

Міністерство освіти і науки України
Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки
інформації та управління

Мультипарадигмненне програмування

ЗВІТ

з лабораторної роботи № 3
Варіант №1

**Виконав
студент**

групи ПП-71, Амброс
Всеволод Володимирович

(№ групи, прізвище, ім'я, по батькові)

Прийняв

Очеретяний Олександр
Костянтинович

(посада, прізвище, ім'я, по батькові)

1. Ціль та задачі

Целью работы является изучение возможностей GCLisp (newLISP-tk) по организации динамических баз данных.

2. Завдання

Ознакомиться (на материале лекционного курса) с имеющимися в GCLisp'e (newLISP-tk) средствами для работы с файлами на внешних носителях, ассоциативными списками и списками свойств.

Ознакомиться с примером динамической базы данных машинного словаря основ слов русского языка из лекционного курса. Выполнить тестовый пример.

Написать программу, обеспечивающую создание на диске базы данных и работу с ней. В функции программы должно входить :

- создание базы данных;
- добавление информации в базу данных;
- модификацию (редактирование) информации;
- запись базы данных на диск;
- загрузку базы данных в оперативную память;
- просмотр информации;
- удаление информации из базы данных;
- поиск информации в базе данных;
- сортировка информации.

Программа должна предоставлять пользователю дружелюбный интерфейс. Вызов функций программы должен осуществляться из меню. Ввод исходных данных и вывод результата осуществлять в рабочем окне. За основу рекомендуется взять пользовательский интерфейс из лабораторной работы №6.

Вариант задания указан в таблице.

3. Звіт

1. Опис команд для командного рядку

```
1  ;; usage
2  ;; exit, q - exit cli
3  ;; set - set key
4  ;; get - get key
5  ;; del - delete key
6  ;; print - print tabled database
7  ;; load - load database from file
8  ;; save - save database to file
```

2. Функція що запитує у користувача ввід даних або команди

```
12  (defun prompt (&optional (str ""))
13    (format t "~a > " str)
14    (read)
15  )
```

3. Код роботи з командним рядком

```
17 (defun exec-cmd (cmd &optional (db (db-init)))
18   (cond
19     ((member cmd '(exit q))
20      (format t "see you soon...")
21     )
22     ((eq cmd 'save)
23      (db-save "db.txt" db)
24      (exec-cmd (prompt) db)
25     )
26     ((eq cmd 'load)
27      (let ((t-db (db-load "db.txt")))
28        (exec-cmd (prompt) t-db)
29      )
30     )
31     ((eq cmd 'set)
32      (let
33        ((t-db (db-set db (prompt "key") (prompt "value"))))
34        (exec-cmd (prompt) t-db)
35      )
36     )
37     ((eq cmd 'get)
38      (let ()
39        (print (db-get db (prompt "key")))
40        (terpri)
41        (exec-cmd (prompt) db)
42      )
43     )
44   )
45 )
```

```
44 ✓ ((eq cmd 'del)
45 ✓   (let
46     ((t-db (db-del db (prompt "key"))))
47     (exec-cmd (prompt) t-db)
48   )
49 )
50 ✓ ((eq cmd 'print)
51   (db-print db)
52   (terpri)
53   (exec-cmd (prompt) db)
54 )
55 ✓ (t
56   (format t "there is nothing to do")
57   (terpri)
58   (exec-cmd (prompt) db)
59 )
60 )
61 )
```

4. Код роботи з базою даних

```
2  (defun db-init () '())
3
4  (defun db-size (db)
5    (tuple-len db)
6  )
7
8  (defun db-print (db &optional (do-head t))
9    (terpri)
10   (if do-head
11     (let ()
12       (format t "key~a|~a~a" #\tab #\tab)
13       (terpri)
14     )
15   )
16   (if (not (null db))
17     (let ()
18       (format t "~a~a|~a~a" (caar db) #\tab #\tab (cadr db))
19       (db-print (cdr db) nil)
20     )
21   )
22 )
```

```
24 (defun db-clear ()
25   NIL
26 )
27
28 (defun db-set (db key val)
29   (tuple-set db key val)
30 )
31
32 (defun db-get (db key)
33   (tuple-get db key)
34 )
35
36 (defun db-has (db key)
37   (tuple-has db key)
38 )
39
40 (defun db-del (db key)
41   (tuple-del db key)
42 )
```

5. Код роботи за парами ключ-значення

```
44 ;; tuple (dict) primitive operations
45 (defun tuple-len (tup)
46   (if (null tup)
47       0
48       (length tup)
49   )
50 )
51
52 (defun tuple-verify (tup)
53   (if (and (listp (car tup)) (eq (length tup) 2))
54       tup
55       (error "Storage injured!")
56   )
57 )
58
59 (defun tuple-get (tup key)
60   (cond
61     ((null tup) nil)
62     ((equal (caar tup) key) (cadar tup))
63     (t (tuple-get (cdr tup) key))
64   )
65 )
66
67 (defun tuple-set (tup key val)
68   (cons (list key val) tup)
69 )
70
71 (defun tuple-del (tup key)
72   (cond
73     ((null tup) nil)
74     ((equal (caar tup) key) (cdr tup))
75     (t (cons
76         (car tup)
77         (tuple-del (cdr tup) key)
78       ))
79   )
80 )
```

```
81
82  (defun tuple-has (tup key)
83    (cond
84      ((null tup) nil)
85      ((equal (caar tup) key) t)
86      (t (tuple-has (cdr tup) key))
87    )
88  )
89
```


6. Код роботи зі збереженням та завантаженням БД

```
91 ;; load/save
92 (defun db-save (path db)
93   (with-open-file
94     (
95       out path :direction :output
96       :if-exists :supersede :if-does-not-exist :create
97     )
98     (with-standard-io-syntax (print db out))
99     (format out "~%")
100   )
101   db
102 )
103
104 (defun db-load (path)
105   (with-open-file (in path)
106     (with-standard-io-syntax (read in))
107   )
108 )
109
```

7. Приклад роботи з БД

```
> print

key      |      value

> set
key > foo
value > bar
> set
key > hello
value > world
> get
key > foo

BAR
> get
key > hello

WORLD
>
print

key      |      value
HELLO    |      WORLD
FOO      |      BAR

> get bar
key >
NIL
```

4. Висновки

В даній лабораторній роботі мною було розроблене просте сховище даних (key-value storage) мовою GNU/Lisp. Програма розбита на 2 файли. В одному описана логіка роботи бази даних "ключ-значення". В іншому — опис роботи з бд через командний рядок. Були реалізовані такі операції:

- додавання пари ключ-значення в БД (set);
- видалення значення через ключ (del);
- отримання значення через ключ (get);
- збереження БД на постійну пам'ять (save);
- завантаження БД з постійної пам'яті (load);
- друк БД в командний рядок в табличному виді (print);
- вихід (exit або q).

Сама БД зберігається у вигляді списку списків-пар ((key1 value1) (key2 value2) ...).

Пошук, отримання та видалення ключа реалізовано через рекурсію. Збереження та завантаження БД реалізовано через вбудовані макроси GNU/Lisp.