Міністерство освіти і науки України

Національний лісотехнічний університет України

Кафедра інформаційних технологій

**Звіт до лабораторної роботи №8**

з навчальної дисципліни

**«Комп’ютерна графіка»**

на тему:

**“AutoLisp”**

**Виконав:**

Студент групи: КН-11/2

Манжула Д. В.

**Перевірив:**

Сторожук О.Л.

Львів – 2016

**Робоча програма:**

Створюю функцію зміни кольору:

*(defun c:zmkol ( / kol le ll ls )*

Вибираю колір:

*(setq kol (acad\_colordlg 3 ))*

Вибираю об’єкт:

*(setq le (car(entsel "Введіть об’єкт: ")))*

Перевірка, чи вибрали об’єкт:

*(if (not le)*

*(progn*

*(alert "Ви не вибрали обє’кт")*

*(while (not le) ;перевірка до того часу, пои не буде вибраний обє’кт*

*(setq le (car(entsel "Виберіть об’єкт: ")))*

*(if (not le) (alert "Ви не вибрали обєкт"));*

*)*

*)*

*) (setq ll (entget le)) ;для відрізка повертається обєкт*

Перевірка на наявність точкової пари з -кодом 62, якщо є, то змінюється, якщо нема, то добавляється

*(if (assoc 62 ll)*

*(setq ls (subst (cons 62 kol) (assoc 62 ll) ll))*

*(setq ls (list (cons -1 le) (cons 62 kol)))*

*)*

*(entmod ls)*

*(princ)*

Функція **defun** – це спеціальна функція, призначена для створення функцій користувача.

**Структура:**

(**defane** *<ім’я> ([<аргументи>] [ / <робочі>]) <вираз>*).

Аргументи: <*ім’я*>- символ, що використовується в якості імені нової функції; <*аргументи*> – символи, розділені пробілами і використовуються в якості аргументів нової функції; <*робочі*> - символи, розділені пробілами, які використовуються в якості (робочих) змінних нової функції; <*вираз*> – будь-яка послідовність виразів, які використовуються як аргументи і робочі змінні, так і інші функції і глобальн змінні.

**Приклад:**

*(*

*defune orsum (a1 a2 a3 / s)*

*(setq s (+a1 a2))*

*(-s a3)*

*)*

Функція **not** виконую оперецію логічного заперечення

**Структура:**

(not<аргумент>)

**Приклад:**

(not T) повертає nil.

(not nil) повертає Т.

Функція **if** виконує умовну операцію типу if-then-else (якщо-то-інакше)

**Структура:**

(if <умова> <вираз1> [<вираз2>])

**Приклад:**

(if ( > a b) 2 8) повертає 2, якщо a>b і 8 – в іншому випадку.

Функцфя **рrogn** об’єднює декілька (не менше двох) виразів в одне, коли по синтаксису LISP може використовуватися тільки одне (так як функція if).

**Структура:**

(progn <вираз1> <вираз2> … [<виразN>]).

**Приклад:**

(if ( > a b)

(progn

(setq c-1)

(setq d (- c a b))

(\* d b)

)

(progn

(setq c 2)

(setq d (\* a b ))

(\* c d)

)

)

Функція **While** виконує операцію цикла по багаторазовій умові, яка перевіряється.

**Структура:**

(while <умова> [<вираз1>] [<вираз2> … [<виразN>] …]])

**Приклад:**

(setq I l factorial l)

(while (<I n)

(setq I (l+1))

(setq factorial l (\* factorial i))

)

Функція **list -**основна функція, яка об’єднює всі елементи в список.

**Структура:**

(list [<елемент1>] [<елемент2> … [<елемент N>] …]])

**Приклад:**

(list 3.62 1.49 4.12) повертає (3.62 1.49 4.12)

Функція **car** видаляє перший елемент списка, точкової пари чи точкового списку.

**Структура:**

(car ,список>).

**Приклад:**

(car ‘(11 12 13 14 15)) повертає 11

Функція **cons** добавляє до списку перший елемент або створює точкову пару.

**Структура:**

(cons <аргумент1> <аргумент2>).

**Приклад:**

(cons 5.4 ‘(8.3 -1.0)) повертає (5.4 8.3 -1.0).

Функція **assoc** застосовується до складного списку, в якому елементами являються списки або точкові пари і вибирає з цих елементів той, у якого перший елемент має задане значення.

**Структура:**

(assoc <код> <список>).

**Приклад:**

(assoc 8 ‘((8 . “Walls”) (6 . “Dashed”))) повертає (8 . “Walls”).

Функція **subst** заміняє в списку старий елемент (всі входження) на новий.

**Структура:**

( subst <новий> <старий> <список>).

**Приклад:**

(subst 7 2 ‘(22 3 2 0)) повертає (22 3 7 0)).

Функція **acad\_colordlg** викликає діалогове вікно з палітрою, в якому користувач повинен вибрати колір в діапазоні від 0 до 256.

**Структура:**

(acad\_colordlg <номер> [<прапорець>]).

**Приклад:**

**(**acad\_colordlg 5**).**

Функція **princ** записує значення аргумента, який може бути будь-яким виразом LISP (а також числом, списком, стрічкою, ім’ям примітиву т т.д.) в вигляді текстового рядка в файл з заданим дескриптором.

**Структура:**

(princ [<аргумент> [<дескриптор>]]).

**Приклад:**

(princ ‘(4.5678123 11.1 ) fo) пише в файл і повертає (4.5678123 11.1 ).

Функція **alert** дозволяє вивести на екран діалогове вікно з повідомленням системи AutoCad**.**

**Структура:**

(alert <повідомлення>).

**Приклад:**

(setq lines («НЛТУ»))

(alert lines).

Функція **entsel** пропонує користувачу вказати один об’єкт, видаючи запит, текст якого заданий в якості аргумена.

**Структура:**

(entsel [<запит>]).

**Приклад:**

(setq esl (etsel “Виберіть об’єкт: ”)) повертає (<Entity name: 7ef66fa0> (475.13 284.759 0.0)).

Функція **entget** отримує список з характеристиками примітиву і являється основним інструментом видалення інформації про геометрію і властивості.

**Структура:**

(entget <примітив> [< додаток>]).

**Приклад:**

(setq le (entget ( entnext))) ((-1 . <Entity name: 7ef66fa0>) (0 . “Line”) (330 . <Entity name: 7ef66cf8>) (5 . “EC”) (100 . “AcDdEntity”) (67 . 0) (410 . “Model”

(8 . “0”) (100 . “AcDbLine”) (10 214.31 409.132 0.0)

(11 608.276 172.218 0.0) (210 0.0 0.0 1.0  
)).

Функція **entmod** змінює примітим в базі рисунка і повертає значення аргумента <список> при успішному завершенні або nil - при неможливості виконати зміни.

**Структура:**

(entmod <список>).

**Приклад:**

(setq le (subst (cons 62 le) le)) повертає список, в якому точкова пара кольору змінена на точкову пару з синім кольором (номер 5) (entmod le) міняє колір відрізка на синій.

**Висновок:** В цій лабораторній роботі було ознайомлено з мовою програмування Visual Lisp.