Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

ОТЧЕТ

по дисциплине

«Программирование на языке Лисп для систем искусственного интеллекта»

к лабораторной работе № 3

«Применение списков и функций высших порядков для организации баз данных»

Выполнил: ст. гр. ИСб-41-о

Никулин К.В.

Проверил:

Сметанина Т.И.

Севастополь

2016

1 ЦЕЛЬ

Исследование способов организации простых баз данных с помощью А- списков и списков свойств, получение практических навыков использования и разработки функций высшего порядка, изучение средств файлового ввода-вывода в языке Лисп.

ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

3.4.1. Ознакомиться по лекционному материалу или учебному пособию [1] с функциями ввода-вывода языка Лисп, функциями обработки А-списков и списков свойств, функционалами и замыканиями. Выполнить примеры функций, приведенные в разделе 3.2 настоящей лабораторной работы.
3.4.2. Ознакомиться с вариантом задания и выбрать одну из списковых структур (А-список, список свойств символа, список символов-ключей и их значений) для хранения записей таблицы. Привести обоснование выбора.
3.4.3. Определить на языке Лисп функции добавления записи в базу, функции сохранения базы на диске и загрузки базы в оперативную память, функцию просмотра базы на экране.
3.4.4. Создать в среде программирования Лисп-проект в соответствии с методическими указаниями [2], содержащий подготовленные определения функций, указанных в п. 3.4.3.
3.4.5. Выполнить частичную отладку проекта.
3.4.6. Подготовить определения дополнительных функций в соответствии с вариантом.
3.4.7. Выполнить полную отладку проекта и зафиксировать результаты работы программы в виде экранных копий.
3.4.8. Придумать 3-4 дополнительных запроса к базе данных и оценить объем возможных изменений (дополнений) в программе

2 ТЕКСТ ПРОГРАММЫ

Вариант 15

```
;gnu dmd 2.49
#|Написать программу, обеспечивающую создание на диске базы данных.
Структура базы данных определяется одной из таблиц в соответствии с вариан-
том задания. В функции программы должно входить :
− создание базы данных; +
− добавление записи в базу данных; +
− сохранение базы данных на диске;
− загрузка базы данных в оперативную память;
− просмотр информации. +
Таблица 3.9. Корректировка данных в базе по фамилии; +
вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет,
выдать на дисплей соответствующее сообщение. +
```

Таблица 3.9:
Фамилия_Имя Знак_зодиака Дата_рождения
день месяц год

```
|#
;создание одной записи
(defun make-record (fio zodiak day month year)
      (list :fio fio :zodiak zodiak :birthDate (list day month year)))

;определение базы
(defvar *db* nil)


(defun add-record (record) (push record *db*))
(add-record (make-record "Fio1" "Zodiak1" 11 2 3))
(add-record (make-record "Fio2" "Zodiak2" 1 23 3))
(add-record (make-record "Fio3" "Zodiak3" 1 2 32))


(defun dump-dp (*data*)
      (dolist (rec *data*)
            (;format t "~{~a:~12t~a~}~%" rec
            format t "Name: ~a; Zodiak: ~a; Birtday: ~a.~a.~a~%" (getf rec :fio) (getf rec :zodiak) (first
(getf rec :birthDate))
                                    (second (getf rec :birthDate)) (third (getf rec :birthDate))
            )
      )
)
;(dump-dp *db*)

(defun find-data(*data* who)
   (setf isInDB 0)
   (dolist (rec *data*)
      (when (equalp (getf rec :fio) who)
         (format t "Name: ~a; Zodiak: ~a; Birtday: ~a.~a.~a~%" (getf rec :fio) (getf rec :zodiak) (first (getf
rec :birthDate))
                                    (second (getf rec :birthDate)) (third (getf rec :birthDate)))
         (setf isInDB 1)
         (return)
      )
   )
   (when (= isInDB 0)
      (print "there is no such record")
   )
)

;(find-data *db* "Fio3")

(defun replace-data(*data* who whom)
   (setf isInDB 0)
   (dolist (rec *data*)
      (when (equalp (getf rec :fio) who)
         (setf (getf rec :fio) whom)
         (setf isInDB 1)
         (return)
      )
   )
   (when (= isInDB 0)
```

```
        (print "there is no such record for replace")
    )
)

(defun load-file()
  (with-open-file (stream "C:/Users/Konstantin/Desktop/in.txt")
    (do (
         (
          line (read-line stream nil)
          (read-line stream nil)
          )
         )
        ((null line))

      (setq fio line)
      (setq zod (read-line stream nil))
      (setq day (read-line stream nil))
      (setq mon (read-line stream nil))
      (setq year (read-line stream nil))
      ;(format t "~a ~a ~a ~a ~a" fio zod day mon year)
      (add-record (make-record fio zod day mon year))
      ;(add-record (make-record "Fio32" "Zodiak32" 8 8 8))
      ;(print "hello")
    )
    )
)

;(load-file)
;(dump-dp *db*)

(defun save-to-file()
  (with-open-file (out "C:/Users/Konstantin/Desktop/out.txt"
                :direction :output
                :if-does-not-exist :create
                )
      (dolist (rec *db*)
          (print (getf rec :fio) out)
          (print (getf rec :zodiak) out)
          (print (first (getf rec :birthDate)) out)
          (print (second (getf rec :birthDate)) out)
          (print (third (getf rec :birthDate)) out)
      )
  )
)

;(save-to-file)
(setq *db* nil)
```

3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ

```
Name: Fio3; Zodiak: Zodiak3; Birtday: 1.2.32
Name: Fio2; Zodiak: Zodiak2; Birtday: 1.23.3
Name: Fio1; Zodiak: Zodiak1; Birtday: 11.2.3
;;;*** Warning in FIND-DATA: *DATA* bound lexica
```

Рисунок 1 – Результат добавления трех записей и вывода их в буфер

```
"Fio3"
"Zodiak3"
1
2
32
"Fio2"
"Zodiak2"
1
23
3
"Fio1"
"Zodiak1"
11
2
3
```

Рисунок 2 – Результат записи в файл

ВЫВОДЫ

В ходе работы исследовали способы организации простых баз данных с помощью А-списков и списков свойств, получили практические навыки использования и разработки функций высшего порядка, изучили средства файлового ввода-вывода в языке Лисп.