

Министерство науки и высшего образования Российской Федерации  
Севастопольский государственный университет  
Кафедра ИС

Отчет  
по лабораторной работе №4  
«ЕЯ доступ к базе данных на основе алгоритма сопоставления с образцом»  
по дисциплине  
«МЕТОДЫ И СРЕДСТВА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

Выполнил студент группы ИС/б-17-2-о  
Горбенко К. Н.  
Проверил  
Сметанина Т.И.

Севастополь  
2020

# 1 ЦЕЛЬ РАБОТЫ

Исследование алгоритма сопоставления с образцом и особенностей его применения для формирования запросов к базам данных, а также для организации доступа к базам данных на ограниченном подмножестве естественного языка.

## 2 ПОСТАНОВКА ЗАДАЧИ

Для базы данных, созданной в лабораторной работе 3, необходимо написать на языке Лисп интерфейс, который позволяет выполнять ЕЯ-запросы с помощью алгоритма сопоставления с образцом. Ответ на запрос должен также представляться на естественном языке в виде списка слов предложения. Кроме запроса, заданного по варианту задания, предусмотреть 5-6 различных дополнительных запросов.

## 3 ХОД РАБОТЫ

Код программы:

```

1  (defun main ()
2  (query '(Загрузить базу))
3
4  (format t "Добавить маршрут из Москва в Краснодар с номером 418~%")
5  (query '(Добавить маршрут из "Москва" в "Краснодар" с номером "418"))
6  (format t "Добавить маршрут из Краснодар в Симферополь с номером 8~%~%")
7  (query '(Добавить маршрут из "Краснодар" в "Симферополь" с номером "8"))
8
9  (format t "Какие маршруты идут в город Симферополь~%")
10 (selectQuery '(Какие маршруты идут в город "Симферополь"))
11
12 (format t "Найти маршрут номер 8~%")
13 (selectQuery '(Найти маршрут номер "8"))
14
15 (format t "Какой маршрут идет в Краснодар~%")
16 (selectQuery '(Какой маршрут идет в "Краснодар"))
17
18 (format t "Показать все маршруты")
19 (selectQuery '(Показать все маршруты))
20
21 (format t "Изменить маршруты с конечной станцией Симферополь на Омск~%")
22 (query '(Изменить маршруты с конечной станцией "Симферополь" на "Омск"))
23
24 (format t "Изменить начальный маршрут у 1 номера на город Красноярск~%~%")

```

```

25 (query '(Изменить начальный маршрут у "1" номера на город "Красноярск"))
26
27 (format t "Показать все маршруты")
28 (selectQuery '(Показать все маршруты))
29
30 (format t "Выбрать маршруты с номерами от 1 до 9~%")
31 (selectQuery '(Выбрать маршруты с номерами от "1" до "9"))
32 )
33
34 (defvar *db* nil)
35
36 (defun insert (start end number)
37   (push (list :start start :end end :number number) *db*)
38 )
39
40 (defun savef (filename)
41   (with-open-file (out filename :direction :output :if-exists :supersede)
42     (with-standard-io-syntax
43       (print *db* out)
44     )
45   )
46 )
47
48 (defun loadf (filename)
49   (with-open-file (in filename)
50     (with-standard-io-syntax
51       (setf *db* (read in))
52     )
53   )
54 )
55
56 (defun select* ()
57   (format t "~%")
58   (format t "~%~{~{~a:~a~%~}~%~}" *db*)
59 )
60
61 (defun where(&key start end number)
62   #'(lambda (row)
63     (and
64       (if start (equal (getf row :start) start) t)
65       (if end (equal (getf row :end) end) t)
66       (if number (equal (getf row :number) number) t)
67     )
68   )
69 )
70
71 (defun update (where-func &key start end number)
72   (setf *db*

```

```

73      (mapcar
74        #'(lambda (row)
75          (when (funcall where-func row)
76            (if start (setf (getf row :start) start))
77            (if end (setf (getf row :end) end))
78            (if number (setf (getf row :number) number))
79            )
80          row
81        )
82      *db*
83    )
84  )
85 )
86
87 (defun selectByRangeNumber (startNumber endNumber)
88   (remove-if-not #'(lambda (row) (and (> (parse-integer (getf row :number)) (
      parse-integer startNumber)) (< (parse-integer (getf row :number)) (parse-
      integer endNumber)))) *db*)
89 )
90
91 (defun match (p d)
92   (cond
93     ;; правило 1
94     ((and (null p) (null d)) t)
95
96     ;; правило 2
97     ((and (null d)
98           (eq (car p) '$)
99           (null (cdr p))) t)
100
101     ;; один из списков исчерпан
102     ((or (null p) (null d)) nil)
103
104     ;; правило 3 и правило 4
105     ((or (equal (car p) '?)
106          (equal (car p) (car d)))
107      (match (cdr p) (cdr d)))
108
109     ;; правило 5 и 6
110     ((eq (car p) '$)
111      (cond ((match (cdr p) d) t)
112            ((match p (cdr d)) t)))
113
114     ;; правило 7 - сопоставление списков, включающих подписки
115     ((and (not (atom (car p)))
116           (not (atom (car d)))
117           (match (car p) (car d)))
118      (match (cdr p) (cdr d)) )

```

```

119
120 ;; правило 8 - подстановка значения в переменную
121 ((and (atom (car p))
122       (eq (car-letter (car p)) #\?)
123       (match (cdr p)(cdr d)))
124      (set (cdr-name (car p)) (car d)) t)
125
126 ;; правило 9 - подстановка сегмента значений в переменную
127 ((and (atom (car p))
128       (eq (car-letter (car p)) #\$))
129      (cond ((match (cdr p)(cdr d))
130            (set (cdr-name (car p)) (list (car d)))
131            t)
132            ((match p (cdr d))
133             (set (cdr-name (car p))
134                 (cons (car d)(eval (cdr-name (car p)))))
135             t)))
136
137 ;; правило 10 - обработка пакета ограничений, если в пакете есть« ?
138 ((and (not(atom (car p)))
139       (eq (caar p) 'restrict)
140       (eq (cadar p) '?))
141      (and-to-list
142       (mapcar #'(lambda (pred)
143                   (funcall pred (car d))) (cddar p))))
144      (match (cdr p)(cdr d)))
145
146 ;; правило 11 - обработка пакета ограничений, если в пакете есть« ?»V
147 ;; например: (match '((restrict ?V integerp evenp) b c) '(36 b c))
148 ((and (not (atom (car p)))
149       (not (atom d))
150       (eq (caar p) 'restrict)
151       (eq (car-letter (cadar p)) #\?)
152       (and-to-list
153        (mapcar #'(lambda (pred)
154                    (funcall pred (car d))) (cddar p))))
155       (match (cdr p)(cdr d)))
156      (set (cdr-name (cadar p)) (car d))
157      t)
158 ))
159
160 (defun car-letter (x) (if (not (numberp x)) (car (coerce (string x) 'list))))
161
162 (defun cdr-name (x)
163   (intern (coerce (cdr (coerce (string x) 'list)) 'string)))
164 )
165
166 (defun and-to-list (lis)

```

```

167 (let ((res t))
168   (dolist (temp lis res)
169     (setq res (and res temp)))
170   )
171 )
172
173 (defun get-matches (p database)
174   (remove-if-not #'(lambda (record) (match p record)) database)
175   )
176
177 (defun query (q)
178   (cond
179     ((match '($ загрузить $) q)
180      (loadf "database.txt"))
181     ((match '($ сохранить $) q)
182      (savef "database.txt"))
183     ((match '(Добавить $ из ?start в $ ?end $ с номером $ ?number) q)
184      (insert start end number))
185     ((match '($ в город $ ?end) q)
186      (setf temp (get-matches '($ :end ,end $) *db*))
187      (if (null temp) "Маршрутов туда нет" temp))
188     ((match '($ номер $ ?number) q)
189      (setf temp (get-matches '($ :number ,number $) *db*))
190      (if (null temp) "Маршрутов с таким номером нет" temp))
191     ((match '($ идет в $ ?end) q)
192      (setf temp (get-matches '($ :end ,end $) *db*))
193      (if (null temp) "Отсутствуют маршруты с таким пунктом назначения" temp))
194     ((match '($ все $) q)
195      (select*))
196     ((match '(Изменить $ конечной станцией ?end на ?endNew) q)
197      (update (where :end end) :end endNew))
198     ((match '(Изменить начальный маршрут у ?number номера на город ?start) q)
199      (update (where :number number) :start start))
200     ((match '(Выбрать $ с номерами от ?startNumber до ?endNumber) q)
201      (selectByRangeNumber startNumber endNumber))
202   )
203 )
204
205 (defun selectQuery (q)
206   (setf temp (query q))
207   (if (listp temp)
208       (format t "~%~{~{a:~a~%}~%~}" temp)
209       (format t "~%" temp))
210   )
211 )

```

Результат работы программы: Содержимое файла до выполнения запросов – ((:START "Москва":END "Симферополь":NUMBER "99") (:START "Ев-

патория":END "Санкт-Петербург":NUMBER "2") (:START "Севастополь":END  
"Москва":NUMBER "1"))

Добавить маршрут из Москва в Краснодар с номером 418  
Добавить маршрут из Краснодар в Симферополь с номером 8

Какие маршруты идут в город Симферополь

START:Краснодар  
END:Симферополь  
NUMBER:8

START:Москва  
END:Симферополь  
NUMBER:99

Найти маршрут номер 8

START:Краснодар  
END:Симферополь  
NUMBER:8

Какой маршрут идет в Краснодар

START:Москва  
END:Краснодар  
NUMBER:418

Показать все маршруты

START:Краснодар  
END:Симферополь  
NUMBER:8

START:Москва  
END:Краснодар  
NUMBER:418

START:Москва  
END:Симферополь  
NUMBER:99

START:Евпатория  
END:Санкт-Петербург  
NUMBER:2

START:Севастополь  
END:Москва  
NUMBER:1



Изменить маршруты с конечной станцией Симферополь на Омск  
 Изменить начальный маршрут у 1 номера на город Красноярск

Показать все маршруты

START:Краснодар  
 END:Омск  
 NUMBER:8

START:Москва  
 END:Краснодар  
 NUMBER:418

START:Москва  
 END:Омск  
 NUMBER:99

START:Евпатория  
 END:Санкт-Петербург  
 NUMBER:2

START:Красноярск  
 END:Москва  
 NUMBER:1

NIL

Рисунок 2 – Результат работы программы. Часть 2

Выбрать маршруты с номерами от 1 до 9

START:Краснодар  
 END:Омск  
 NUMBER:8

START:Евпатория  
 END:Санкт-Петербург  
 NUMBER:2

Рисунок 3 – Результат работы программы. Часть 3

## **ВЫВОДЫ**

В ходе выполнения лабораторной работы был исследован алгоритм сопоставления с образцом и особенности его применения для формирования запросов к базам данных, а также для организации доступа к базам данных на ограниченном подмножестве естественного языка.