МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

«**Севастопольский государственный университет**»

кафедра Информационных систем

Институт информационных технологий и управления в технических системах

Лисянский Александр Игоревич

курс 4 группа ИС/б-42-о

09.03.02 Информационные системы (уровень бакалавриата)

**ОТЧЁТ**

о лабораторном практикуме №4

по дисциплине «Методы создания искусственного интеллекта»

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность) (подпись) (инициалы, фамилия)

Севастополь

2015

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.

1. **Задание**

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.

Таблица 3.12. Корректировка данных в базе по номеру; вывод на дисплей анкетных данных студентов, получивших одну оценку 3; если таких студентов нет, вывести соответствующее сообщение.



1. **Код Lisp**

(**defun** CreateDB (a b c d e f g)

(**list** (**list** :number a :fio b :yearb c :yearp d :markF e :markVM f :markPr g))

)

(**defun** AddRow(db a b c d e f g)

(adjoin (**list** :number a :fio b :yearb c :yearp d :markF e :markVM f :markPr g) db)

)

(**defun** match (p d)

(**cond**

*;правило 1*

((**and** (**null** p)(**null** d)) **t**)

*;правило 2*

((**and** (**null** d)

(**eq** (**car** p) ’$)

(**null** (**cdr** p))) **t**)

*; один из списков исчерпан*

((**or** (**null** p)(**null** d)) **nil**)

*;правило 3 и правило 4*

((**or** (**equal** (**car** p) ’?)

(**equal** (**car** p) (**car** d)))

(match (**cdr** p)(**cdr** d)))

*;правило 5 и 6*

((**eq** (**car** p) ’$)

(**cond** ((match (**cdr** p) d) **t**)

((match p (**cdr** d)) **t**)))

*;правило 7*

((**and** (**not** (**atom** (**car** p)))

(**not** (**atom** (**car** d)))

(match (**car** p)(**car** d)))

(match (**cdr** p) (**cdr** d)) )

*;правило 8*

((**and** (**atom** (**car** p))

(**eq** (car-letter (**car** p)) #\$))

((match (**cdr** p)(**cdr** d))

(**set** (cdr-name (**car** p)) (**list** (**car** d)))

**t**)

((match p (**cdr** d))

(**set** (cdr-name (**car** p))

(**cons** (**car** d)(**eval** (cdr-name (**car** p)))))

**t**)

)

*;правило 9 и 10*

((**and** (**not**(**atom** (**car** p)))

(**eq** (**caar** p) 'restrict)

(**eq** (**cadar** p) '?)

(and-to-list

(**mapcar** #'(**lambda** (pred)

(**funcall** pred (**car** d))) (**cddar** p)

)))

(match (**cdr** p)(**cdr** d)))

*;правило 11*

((**and** (**not** (**atom** (**car** p)))

(**not** (**atom** d))

(**eq** (**caar** p) 'restrict)

(**eq** (car-letter (**cadar** p)) #\?)

(and-to-list

(**mapcar** #'(**lambda** (pred)

(**funcall** pred (**car** d))) (**cddar** p)))

(match (**cdr** p)(**cdr** d)))

(**set** (cdr-name (**cadar** p)) (**car** d))

**t**)

)

)

(**defun** get-matches (p db) *;db - база данных ;p - запрос (образец)*

(**cond** ((**null** db) **nil**)

((match p (**car** db))

(**cons** (**car** db) (get-matches p (**cdr** db)))

)

(**t** (get-matches p (**cdr** db)))

)

)

(**defun** car-letter (x) (**if** (**not** (**numberp** x)) (**car** (coerce (**string** x) 'list))))

(**defun** cdr-name (x)

(**intern** (coerce (**cdr** (coerce (**string** x) 'list)) 'string)))

(**defun** and-to-list ( lis )

(**let** ((res **t**))

(**dolist** (temp lis res)

(**setq** res (**and** res temp)))))

(**defun** zapros (db)

(**cond**

((match '($ state in ?V $) (read))

(**dolist** (elem (get-matches (**list** ':number '? ':fio '? ':yearb '? ':yearp V ':markF '? ':markVM '? ':markPr '?) db))

(format **t** "~%~{~a:~10t~a~%~}" (**list** :fio (**getf** elem :fio)) ))

)

((match '($ born in ?V $) (read))

(**dolist** (elem (get-matches (**list** ':number '? ':fio '? ':yearb V ':yearp '? ':markF '? ':markVM '? ':markPr '?) db))

(format **t** "~%~{~a:~10t~a~%~}" (**list** :fio (**getf** elem :fio)) ))

)

((match '($ student ?S mark of programming $) (read))

(**dolist** (elem (get-matches (**list** ':number '? ':fio S ':yearb '? ':yearp '? ':markF '? ':markVM '? ':markPr '?) db))

(format **t** "~%~{~a:~10t~a~%~}" (**list** :fio (**getf** elem :fio) :markPr (**getf** elem :markPr)) ))

)

((match '($ student ?S mark of fizics $) (read))

(**dolist** (elem (get-matches (**list** ':number '? ':fio S ':yearb '? ':yearp '? ':markF '? ':markVM '? ':markPr '?) db))

(format **t** "~%~{~a:~10t~a~%~}" (**list** :fio (**list** :fio (**getf** elem :fio) :markF (**getf** elem :markF)) ))

)

((match '($ student ?S mark of mathematics $) (read))

(**dolist** (elem (get-matches (**list** ':number '? ':fio S ':yearb '? ':yearp '? ':markF '? ':markVM '? ':markPr '?) db))

(format **t** "~%~{~a:~10t~a~%~}" (**list** :fio (**getf** elem :fio) :markVM (**getf** elem :markVM)) ))

)

(**t** (**princ** "Неизвестный запрос"))

)

)

)

(**setf** db (CreateDB 1 "Икитян" 1995 2012 5 3 4))

(**setf** db (AddRow db 2 "Лисянский" 1994 2012 5 5 5))

(**setf** db (AddRow db 4 "Дядюшенко" 1994 2012 5 4 5))

(**setf** db (AddRow db 5 "Инальев" 1994 2012 4 3 4))

*;(zapros db)*

*;("запрос")*

1. **Результат выполнения**
2. **Вывод**

В ходе работы были изучены способы организации простых баз данных с помощью А-списков и списков свойств, получены практических навыки использования и разработки функций высшего порядка, изучение средств файлового ввода-вывода в языке Лисп.