ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент |  |  |  | Степанов П.А. |
| должность, уч.степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4 |
| Использование функционалов в языке LISP |
| по курсу: ФУНКЦИОНАЛЬНОЕ И ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4136 |  | 09.06.2023 |  | Н. С. Бобрович |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

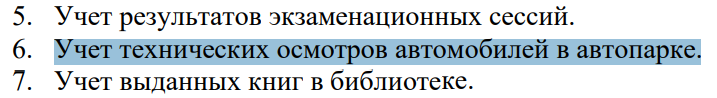
**Цель:** Ознакомиться с применением функционалов в языке Lisp.

**Указания:** Используя связывание свойств с символами, структуры, последовательности и функционалы реализовать программу для предложенного ниже варианта, выполняющую:

1. включение/исключение/замену данных;

2. 4-5 поисковых запроса (разработать самостоятельно и согласовать с преподавателем). Формальным требованием является использование последовательностей и функционалов при выполнении задания

**Задание:**

****

**Ход работы.**

**Листинг:**

(require 'cl-lib)

(defstruct

my-struct

(id nil)

(name nil)

(value nil)

(year nil)

)

(defun msc (id name value year)

(let

((s (make-my-struct)))

(setf (my-struct-id s) id)

(setf (my-struct-name s) name)

(setf (my-struct-value s) value)

(setf (my-struct-year s) year)

s

)

)

(defun replace-data (seq id new-value)

(mapcar

(lambda (elem)

(if (equal (get-value-by-field-name-from-my-struct elem 'id) id)

(setf (my-struct-value elem) new-value))

elem)

seq)

)

(defun delete-data (seq id)

(remove-if

(lambda (elem)

(equal (get-value-by-field-name-from-my-struct elem 'id) id))

seq)

)

(defun add-data (seq new-value)

(cons new-value seq)

)

(defun get-value-by-field-name-from-my-struct (elem field-name)

(cond

((equal field-name 'id) (my-struct-id elem))

((equal field-name 'name) (my-struct-name elem))

((equal field-name 'value) (my-struct-value elem))

((equal field-name 'year) (my-struct-year elem))

(t nil)

)

)

(defun find-in-my-struct (seq field-name pattern)

(remove-if-not

(lambda (elem)

(equal (get-value-by-field-name-from-my-struct elem field-name) pattern))

seq)

)

(defun find-bad-cars (seq &optional value)

(setq value (or value 60))

(remove-if

(lambda (elem)

(< (get-value-by-field-name-from-my-struct elem 'value) value))

seq)

)

(defun find-cars-beetwen (seq min-val max-val)

(remove-if-not

(lambda (elem)

(and (<= (get-value-by-field-name-from-my-struct elem 'value) max-val)

(>= (get-value-by-field-name-from-my-struct elem 'value) min-val)))

seq)

)

(defun map-seq (seq fn)

(mapcar fn seq)

)

(setq my-sequence (list (msc 1 "audi" 10 2022)

(msc 2 "audi" 20 2023)

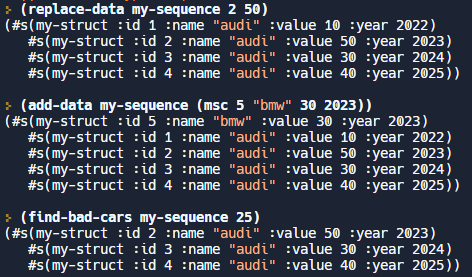
(msc 3 "audi" 30 2024)

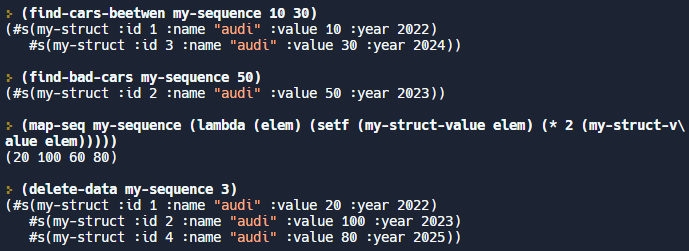
(msc 4 "audi" 40 2025)

)

)

**Output:**

****

****

**Вывод:** Ознакомился с применением функционалов в языке Lisp.