|  |  |
| --- | --- |
| 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»\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 6**

|  |  |
| --- | --- |
| **Дисциплина** Функциональное и логическое программирование  **Тема** \_Рекурсивные функции\_  **Студент** \_Ильясов И. М.\_  **Группа** \_ИУ7-63Б\_  **Оценка (баллы)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Преподаватель** \_Толпинская Н. Б.\_ |  |

Москва, 2020 г.

**Задание.** Есть два списка с фамилиями работников и их зарплатами соответственно. Позиции фамилий работников из первого списка соответствуют позициям их зарплат из второго списка (Ivanov => 50000, Stepanov => 100000).

Необходимо получить результирующий список из двух списков в формате «фамилия-зарплата»:

* при помощи двухэлементных списков;
* при помощи точечных пар;

Два списка с фамилиями работников и их зарплатами:

(defvar workers)

(defvar salary)

(setq workers (Stepanov Ivanov Gorbunov))

(setq salary (100000 50000 80000))

**Способ №1**: с использованием двухэлементных списков.

*Функция создания результирующего списка, состоящего из двухэлементных списков:*

(defun merge\_list (workers salary)

(mapcar #'list workers salary)

)

**Способ №2**: с использованием точечных пар.

*Функция создания результирующего списка, состоящего из точечных пар:*

(defun merge\_cons (workers salary)

(mapcar #'cons workers salary)

)

*Изменение зарплат всех работников с использованием функционалов:*

(defun multiply (el)

(list (car el) (\* (cadr el) 2))

)

(defun task (lst)

(mapcar #'func lst)

)

*Изменение зарплат работников, у которых зарплата меньше 60000 с использованием функционалов:*

(defun task\_if (lst)

(if (< (cadr lst) 60000)

(func lst)

lst

)

)

(print (mapcar #'task\_if lst1))

**Задание.**

* (defun how\_alike (x y)

(cond

((or (= x y) (equal x y)) ‘same)

((and (oddp x) (oddp y)) ‘odd)

((and (evenp x) (evenp y) ‘even)

(T ‘difference)

)

)

* (defun how\_alike\_if (x y)

(if (or (= x y) (equal x y))

‘same

(if (and (oddp x) (oddp y))

‘odd

(if (and (evenp x) (evenp y))

‘even

‘difference

)

)

)

)

* (defun how\_alike\_or (x y)

(or

(and (or (= x y) (equal x y)) ‘same)

(and (and (oddp x) (oddp y)) ‘odd)

(and (and (evenp x) (evenp y)) ‘even)

)

)