|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.03 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 8 |

**Дисциплина:** Функциональное и логическое программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-62Б |  |  | Н.А. Гарасев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Н.Б.Толпинская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

**Задание 1. Умножение списка чисел и списка разных объектов на число.**

; Умножение, только если все числа.

**(defun** mulla **(**lst k**)**

**(and** **(**every #'numberp lst**)**

**(mapcar** **(lambda** **(**a**)** **(\*** a k**))** lst**)))**

; Умножить все числа.

**(defun** mullb **(**lst k**)**

**(mapcar** **(lambda** **(**a**)**

**(cond** **((listp** a**)** **(**mullb a k**))**

**((numberp** a**)** **(\*** a k**))**

**(**T a**)))** lst**))**

\* (mulla `(1 2 3 4 5 6) 5)

(5 10 15 20 25 30)

\* (mullb `(((1 ((2) ((3)))) 4 5) 6) 5)

(((5 ((10) ((15)))) 20 25) 30)

**Задание 2. Select-between с сортировкой**

; select-between

**(defun** select-between **(**lst a b**)**

**(**reduce **(lambda** **(**res tmp**)**

**(cond** **((and** **(>** tmp a**)** **(<** tmp b**))** **(append** res **(list** tmp**)))**

**(**T res**)))**

lst **:initial-value** Nil**))**

; sort

**(defun** my\_m **(**lst m**)**

**(cond** **((null** lst**)** m**)**

**((<(car** lst**)** m**)** **(**my\_m **(cdr** lst**)** **(car** lst**)))**

**(**T **(**my\_m **(cdr** lst**)** m**))))**

**(defun** my\_min **(**lst**)**

**(cond** **((null** lst**)** Nil**)**

**(**T **(**my\_m **(cdr** lst**)** **(car** lst**)))))**

**(defun** my\_rem **(**el lst**)**

**(cond** **((null** lst**)** Nil**)**

**(**T **(**reduce **(lambda** **(**res tmp**)**

**(cond** **((=** el tmp**)** res**)**

**(**T **(append** res **(list** tmp**)))))**

lst **:initial-value** Nil**))))**

**(defun** my\_r **(**el lst res**)**

**(cond** **((null** lst**)** res**)**

**((=** el **(car** lst**))** **(append** res **(cdr** lst**)))**

**(**T **(**my\_r el **(cdr** lst**)** **(append** res **(list** **(car** lst**)))))))**

**(defun** my\_rem\_v2 **(**el lst**)**

**(cond** **((null** lst**)** Nil**)**

**(**T **(**my\_r el lst Nil**))))**

**(defun** my\_s **(**lst res**)**

**(cond** **((null** lst**)** res**)**

**(**T **(**my\_s **(remove** **(**my\_min lst**)** lst**)** **(append** res **(list** **(**my\_min lst**)))))))**

**(defun** my\_sort **(**lst**)**

**(cond** **((null** lst**)** Nil**)**

**(**T **(**my\_s lst Nil**))))**

\* (my\_sort (select-between `(5 4 7 8 1 2 3 6 5 4 8 9 6 5 7 4 1 2 3 5 6) 2 8))

(3 3 4 4 4 5 5 5 5 6 6 6 7 7)

**Задание 3. Уменьшить на 10.**

; Все числа на 10 уменьшить.

**(defun** decr10 **(**lst**)**

**(mapcar** **(lambda** **(**a**)** **(-** a 10**))** lst**))**

\* (decr10 `(1 2 3 4 5 6 7 8 9))

(-9 -8 -7 -6 -5 -4 -3 -2 -1)

**Задание 4. Поиск первого не пустого списка.**

; Поиск первого не пустого списка.

**(defun** find-list **(**lst**)**

**(**find-if **(lambda** **(**a**)** **(and** **(listp** a**)** **(not(null** a**))))** lst**))**

\* (find-list `(1 () (2 () 3) 4 5 6 7 8 9))

(2 NIL 3)

\* (find-list `(1 (()) (2 () 3) 4 5 6 7 8 9))

(NIL)

**Дополнительное задание. Сумма чисел сложной структуры.**

; sum

**(defun** summa **(**lst**)** **(**summa-inner lst 0**))**

**(defun** summa-inner **(**lst res**)**

**(**reduce **(lambda** **(**res a**)**

**(cond** **((listp** a**)** **(**summa-inner a res**))**

**((numberp** a**)** **(+** res a**))**

**(**T res**)))**

lst **:initial-value** res**))**

\* (summa `(1 (((2))3) 4 (5) 6))

21

**Вопросы.**

**Вопрос 1. Порядок работы и варианты использования функционалов**

Функционалы делятся на применяющие и отображающие. Применяющие функционалы нужны для того, чтобы вызвать функцию сразу для нескольких аргументов или для списка аргументов. Отображающие служат для упрощения решения более сложных задач.