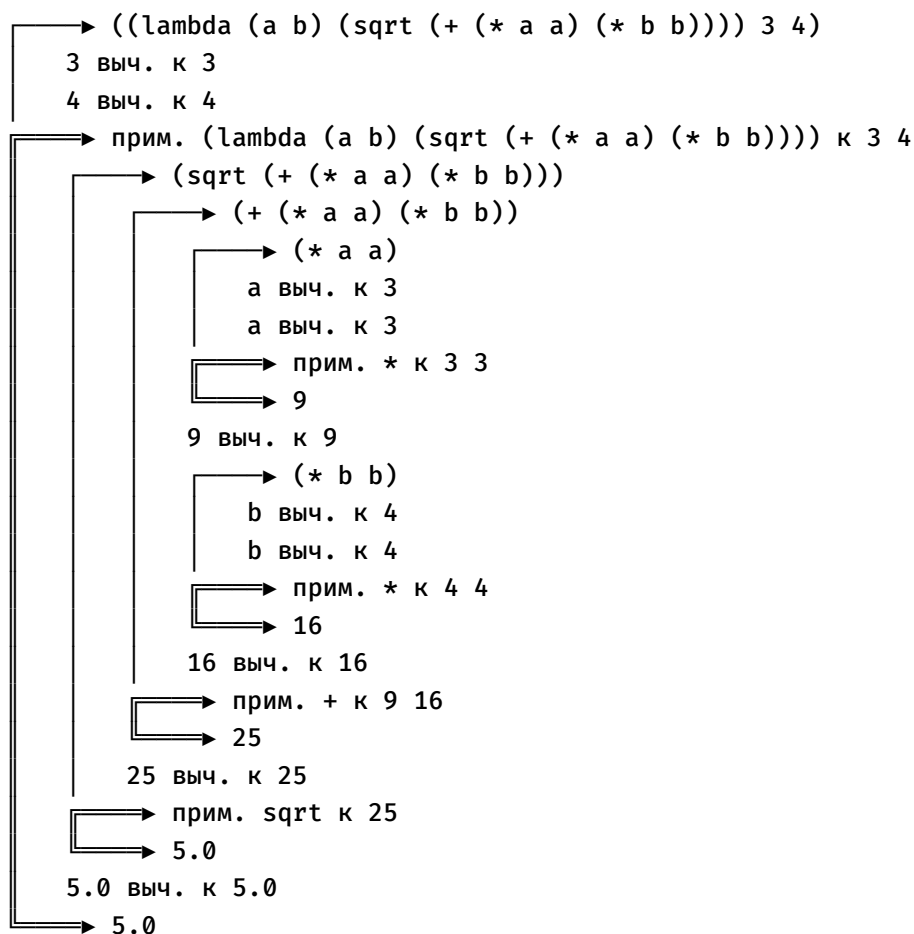


Рисунок 1.7 – Диаграмма выполнения для Листинга 1.1

1.3 Каковы результаты вычисления следующих выражений?

В Таблице 1.1 приведены результаты вычисления выражений, а так же варианты устранения возникших ошибок.

Таблица 1.1 – Результаты вычисления выражений

Выражение	Результат	Исправление
(list 'a c)	Несвязная переменная c	(list 'a 'c)
(cons 'a 'b 'c)	Неверное число аргументов: функция cons ожидает 2 аргумента, а передано 3.	(cons '(a b) 'c)
		(cons 'a '(b c))
		(cons 'b 'c)
		(cons 'a 'c)
		(cons 'a 'b)
(cons 'a (b c))	Несвязная переменная c	(cons 'a '(b c))
(list 'a (b c))	Несвязная переменная c	(list 'a '(b c))
(cons 'a '(b c))	(a b c)	Не требуется
(list a '(b c))	Несвязная переменная a	(list 'a '(b c))
(caddy (1 2 3 4 5))		(caddr '(1 2 3 4 5))
(list (+ 1 '(length '(1 2 3))))	Ошибка вычисления. Чистая математическая функция + ожидает фактический параметр NUMBER	(list (+ 1 (length '(1 2 3))))

1.4 Написать функцию от двух списков-аргументов, которая возвращает Т, если первый аргумент имеет большую длину

Листинг 1.2 – Lambda-функция для сравнения длин списков

```
1 (lambda (l1 l2) (> (length l1) (length l2)))
```

1.5 Каковы результаты вычисления следующих выражений?

В таблице 1.2 приведены результаты вычисления соответствующих выражений.

Таблица 1.2 – Результаты вычисления выражений

Выражение	Результат
(cons 3 (list 5 6))	(3 5 6)
(cons 3 '(list 5 6))	(3 list 5 6)
(list 3 'from 9 'lives (- 9 3))	(3 from 9 lives 6)
(+ (length for 2 too) (car '(21 22 23)))	Несвязная переменная for
(cdr '(cons is short for ans))	(is short for ans)
(car (list one two))	Несвязная переменная one
(car (list 'one 'two))	one

1.6 Какие результаты вычисления следующих выражений?

Дана функция

Листинг 1.3 – Lambda-функция, заданная в условии

```
1 (defun mystery (x) (list (second x) (first x)))
```

В таблице 1.3 приведены результаты вычисления соответствующих выражений.

Таблица 1.3 – Результаты вычисления выражений

Выражение	Результат
(mystery (one two))	Несвязная переменная two
(mystery one 'two))	Несвязная переменная one
(mystery (last one two))	Несвязная переменная one
(mystery free)	Несвязная переменная free

1.7 Написать функцию, которая переводит температуру в системе Фаренгейта температуру по Цельсию

Листинг 1.4 – Lambda-функция, которая переводит температуру в системе Фаренгейта температуру по Цельсию

```
1 (lambda (temp) (* 5/9 (- temp 32)))
```

1.8 Что получится при вычисления каждого из выражений?

В таблице 1.4 приведены результаты вычисления соответствующих выражений.

Таблица 1.4 – Результаты вычисления выражений

Выражение	Результат
(list 'cons t NIL)	(cons t NIL)
(eval (list 'cons t NIL))	(T)
(eval (eval (list 'cons t NIL)))	Неопределенная функция T
(apply #cons "(t NIL))	Неверный формат записи компл. числа
(eval NIL)	NIL
(list 'eval NIL)	(eval NIL)
(eval (list 'eval NIL))	NIL

2 Контрольный вопросы

2.1 Базис языка

Базис состоит из:

1. структуры, атомы;
2. встроенные (примитивные) функции (`atom`, `eq`, `cons`, `car`, `cdr`);
3. специальные функции и функционалы, управляющие обработкой структур, представляющих вычислимые выражения (`quote`, `cond`, `lambda`, `label`, `eval`).

2.2 Классификация функций

Функции в `Lisp` классифицируют следующим образом:

- чистые математические функции;
- рекурсивные функции;
- специальные функции — формы (сегодня 2 аргумента, завтра - 5);
- псевдофункции (создают эффект на внешнем устройстве);
- функции с вариативными значениями, из которых выбирается 1;
- функции высших порядков — функционал: используется для синтаксического управления программ (абстракция языка).

По назначению функции разделяются следующим образом:

- конструкторы — создают значение (`cons`, например);
- селекторы — получают доступ по адресу (`car`, `cdr`);
- предикаты — возвращают `Nil`, `T`.
- функции сравнения — такие как: `eq`, `eql`, `equal`, `equalp`.

2.3 Способы создания функций

Функции в `Lisp` можно задавать следующими способами:

Lambda-выражение

Синтаксис:

`(lambda <λ-список> форма)`

Пример:

Листинг 2.1 – Функция определенная Lambda-выражением

```
1 (lambda (a b) (sqrt (+ (* a a) (* b b))))
```

Именованная функция

Синтаксис:

`(defun <имя функции> <λ-выражение>)`

Пример:

Листинг 2.2 – Пример определения именованной функции

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

2.4 Функции `car` и `cdr`

`car` — функция получения первого элемента точечной пары.

Примеры:

S-выражение	Результат выполнения <code>car</code>
<code>(A . B)</code>	<code>A</code>
<code>((A . B) . C)</code>	<code>(A . B)</code>
<code>A</code>	ошибка

`cdr` — функция получения второго элемента точечной пары.

S-выражение	Результат выполнения <code>cdr</code>
<code>(A . B)</code>	<code>B</code>
<code>(A . (B . C))</code>	<code>(B . C)</code>
<code>A</code>	ошибка

2.5 Назначение и отличие `list` от `cons`

`cons` — функция конструирования точечной пары, на вход получает 2 значения и делает из них точечную пару.

`list` — функция конструирования списка. На вход получает произвольное количество элементов и делает из них список.

Вызовы `(list 1 2 3 4)` и `(cons 1 (cons 2 (cons 3 (cons 4 Nil))))` эквивалентны, то есть дают одинаковый результат.