Міністерство освіти та науки України

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

**ЗВІТ**

про виконання

Лабораторної роботи №4

з дисципліни:

«**Декларативне програмування**»

**Тема: «Робота з динамічними базами даних»**

**Варіант 6**

**Виконав:** студент групи ІП-51

Зарічковий Олександр Анатолійович

**Перевірив**: доц. Баклан І. В.

Київ 2017

**Варіант 6**

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

**Завдання:** ознайомитися (на матеріалі лекційного курсу) з наявними в GCLisp'е (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.

**Виконання завдань**

***Завдання***

|  |  |
| --- | --- |
| *Варіант* | *Задача* |
| 6 | Продукція підприємства. Пристрої для організації телевізійних систем [8]. |

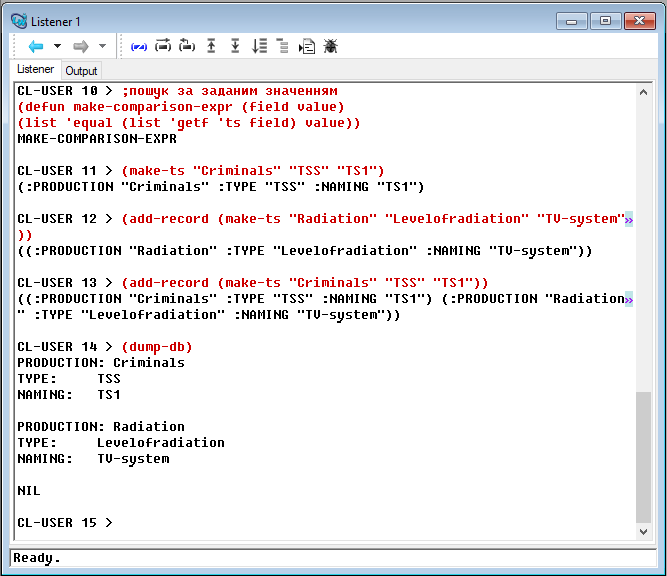
*Розв’язання*

Код програми:

|  |
| --- |
| (defun make-ts (production type naming)  (list :production production :type type :naming naming))  ;глобальна змінна  (defvar \*db\* nil)  ;додавання запису в базу даних  (defun add-record (ts) (push ts \*db\*))  ;виводить зміст бази данних в більш читабельній формі  (defun dump-db ()  (dolist (ts \*db\*)  (format t "~{~a:~10t~a~%~}~%" ts)))  ;вибирає деяке значення з бази даних  (defun select (selector-fn)  (remove-if-not selector-fn \*db\*))  ;вибирає тип телевізійної системи  (defun type-selector (type)  (lambda (ts) (equal (getf ts :type) type)))  ;генерує вираз вибору, яке повертає всі записи про тел. системи, які співпадають зі значеннями, заданими в where  (defun where (&key production type naming)  (lambda (ts)  (and  (if production (equal (getf ts :production) production) t)  (if type (equal (getf ts :type) type) t)  (if naming (equal (getf ts :naming) naming) t))))  ;оновлення та використання аргументов-ключів для задання нового значення  (defun update (selector-fn &key production type naming (ripped nil ripped-p))  (setf \*db\*  (mapcar  (lambda (row)  (when (funcall selector-fn row)  (if production (setf (getf row :production) production))  (if type (setf (getf row :type) type))  (if naming (setf (getf row :naming) naming)))  row) \*db\*)))  ;видалення рядків із бази даних  (defun delete-rows (selector-fn)  (setf \*db\* (remove-if selector-fn \*db\*)))  ;пошук за заданим значенням  (defun make-comparison-expr (field value)  (list 'equal (list 'getf 'ts field) value)) |

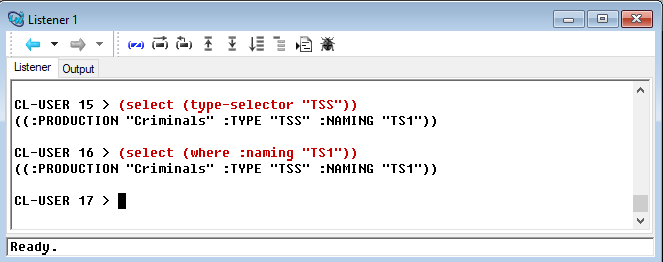
Нижче приведено реалізацію програми у середовищі LispWorks Personal Edition 6.0.1. На рисунку бачимо

* Представлення телевізійної системи
* Додавання запису в базу даних
* Перегляд вмісту бази даних



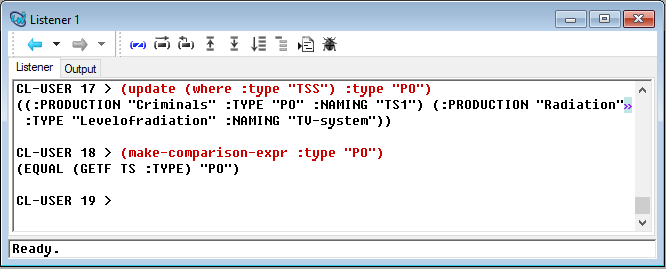
***Рис. 1.*** Реалізація завдання в LispWorks Personal Edition 6.0.1

Виберемо тип телевізійної системи та здійснимо пошук певного значення в базі даних.



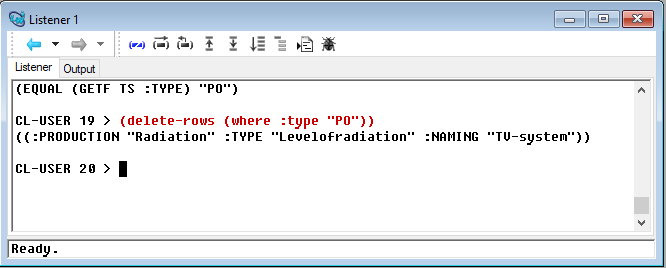
***Рис. 2.*** Реалізація завдання в LispWorks Personal Edition 6.0.1

Уведемо нове значення в базу даних та виконаємо пошук за заданим значенням.



***Рис. 3.*** Реалізація завдання в LispWorks Personal Edition 6.0.1

Видалимо елемент.



***Рис. 4.*** Реалізація завдання в LispWorks Personal Edition 6.0.1

**Висновок:**  у ході виконання лабораторної роботи я познайомився з можливостями GCLisp (newLISP-tk) по організації динамічних баз даних. Також я ознайомився (на матеріалі лекційного курсу) з наявними в GCLisp'е (NewLISP-tk) засобами для роботи з файлами на зовнішніх носіях, асоціативними списками і списками властивостей. Переглянув приклад динамічної бази даних машинного словника на основ слів російської мови з лекційного курсу.