

Міністерство освіти та науки України
Національний технічний університет України
“Київський політехнічний інститут”
Кафедра АСОІУ

ЗВІТ
про виконання комп'ютерного практикуму №4
з дисципліни
“Мультипарадигменне програмування”
Тема: «Робота з динамічними базами даних»

Перевірів:
Баклан І.В.

Виконав:
студент 3-го курсу
групи ПІ-52
Набоков Е.М.

Київ 2017

Мета роботи: вивчення можливостей GCLisp по організації динамічних баз даних.

Завдання: ознайомитися (на матеріалі лекційного курсу) з наявними в GCLisp'e (NewLISP-tk) засобами для роботи з файлами на зовнішніх носіях, асоціативними списками і списками властивостей.

Ознайомитися з прикладом динамічної бази даних машинного словника основ слів російської мови з лекційного курсу.

Виконати тестовий приклад. Написати програму, що забезпечує створення на диску бази даних і роботу з нею.

Результат

```
(DEFUN SETUP (PROD type naming)
  (LIST :PROD PROD :type type :naming naming))
```

```
(DEFVAR *db* nil)
```

```
(DEFUN add-record (PARAM) (PUSH PARAM *db*))
```

```
(DEFUN dump-db ()
  (MAKELIST (PARAM *db*)
    (format t "~{~a:~10t~a~%~}~%" PARAM)))
```

```
(DEFUN select (selector_paramFN)
  (DROP-IF-NOT selector_paramFN *db*))
```

```
(DEFUN selector (type)
  (LAMBDA (PARAM) (equal (GETF PARAM :type) type)))
```

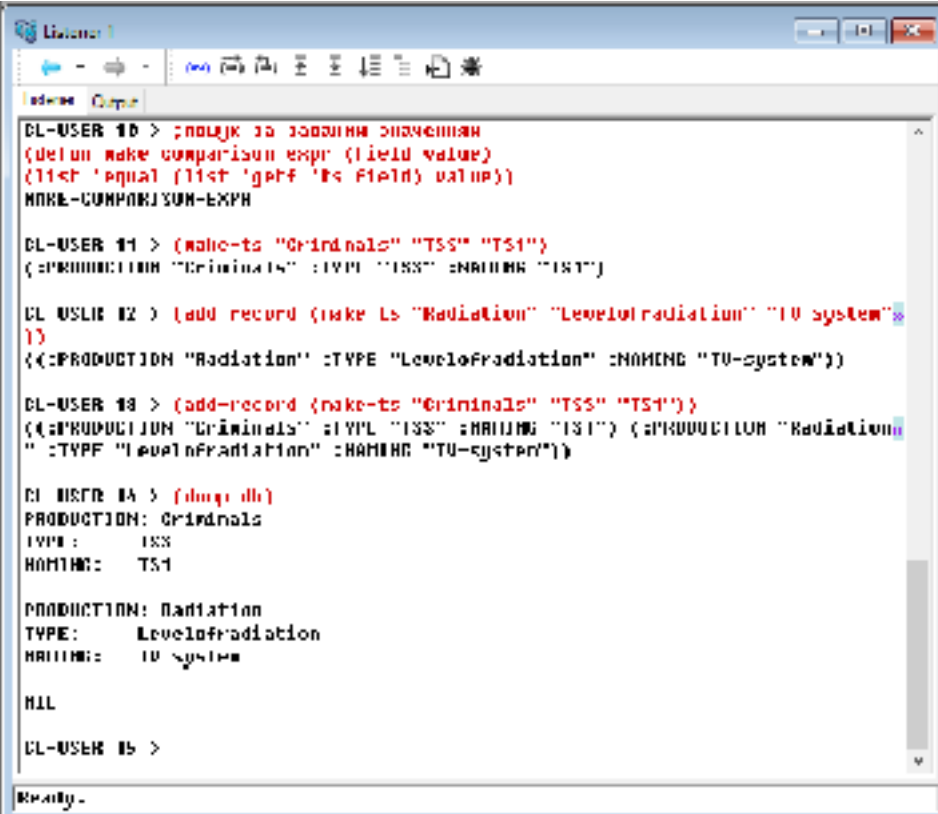
```
(DEFUN where (&key PROD type naming)
  (LAMBDA (PARAM)
    (and
      (IF PROD (equal (GETF PARAM :PROD) PROD) t)
      (IF type (equal (GETF PARAM :type) type) t)
      (IF naming (equal (GETF PARAM :naming) naming) t))))
```

```
(DEFUN update (selector_paramFN &key PROD type naming (ripped nil ripped-p))
  (SETF *db*
    (MAPCAR
      (LAMBDA (row)
        (WHEN (FUNCALL selector_paramFN row)
          (IF PROD (SETF (GETF row :PROD) PROD))
          (IF type (SETF (GETF row :type) type))
          (IF naming (SETF (GETF row :naming) naming))))
```

```
row) *db*))
```

```
(DEFUN delete-rows (selector_paramFN)  
(SETF *db* (DROP-IF selector_paramFN *db*)))
```

```
(DEFUN COMPARE_EXPRESSIONS (field value)  
(LIST 'equal (LIST 'GETF 'PARAM field) value))
```



```
CL-USER 10 > ;написати функцію порівняння  
(defun make-comparison expr (field value)  
  (list 'equal (list 'getf 'PARAM field) value))  
MAKE-COMPARISON-EXPR  
  
CL-USER 11 > (make-ts "Criminals" "TSS" "TS1")  
(=:PRODUCTION "Criminals" :TYPE "TSS" :NAMING "TS1")  
  
CL-USER 12 > (add-record (make-ts "Radiation" "Levelofradiation" "TU-system")  
  )  
(=:PRODUCTION "Radiation" :TYPE "Levelofradiation" :NAMING "TU-system")  
  
CL-USER 13 > (add-record (make-ts "Criminals" "TSS" "TS1")  
  (=:PRODUCTION "Criminals" :TYPE "TSS" :NAMING "TS1") (=:PRODUCTION "Radiation"  
    :TYPE "Levelofradiation" :NAMING "TU-system"))  
  
CL-USER 14 > (dump db)  
PRODUCTION: Criminals  
TYPE: TSS  
NAMING: TS1  
  
PRODUCTION: Radiation  
TYPE: Levelofradiation  
NAMING: TU-system  
  
NIL  
  
CL-USER 15 >
```

Рис. 1. Реалізація завдання в Lisp

Висновок: у даній лабораторній роботі я дізнався можливості організації динамічних баз даних. Також я ознайомився з наявними GCLisp'є засобами для роботи з файлами на зовнішніх носіях, асоціативними списками і списками властивостей.