

**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 дисциплине «Функциональное и логическое
программирование»**

Тема Определение функций пользователя.

Студент Калашников С.Д.

Группа ИУ7-63Б

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

Москва, 2023

СОДЕРЖАНИЕ

1	Теоретическая часть	3
1.1	Базис языка	3
1.2	Классификация функций	3
1.3	Способы создания функций	4
2	Практическая часть	5
2.1	Задание 1 Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента. . .	5
2.2	Задание 2 Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.	5
2.3	Задание 3 Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию.	5
2.4	Задание 4 Написать функцию, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим.	6

1 Теоретическая часть

1.1 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).

1.2 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.

1.3 Способы создания функций

Функции в Lisp можно задавать следующими способами:

Lambda-выражение

Синтаксис:

(lambda <λ-список> форма)

Пример:

Листинг 1.1 — Функция определенная Lambda-выражением

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

Именованная функция

Синтаксис:

(defun <имя функции> <λ-выражение>)

Пример:

Листинг 1.2 — Пример определения именованной функции

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

2 Практическая часть

2.1 Задание 1 Написать функцию, которая принимает целое число и возвращает первое четное число, не меньшее аргумента.

Листинг 2.1 — Выражение 1

```
1 (defun t1(x) (if(eq 0 (mod x 2)) x (+ x 1)))
```

2.2 Задание 2 Написать функцию, которая принимает число и возвращает число того же знака, но с модулем на 1 больше модуля аргумента.

Листинг 2.2 — Выражение 2

```
1 (defun t2(x) (if (> x 0) (+ x 1) (- x 1)))
```

2.3 Задание 3 Написать функцию, которая принимает два числа и возвращает список из этих чисел, расположенный по возрастанию.

Листинг 2.3 — Выражение 3

```
1 (defun t3(x y) (if (< x y) (list x y) (list y x)))
```

2.4 Задание 4 Написать функцию, которая принимает три числа и возвращает Т только тогда, когда первое число расположено между вторым и третьим.

Листинг 2.4 — Выражение 4

```
1 (defun t4(x y z) (and (< y x) (< x z)))
```