|  |  |
| --- | --- |
| **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 «Программное обеспечение ЭВМ и информационные технологии»

**ОТЧЕТ**

*к лабораторной работе №6*

*По курсу: «Функциональное и логическое программирование»*

Студентка ИУ7-65Б

Оберган Т.М

Преподаватель

Толпинская Н.Б

*2020 г.*

**1. Дано два списка: название стран (4 шт) и список столиц. Создать**

**а) список из двухэлементных списков**

**б) список точечных пар**

**по созданным спискам:**

**1 - по столице найти страну**

**2 - по стране найти столицу**

(setq c '(usa russia canada gb))

(setq s '(washington moscow ottawa london))

**Формирование**:

(defun fa (lst1 lst2 &optional res)

(if (or (null lst1) (null lst2))

res

(fa (cdr lst1) (cdr lst2) (cons (list (car lst1) (car lst2)) res))

))

(print (fa c s)) ; ((GB LONDON) (CANADA OTTAWA) (RUSSIA MOSCOW) (USA WASHINGTON))

Список точечных пар получается путем замены функции list на cons.

(defun fb (lst1 lst2 &optional res)

(if (or (null lst1) (null lst2))

res

(fb (cdr lst1) (cdr lst2) (cons (cons (car lst1) (car lst2)) res))

))

(print (fb c s)) ;((GB . LONDON) (CANADA . OTTAWA) (RUSSIA . MOSCOW) (USA . WASHINGTON))

**Поиск по ключу в списке списков:**

(defun fa1 (capital lst)

(cond

((null lst) nil)

((eq capital (cadar lst)) (caar lst))

(T (fa1 capital (cdr lst)))

)

)

;(print (fa1 'moscow (fa c s))) ; russia

(defun fa2 (country lst)

(cond

((null lst) nil)

((eq country (caar lst)) (cadar lst))

(T (fa2 country (cdr lst)))

)

)

;(print (fa2 'russia (fa c s))) ; moscow

**Поиск по ключу в списке точечных пар:**

(defun fb1 (capital lst)

(cond

((null lst) nil)

((eq capital (cdar lst)) (caar lst))

(T (fb1 capital (cdr lst)))

)

)

;(print (fb1 'moscow (fb c s))) ; russia

(defun fb2 (country lst)

(cond

((null lst) nil)

((eq country (caar lst)) (cdar lst))

(T (fb2 country (cdr lst)))

)

)

;(print (fb2 'russia (fb c s))) ; moscow

**2. Переписать функцию how-alike, приведенную в лекции и использующую COND, используя конструкции IF, AND/OR.**

**Приведенная в лекции:**

(defun how\_alike (x y)

(cond

((or (= x y) (equal x y)) 'the\_same)

((and (oddp x) (oddp y)) 'both\_odd)

((and (evenp x) (evenp y)) 'both\_even)

(t 'different)))

**Моя реализация используя if:**

(defun my\_how\_alike (x y)

(if (or (= x y) (equal x y)) 'the\_same

(if (and (oddp x) (oddp y)) 'both\_odd

(if (and (evenp x) (evenp y)) 'both\_even

'different))))