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:

«**Мультипарадигменне програмування**»

**Тема: «Локальные определения и функционалы»**

**Варіант 1**

**Виконав:** студент групи ІП-51

Булатов Дмитро Эгорович

**Перевірив**: викл.

Баклан Ігор Всеволодович

Київ 2017

**Мета роботи:**

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

**Основні задачі:**

* Изучить применение техники нисходящей и восходящей рекурсии при написании рекурсивных функций с использованием локальных определений;
* Сравнить возможности локальных определений LET и LAMBDA по организации вычислений в рекурсивных программах;

**Завдання**

Ознакомиться (на материале лекционного курса) с имеющимися в GCLisp’е (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.

**Код програми:**

(defvar db)

(setf db '())

(defun insert(A)

(setf db (append (list A) db))

)

(defun select(A)

(print A)

)

(defun where(A B)

(remove nil (mapcar B A))

)

(defun update(A) ;update with nil to delete

(setf db (remove nil (mapcar A db)))

)

(defun ins (A B cmp)

(cond

((null B) (cons A '()))

((funcall cmp A (car B)) (cons A B))

(T (cons (car B) (ins A (cdr B) cmp)))

)

)

(defun insSort (A cmp)

(cond

((null A) '())

(T (ins (car A) (insSort (cdr A) cmp) cmp))

)

)

(insert '(1 2 3))

(insert '((3 2 1)))

(insert '("a" "b" "c"))

(select db)

(select (where db (

lambda(A)

(cond

((atom (car A)) A)

(T nil)

)

)))

(update

(

lambda(A)

(cond

((atom (car A)) A)

(T (car A))

)

))

(select (where db (

lambda(A)

(cond

((atom (car A)) A)

(T nil)

)

)))

(update

(

lambda(A)

(cond

((numberp (car A)) A)

(T nil)

)

))

(select db)

(insert '(2 2 4))

(insert '(7 3 7))

(insert '(5 2 1))

(select (insSort db

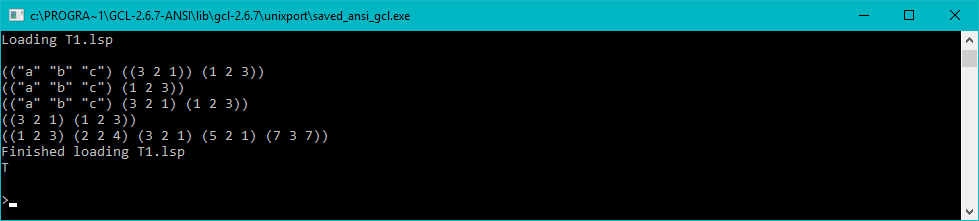
(lambda (A B)

(< (car A) (car B))

)

))

**Скріншот програми:**



**Висновок:**

В цій лабораторній роботі я опанував використання глобальних змінних та навчився реалізовувати обробку та виведення данних у мові Lisp.