



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 к лабораторной работе №4
по курсу: «Функциональное и логическое
программирование»
Тема: **Определение функций пользователя**

Студент группы ИУ7-62Б

А.П. Сорокин

(И.О. Фамилия)

Преподаватель

Н.Б. Толпинская

(И.О. Фамилия)

2020 г.

СОДЕРЖАНИЕ

1	Цели и задачи	2
2	Теоретическая часть	2
3	Практическая часть	4
3.1	Задание 7	4
3.2	Задание 8	4
3.3	Задание 9	5
3.4	Задание 10	6

1 Цели и задачи

Цель работы: приобрести навыки создания и использования функций пользователя в Lisp.

Задачи работы: изучить способы создания и использования именованных и неименованных функций пользователя для обработки списков.

2 Теоретическая часть

Базис - это минимально необходимый набор конструкций, с помощью которого можно реализовать задачу.

Классификация функций:

- чистые (математические): принимают строго определённое число аргументов и возвращают одно значение
- формы: могут принимать разное число аргументов, в зависимости от чего по-разному себя ведёт
- функционалы: принимают функциональные описания.

Классификация базовых функций Lisp:

- функции-селекторы (функции доступа к данным): car, cdr
- функции-конструкторы: cons
- функции-предикаты: atom, Null, lisp и т. д.
- функции-сравнения: eq, eql, =, equal, equalp

Списки в оперативной памяти представляются с помощью списковых ячеек. В списковой ячейке хранится два указателя: один указывает на голову (car), другой - на хвост (cdr).

Функции CAR и CDR являются базовыми функциями-селекторами. **CAR** принимает в качестве аргумента точечную пару или пустой список. В первом случае функция вернёт первый элемент, во втором - nil. **CDR** также принимает в качестве аргумента точечную пару или пустой список. Функция возвращает список, состоящий из всех элементов списка-аргумента,

кроме первого. Если в списке меньше двух элементов, то функция возвращает `nil`.

Функции LIST и CONS являются функциями-конструкторами, причём функция **CONS** является базовой, а **LIST** - нет.

CONS создает списочную ячейку и устанавливает два указателя на принимаемые два аргумента.

LIST принимает переменное число аргументов и возвращает список, элементами которого являются аргументы функции.

Пример использования функций **CONS** и **LIST** для создания списка (a b c):

Листинг 1 – Создание списка (a b c)

```
1      (cons 'a (cons 'b (cons 'c nil)))  
2      (list 'a 'b 'c)
```

3 Практическая часть

3.1 Задание 7

Функция, которая переводит температуру в системе Фаренгейта в температуру по Цельсию:

Листинг 2 – Функция перевода температуры

```
1 (defun f-to-c (temp) (/ (* 5 (- temp 32)) 9.0))
```

Примеры работы:

Листинг 3 – Примеры работы функции f-to-c

```
1 (f-to-c 451)
2 > 72.77778
3 (f-to-c 320)
4 > 0.0
5 (f-to-c 220)
6 > -55.555557
```

3.2 Задание 8

Результаты вычисления выражений представлены в таблице 1.

Таблица 1 – Выражения и результаты их вычислений задания 8

№	Выражение	Результат
1	(list 'cons t NIL)	(cons T NIL)
2	(eval (list 'cons t NIL))	T
3	(eval (eval (list 'cons t NIL)))	function T is undefined
4	(apply #'cons '(t NIL))	(T)
5	(eval NIL)	NIL
6	(list 'eval NIL)	(eval NIL)
7	(eval (list 'eval NIL))	NIL

3.3 Задание 9

Функция, вычисляющая катет по заданной гипотенузе и другому катету прямоугольного треугольника:

Листинг 4 – Функция вычисления гипотенузы

```
1 (defun cath (hyp leg) (sqrt (- (* hyp hyp) (* leg leg))))
```

Диаграмма вычисления функции:

→ (cath c b)

- вычисление c к c
- вычисление b к b

⇒ применение cath к c, b

- создание переменной hyp со значением c
- создание переменной leg со значением b

→ (sqrt (- (* hyp hyp) (* leg leg)))

→ (- (* hyp hyp) (* leg leg))

→ (* hyp hyp)

- вычисление hyp к c
- вычисление hyp к c

⇒ применение * к c, c

⇒ c^2

→ (* leg leg)

- вычисление leg к b
- вычисление leg к b

⇒ применение * к b, b

⇒ b^2

⇒ применение - к c^2 , b^2

⇒ $c^2 - b^2$

⇒ применение sqrt к $c^2 - b^2$

⇒ $\sqrt{c^2 - b^2}$

⇒ $\sqrt{c^2 - b^2}$

3.4 Задание 10

Функция, вычисляющая площадь трапеции по ее основаниям и высоте:

Листинг 5 – Функция вычисления объёма параллелепипеда

```
1 (defun sq_trapez (base1 base2 height) (/ (* (+ base1 base2) height)
      2.0))
```

Диаграмма вычисления функции:

→ (sq_trapez a b h)

- вычисление a к b
- вычисление b к b
- вычисление h к h

⇒ применение sq_trapez к a, b, h

- создание переменной base1 со значением a
- создание переменной base2 со значением b
- создание переменной height со значением h

→ (/ (* (+ base1 base2) height) 2.0))

→ (* (+ base1 base2) height)

→ (+ base1 base2)

- вычисление base1 к a
- вычисление base2 к b

⇒ применение + к a, b

⇒ $a + b$

- вычисление height к h

⇒ применение * к $a + b$, h

⇒ $(a + b) \cdot h$

- вычисление 2.0 к 2.0

⇒ применение / к $(a + b) \cdot h$, 2

⇒ $\frac{(a + b)}{2} \cdot h$

⇒ $\frac{(a + b)}{2} \cdot h$