МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ:

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

«ИСПОЛЬЗОВАНИЕ ФУНКЦИОНАЛОВ В ЯЗЫКЕ LISP»

ПО ДИСЦИПЛИНЕ «ФУНКЦИОНАЛЬНОЕ И ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ»

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4631 |  |  |  | Гришин.С.А |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2019

**Задание**

Цель работы. Ознакомиться с применением функционалов в языке Lisp

9. Маршруты городского транспорта

**Решение**

(DEFSTRUCT route number type start finish)

(SETQ seq (MAKE-SEQUENCE 'VECTOR 2))

(SETF (ELT SEQ 0) (MAKE-route :number "60" :type "Tramway" :start "Marshala24" :finish "Marshala35"))

(SETF (ELT SEQ 1) (MAKE-route :number "61" :type "Tramway" :start "Marshala35" :finish "Marshala45"))

(defun menu()

(format T "~%1 - Print~%2 - Add~%3 - Delete~%4 - Edit~%5 - Find by number of route~%6 - Find by type~%7 - Find by start~%8 - Find by finish ~%"))

(defun printel (el) (format T "~%~A~10T~A~40T~A~50T~A"

(route-number el)

(route-type el)

(route-start el)

(route-finish el)

) el)

(defun add(seq num rou arr dep)(concatenate 'vector seq (MAKE-SEQUENCE 'VECTOR 1 :initial-element (MAKE-route :number (write-to-string num) :type (write-to-string rou) :start (write-to-string arr) :finish (write-to-string dep)))))

(defun checknum(el num) (

cond

((string= (write-to-string num) (route-number el)) (printel el))

(T NIL)))

(defun checktype(el type) (

cond

((search (write-to-string type) (route-type el)) (printel el))

(T NIL)))

(defun checkarrtime(el arr) (

cond

((search (write-to-string arr) (route-start el)) (printel el))

(T NIL)))

(defun checkdeptime(el dep) (

cond

((search (write-to-string dep) (route-finish el)) (printel el))

(T NIL)))

(defun findbynumber(seq number)(map 'vector 'checknum seq (make-list (length seq) :initial-element number)) seq)

(defun findbytype(seq type)(map 'vector 'checktype seq (make-list (length seq) :initial-element type)) seq)

(defun findbyarrtime(seq arr)(map 'vector 'checkarrtime seq (make-list (length seq) :initial-element arr)) seq)

(defun findbydeptime(seq dep)(map 'vector 'checkdeptime seq (make-list (length seq) :initial-element dep)) seq)

(defun findindbynumber(seq number &optional(ind 0))(

cond

((string= (write-to-string number) (route-number (elt seq ind))) ind)

((= ind (length seq)) NIL)

(T (findindbynumber seq number (+ 1 ind)))

))

(defun edit(seq numroute numfield newdata)(

cond

((= numfield 1)(setf (route-number (elt seq (findindbynumber seq numroute))) newdata))

((= numfield 2)(setf (route-type (elt seq (findindbynumber seq numroute))) newdata))

((= numfield 3)(setf (route-start (elt seq (findindbynumber seq numroute))) newdata))

((= numfield 4)(setf (route-finish (elt seq (findindbynumber seq numroute))) newdata)))

seq)

(defun action(choise seq)(

cond

((= 1 choise) (map 'vector 'printel seq))

((= 2 choise) (add seq (read) (read) (read) (read)))

((= 3 choise) (remove (elt seq (findindbynumber seq (read))) seq))

((= 4 choise) (format T "number of route, number of field, new data")(edit seq (read) (read) (read)))

((= 5 choise) (findbynumber seq (read)))

((= 6 choise) (findbytype seq (read)))

((= 7 choise) (findbyarrtime seq (read)))

((= 8 choise) (findbydeptime seq (read)))

((= 0 choise) (quit))

(T (format T "123"))

))

(defun main(seq)

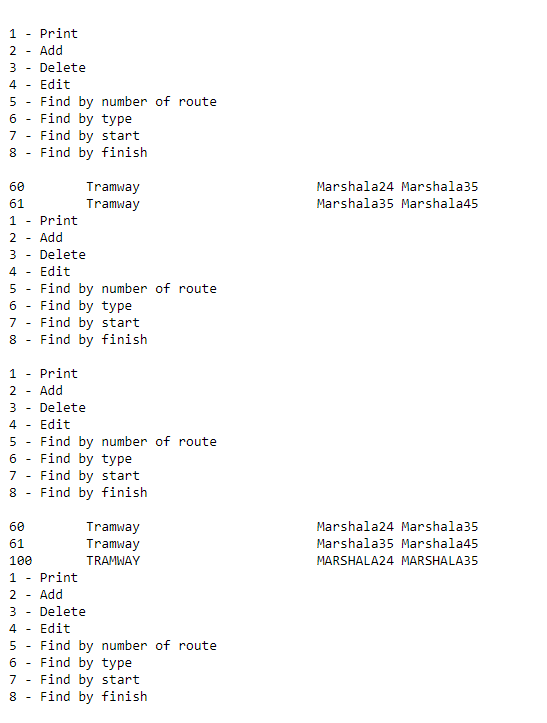
(menu)

(main (action (read) seq))

)

(main seq)

**Результат**

****

**Вывод**

В ходе лабораторной работы были изучены функционалы LISP для выполнения в поиске.