

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Запорізький національний технічний університет**

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до лабораторних робіт з дисципліни**  
**“Візуальне програмування в проектуванні САПР”**  
**для студентів спеціальності 7.080402**  
**„Інформаційні технології проектування ”**  
**денної форми навчання**

**2016**

Методичні вказівки до лабораторних робіт з дисципліни “Візуальне програмування в проектуванні САПР” для студентів спеціальності 7.080402 „Інформаційні технології проектування” денної форми навчання / Укладачі: Л.Ю. Дейнега. – Запоріжжя: ЗНТУ, 2016. – 45 с.

Укладачі: Л. Ю. Дейнега, ст. викладач

Рецензент: канд. техн. наук, доцент А. В. Пархоменко

Відповідальний за випуск: Л. Ю. Дейнега, ст. викладач

Затверджено  
на засіданні кафедри  
“Програмних засобів”  
Протокол № 1 від 16.09.2016 р.

## ЗМІСТ

1 ЛАБОРАТОРНА РОБОТА № 1.....	4
ВКОРИСТАННЯ VISUAL LISP ДЛЯ ПОБУДОВИ ГРАФІЧНИХ ОБ'ЄКТІВ В AUTOCAD.....	4
1.1 Мета роботи.....	4
1.2 Вказівки до виконання лабораторної роботи.....	4
1.3 Завдання до лабораторної роботи .....	5
1.4 Зміст звіту.....	5
1.5 Контрольні питання.....	6
2 ЛАБОРАТОРНА РОБОТА № 2.....	7
СТАНДАРТНІ ФУНКЦІЇ VISUAL LISP. РЕКУРСІЯ.....	7
2.1 Мета роботи.....	7
2.2 Вказівки до виконання лабораторної роботи.....	7
2.3 Завдання до лабораторної роботи .....	8
2.4 Зміст звіту.....	10
2.5 Контрольні питання.....	10
3 ЛАБОРАТОРНА РОБОТА № 3.....	11
РЕКУРСИВНА ОБРОБКА СПИСКІВ В VISUAL LISP .....	11
3.1 Мета роботи.....	11
3.2 Вказівки до виконання лабораторної роботи.....	11
3.3 Завдання до лабораторної роботи .....	12
3.4 Зміст звіту.....	13
3.5 Контрольні питання.....	13
ЛІТЕРАТУРА.....	14
Додаток А .....	15
Середовище Visual LISP .....	15
Додаток Б.....	37
Розробка застосунків .....	37

## 1 ЛАБОРАТОРНА РОБОТА № 1

### ВИКОРИСТАННЯ VISUAL LISP ДЛЯ ПОБУДОВИ ГРАФІЧНИХ ОБ'ЄКТІВ В AUTOCAD

#### 1.1 Мета роботи

**Мета роботи** - одержання практичних навиків побудови графічних об'єктів засобами AutoLisp.

#### 1.2 Вказівки до виконання лабораторної роботи

1.2.1 Розглянути приклади написання програми за темою.

Створити функцію, яка буде графічний об'єкт, поданий на рис.

1.1.

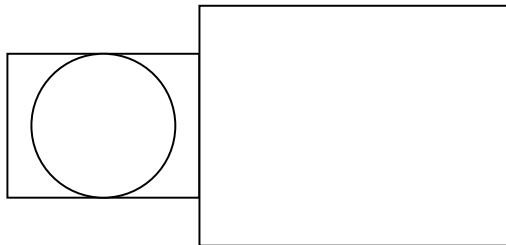


Рисунок 1.1 – Приклад графічного об'єкту

```
(defun graf ()
  (SETVAR "CMDECHO" 0) ; Відключення відлуння команд
  (setvar "BLIPMODE" 0) ; Відключення зображення маркера
  (command "limits" "0,0" "297,210" "ZOOM" "A" )
  (textscr) ; Перехід на текстовий екран
  (setq xt (getreal "\n Введіть координату x базової точки xt=")
    yt (getreal "\n Введіть координату у базової точки yt=")
    bt (list xt yt))
```

```
(setq d 40
      1 50
      r (* d 0.5))
;| функція (polar <T1> <U> <L>) визначає точку T2 шляхом
переміщення з точки <T1> під кутом <U> (кут задається в радіанах,
відлічується проти годинникової стрілки) на відстань <L> |;
(setq t1 (polar bt (/ pi 2) r)
      t2 (polar t1 0 1)
      t4 (polar bt (* 1.5 pi) r)
      t3 (polar t4 0 1))
(command "pline" bt "w" 0.1 0.1 t1 t2 t3 t4 "c")

(setq bt (polar bt 0 50)
      t1 (polar bt (/ pi 2) 30)
      t2 (polar t1 0 80)
      t4 (polar bt (* 1.5 pi) 30)
      t3 (polar t4 0 80))
(command "pline" bt "w" 0.1 0.1 t1 t2 t3 t4 "c")

(command "circle" (polar bt pi (/ 1 2)) r)
)

(graf) ; Виклик функції
```

### 1.3 Завдання до лабораторної роботи

1.3.1 Розробити складний графічний об'єкт, що містить 3-4 простих об'єктів (лінія, коло, дуга, полілінія, еліпс, багатокутник).

1.3.2 Розробити програму, яка малює цей графічний об'єкт.

### 1.4 Зміст звіту

Звіт з лабораторної роботи повинен містити

1.4.1 Тему та мету роботи.

1.4.2 Завдання до лабораторної роботи.

1.4.3 Текст програми.

1.4.4 Результати роботи програми.

### **1.5 Контрольні питання**

1.5.1 Використання в AutoLisp команд AutoCAD.

1.5.2 Основні графічні примітиви AutoCAD.

1.5.3 Основні функції AutoCAD для побудови графічних об'єктів.

## 2 ЛАБОРАТОРНА РОБОТА № 2

### СТАНДАРТНІ ФУНКЦІЇ VISUAL LISP. РЕКУРСІЯ

#### 2.1 Мета роботи

**Мета роботи** - одержання знань і навиків роботи в Visual LISP з використанням стандартних функцій та функцій користувача.

#### 2.2 Вказівки до виконання лабораторної роботи

2.2.1 Перед виконанням завдання рекомендується ознайомитись з матеріалом, який подано в додатку А – „Середовище Visual LISP” та додатку Б – „Розробка програм”.

2.2.2 Розглянемо приклади написання програми за темою.

Приклад 1. Перевірити знак числа.

```
(defun snumberp (num)
  (cond
    ((numberp num)
     (cond
       ((< num 0) 'neg_number)

       ((= num 0) 'zero)
       (> num 0) 'pos_number)
     ))
  (t 'not_number)
  )
)
```

У програмі перевіряється чи є аргумент числом. Якщо аргумент число то перевіряється його знак і виводиться відповідна інформація.

Приклад 2. Обчислити факторіал заданого числа.

(defun factorial (n)

(cond

((= n 0) 1) ;факториал 0! равен 1

(t (\* (factorial (- n 1)) n)))) ;факториал n! равен (n-1)!\*n

### 2.3 Завдання до лабораторної роботи

1 Дано натуральне число  $n$ . Обчислити

$$\frac{1}{\sin 1} + \frac{1}{\sin 2} + \dots + \frac{1}{\sin n}.$$

2 Дано натуральне число  $n$ . Обчислити

$$\sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}; (n \text{ коренів}).$$

3 Дано натуральне число  $n$ . Обчислити

$$\frac{\cos 1}{\sin 1} \cdot \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} \cdot \dots \cdot \frac{\cos 1 + \cos n}{\sin 1 + \sin n}.$$

4 Дано дійсне число  $a$  та натуральне число  $n$ . Обчислити

$$a(a+1)\dots(a+n-1).$$

5 Дано дійсне число  $a$  та натуральне число  $n$ . Обчислити

$$\frac{1}{a} + \frac{1}{a+1} + \dots + \frac{1}{a+n}.$$

6 Дано дійсне число  $a$  та натуральне число  $n$ . Обчислити

$$\frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2n}}.$$



7 Обчислити  $(1+\sin 0,1)(1+\sin 0,2)\dots(1+\sin 10)$ .

8 Дано дійсне число  $x$ . Обчислити

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}$$

9 Дано натуральне число  $n$ . Обчислити  $\sum_{i=1}^n \frac{1}{i^2}$ ;

10 Дано натуральне число  $n$ . Обчислити  $\sum_{i=1}^n \frac{1}{i^3}$ .

11 Дано натуральне число  $n$ . Обчислити  $\sum_{i=1}^n \frac{1}{i!}$ .

12 Дано натуральне число  $n$ . Обчислити  $\sum_{i=1}^n \frac{1}{(2i)^2}$ .

13 Дано натуральне число  $n$ . Обчислити  $\sum_{i=1}^n \frac{1}{(2i)!}$ .

14 Послідовність  $x_1, x_2, \dots$  створено за законом:

$$x_1=0; x_2=5/8; x_i=x_{i-1}/2+(3/4)x_{i-2}, \quad i=3,4,\dots$$

Обчислити  $x_1, x_2, \dots, x_{20}$ .

15 Послідовність  $x_1, x_2, \dots$  створено за законом:

$$x_1=1; x_2=0.3; x_i=(i+1)x_{i-2}, \quad i=3,4,\dots$$

Обчислити  $x_1, x_2, \dots, x_{20}$ .

16 Отримати таблицю температур за Цельсієм від 0 до 100 градусів та їх еквівалентів за шкалою Фаренгейта, використовуючи для переведення формулу

$$t_f = (9/5)t_c + 32.$$

17 Дано натуральне число  $n$ , дійсні числа  $a_1, \dots, a_n$  ( $n \geq 3$ ).

Отримати  $b_1, \dots, b_{n-2}$ , де  $b_i = a_{i+1} + a_{i+2}$ ,  $i = 1, \dots, n-2$ .

## 2.4 Зміст звіту

Звіт з лабораторної роботи повинен містити

- 2.4.1 Тему та мету роботи.
- 2.4.2 Завдання до лабораторної роботи відповідного варіанту (за списком групи).
- 2.4.3 Текст програми.
- 2.4.4 Результати роботи програми.

## 2.5 Контрольні питання

- 2.5.1 Пояснити, що таке символічні вирази Lisp і як вони будуються.
- 2.5.2 Які символи AutoLisp називають зарезервованими?
- 2.5.3 Назвіть можливі способи використання програм на AutoLisp в системі AutoCAD.
- 2.5.4 Перелічити вбудовані арифметичні функції AutoLisp, їх призначення та форму запису. Навести приклади.
- 2.5.5 Перелічити вбудовані логічні функції AutoLisp, їх призначення та форму запису. Навести приклади..
- 2.5.6 Перелічити вбудовані математичні функції AutoLisp, їх призначення та форму запису. Навести приклади.
- 2.5.7 Перелічити вбудовані конструкції AutoLisp, які керують виконанням програми, їх призначення та форму запису. Навести приклади.
- 2.5.8 Як будуються функції користувача?
- 2.5.9 Як будуються рекурсивні функції?

### 3 ЛАБОРАТОРНА РОБОТА № 3

## РЕКУРСИВНА ОБРОБКА СПИСКІВ В VISUAL LISP

### 3.1 Мета роботи

**Мета роботи** - одержання знань і навиків, необхідних для роботи із списками з використанням рекурсії.

### 3.2 Вказівки до виконання лабораторної роботи

3.2.1 Розглянути приклади написання програми за темою.

Приклад 1. Написати функцію копіювання списку.

```
(defun copy_list (list)
  (cond
    ((null list) nil) ;копією пустого списку є пустий список
    (t (cons (car list) (copy_list (cdr list))))))
```

Функція `copy_list` написана методом низхідної рекурсії. Функція одержує голову початкового списку і викликає себе рекурсивно з хвостом початкового списку. Дія повторюється до тих пір, поки хвіст початкового списку не стане порожнім списком. Тоді список результат також подається порожнім списком і до нього додається голова порожнього списку. Дія повторюється до тих пір поки не будуть приєднані всі голови початкового списку.

Наприклад:

```
>(setq N '(1 2 3))
>(setq M (copy_list N))
(1 2 3)
>N
(1 2 3)
>M
```

(1 2 3)  
 >(copy\_list ())  
 NIL

### 3.3 Завдання до лабораторної роботи

**При виконанні завдання обов'язково використовувати рекурсивні функції.**

1 Дано список цілих чисел. Визначити кількість елементів списку, які є непарними числами.

2 Дано список цілих чисел. Визначити кількість елементів списку кратних 3 і не кратних 5.

3 Дано список цілих чисел. Визначити кількість елементів списку, які є квадратами парних чисел.

4 Дано список цілих чисел. Визначити кількість елементів списку, що мають парні порядкові номери і які є непарними числами.

5 Дано список цілих чисел. Знайти ті елементи списку, які є непарними числами.

6 Дано список цілих чисел. Знайти ті елементи списку, які при діленні на 7 дають залишок 1 або 5.

7 Дано список цілих чисел. Отримати суму тих елементів списку, які непарні і негативні.

8 Дано список цілих чисел. Отримати суму тих елементів списку, які діляться на 5 і не діляться на 7.

9 Дано список цілих чисел. Замінити нулями елементи списку з парними порядковими номерами.

10 Дано список цілих чисел. Отримати подвоєну суму всіх позитивних елементів списку.

11 Дано список цілих чисел. Всі негативні елементи списку збільшити на 0.5, а всі інші замінити на 0.1.

12 Дано список цілих чисел. Отримати суму позитивних і число негативних елементів списку.

13 Дано список цілих чисел і задано число  $a$ . Якщо в списку є хоч би один елемент, рівний  $a$ , то отримати суму всіх елементів, наступних за першим таким елементом. Інакше результатом повинне бути число 10.

14 Дано список цілих чисел. Отримати найменше з парних чисел, що входять в список.

15 Дано список цілих чисел. Отримати найбільше з непарних чисел, що входять в список.

16 Дано список цілих чисел. Знайти номер першого парного елемента списку. Якщо парних елементів немає, то результатом повинне бути число 0.

17 Дано список цілих чисел. Знайти номер останнього непарного елемента списку. Якщо непарних елементів немає, то результатом повинне бути число 0.

18 Дано список цілих чисел. Перевірити, чи розташовані елементи списку за збільшенням або за зменшенням.

### 3.4 Зміст звіту

Звіт з лабораторної роботи повинен містити

3.4.1 Тему та мету роботи.

3.4.2 Завдання до лабораторної роботи відповідного варіанту (за списком групи).

3.4.3 Текст програми.

3.4.4 Результати роботи програми.

### 3.5 Контрольні питання

3.5.1 Пояснити чому список є основним типом даних в AutoLisp.

3.5.2 Пояснити, який це тип даних – точкова пара. Для чого використовують такий тип даних?

3.5.3 Перелічити вбудовані функції обробки списків в AutoLisp; їх призначення та форму запису. Навести приклади.

3.5.4 Перелічити вбудовані функції обробки точкових пар в AutoLisp; їх призначення та форму запису. Навести приклади.

## ЛИТЕРАТУРА

1. Кудрявцев Е.М. AutoLISP. Основы программирования в AutoCAD 2000 / Е.М. Кудрявцев.- М.: ДМК Пресс 2000. – 416 с.
2. Маурер У. Введение в программирование на языке ЛИСП / У. Маурер. – М.: Мир, 1978. – 104 С.
3. Полещук Н.Н. AutoLISP и Visual LISP в среде AutoCAD / Н.Н. Полещук, П.В. Лоскутов. – СПб.: БХВ-Петербург, 2006. – 960 с.
4. Хювёнен Э. Мир Лиспа / Э. Хювёнен, Й. Сеппянен . В 2-х т. - М.: Мир, 1990.
5. Хендерсон П. Функциональное программирование: применение и реализация / П. Хендерсон. - М.: Мир, 1983. – 349 С.
6. Бугрименко Г.А. Автолисп - язык графического программирования в системе AutoCAD / Г.А. Бугрименко. - М.: Машиностроение, 1992. - 144 с.
7. Программирование на языке R - ЛИСП/ А.П.Крюков, А.Я.Родионов, А.Ю.Таранов, Е.М.Шаблыгин. - М.:Радио и связь, 1991.

## Додаток А

### Середовище Visual LISP

Середовище Visual LISP вбудовано в систему AutoCAD 2000 для спрощення процесу написання і налагодження програм на мові AutoLISP. На рис. А1 показано діалогове вікно Visual LISP для AutoCAD (Visual LISP for AutoCAD), яке з'являється на екрані, якщо користувач вводить в командному рядку команду VLISP. Таке ж найменування застосовується і в англійській версії AutoCAD. Іншою формою цієї команди є VLIDE. До аналогічного результату приводить використання пункту AutoLISP, Редактор Visual LISP (Visual LISP Editor) падаючого меню Сервіс (Tools).

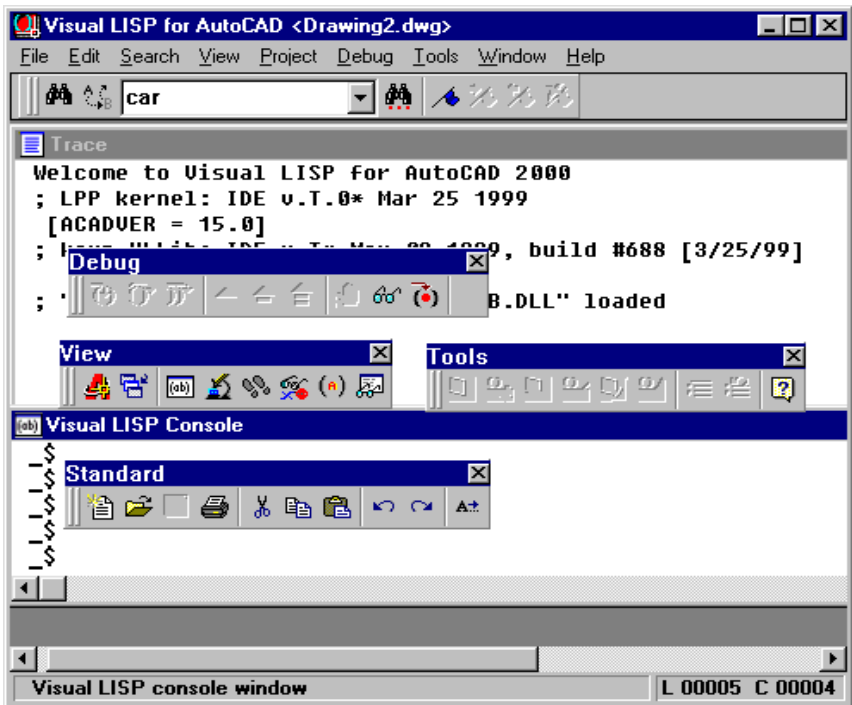


Рисунок А.1 - Плаваюче положення панелей діалогового вікна Visual LISP для AutoCAD

Visual LISP для AutoCAD (Visual LISP for AutoCAD) є самостійною оболонкою усередині системи AutoCAD, зі своїми падаючими меню і панелями інструментів. Панелі інструментів можуть знаходитися не тільки у фіксованому положенні (див. рис. А.1), але і в плаваючому.

Усередині робочої зони розташовуються вікна для відкритих текстових файлів (можуть редагуватися файли LISP-програм, DCL - файли діалогових вікон, SQL-файли операцій з базами даних, файли програм на мові С, вікно Консоль Visual LISP (Visual LISP Console) (використовується для зв'язку з системою AutoCAD), вікно Трасування (Trace) (використовується для налагодження програм).

### Огляд падаючих меню

До складу середовища Visual LISP входять дев'ять падаючих меню: Файл (File), Редагування (Edit), Пошук (Search), Вигляд (View), Проект (Project), Налагодження (Debug), Сервіс (Tools), Вікно (Window), Довідка (Help).

У всіх падаючих меню є постійні пункти, і пункти, які міняють свій стан або зникають залежно від конкретного етапу розробки програми.

**Меню Файл (File)** призначено для виконання операцій з файлами і включає такі пункти:

*Новий файл (New File)*

*Відкрити файл (Open File)*

*Відкрити наново (Reopen)*

*Зберегти (Save)*

*Зберегти як (Save As)*

*Зберегти Все (Save All)*

*Закрити (Close)*

*Повернути (Revert)*

*Закрити все (Close All)*

*Друк (Print)*

*Параметри друку (Print Setup)*

*Зібрати застосунок (Make Application)*



*Завантажити файл (Load File)*  
*Вихід (Exit)*

Призначення пунктів відповідає їх найменуванням. Пункт *Повернути (Revert)* дозволяє відновити вміст файлу в активному вікні, аналогічно пункту *Відкрити наново (Reopen)*.

**Меню Правка (Edit)** містить такі пункти:

Відмінити (Undo)  
 Повторити (Redo)  
 Вирізати (Cut)  
 Копіювати (Copy)  
 Вставити (Paste)  
 Видалити (Delete)  
 Виділити все (Select All)  
 Перевірка дужок (Parentheses Matching)  
 Додаткові функції (Extra Commands)

Перші сім пунктів меню відповідають аналогічним пунктам меню *Правка (Edit)* інших застосунків Windows 95/98. Пункт *Перевірка дужок (Parentheses Matching)* містить підпункти операцій для контролю редагованого тексту:

*Знайти дужку, що закриває вираз (Match Forward)*  
*Знайти дужку, що відкриває вираз (Match Backward)*  
*Виділити до кінця (Select Forward)*  
*Виділити до початку (Select Backward)*

Два перші підпункти дозволяють знаходити в тексті парну дужку, яка закриває (або парну, яка відкриває) для дужки, що знаходиться поряд з поточною позицією курсору. Два останні підпункти виконують аналогічні операції, але при цьому ще і виділяють потрібні ділянки тексту.

**Меню Пошук (Search)** призначено для операцій текстового редактора Visual LISP (зокрема, операцій пошуку і заміни) і включає такі пункти:

*Знайти (Find)*  
*Замінити (Replace)*

*Знайти/замінити далі (Find/Replace Next)*  
*Дописати словом з вікна (Complete Word by Match)*  
*Дописати словом із списку (Complete Word by Apropos)*  
*Закладки (Bookmarks)*  
*Перше повідомлення (First Message)*  
*Наступне повідомлення (Next Message)*  
*Перейти на рядок (Go to Line)*  
*Остання правка (Go to Last Edited)*

Пояснимо призначення деяких з цих пунктів (сене решти пунктів ясний з найменувань). Пункт *Дописати словом з вікна (Complete Word by Match)* дає можливість редакторові Visual LISP дописати (закінчити) почате вами слово аналогічно попередньому слову з таким же початком в поточному вікні редагованої програми. Пункт *Дописати словом із списку (Complete Word by Apropos)* дозволяє отримувати варіанти дописування слова із списку символів AutoLISP, куди входять імена вбудованих функцій (рис. А.2).



Рисунок А.2 - Застосування операції Дописати словом із списку

За допомогою закладок можна позначати потрібні місця в тексті для швидкого переміщення між ними надалі. Операції установки, видалення закладок і переходів між закладками виконуються за допомогою підпунктів *Закладка Вкл/відкл (Toggle Bookmark)*, *Наступна закладка (Next Bookmark)*, *Попередня закладка (Previous Bookmark)*, *Прибрати всі закладки (Clear All Bookmarks)* пункту *Закладки (Bookmarks)*. Ще два підпункти - *Виділити до наступної закладки (Select to Next Bookmark)* і *Виділити до попередньої*

закладки (Select to Previous Bookmark) - того ж пункту меню дозволяють полегшити операції виділення великих ділянок тексту.

**Падаюче меню Вигляд (View)** призначено для операцій переглядання різної допоміжної інформації під час роботи LISP-програми, а також для роботи з панелями інструментів діалогового вікна Visual LISP для AutoCAD (Visual LISP for AutoCAD).

Меню складається з наступних пунктів:

*Оглянути (Inspect)*

*Стек трасування (Trace Stack)*

*Пошук помилок (Error Trace)*

*Інспектор символів (Symbol Service)*

*Вікно контрольних значенні (Watch Window)*

*Параметри довідкового списку (Apropos Window)*

*Вікно точок останову (Breakpoints Window)*

*Вікно виводу (Output Window)*

*Консоль LISP (LISP Console)*

*Огляд бази даних рисунка (Browse Drawing Database)*

*Панелі інструментів (Toolbars)*

Пункт *Оглянути (Inspect)* дозволяє викликати спеціальні вікна, в яких можна подивитись стан змінних, списків, наборів, примітивів, а також виразів AutoLISP. Ім'я об'єкту, що розглядається, або вираз повинні бути заздалегідь виділені в тексті програми; якщо нічого не було виділено, то перед відкриттям чергового вікна *Огляд (Inspect)* видається запит імені. Приклад відкритого вікна *Огляд (Inspect)* приведений на рис. А3, де розглянутий список з характеристиками кола.

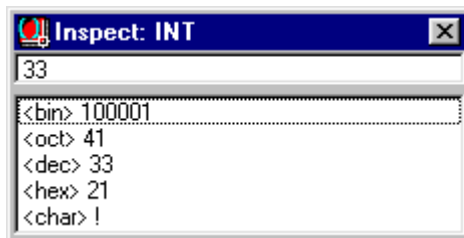


Рисунок А.3 - Вікно Огляд

Пункт *Стек трасування (Trace Stack)* виводить на екран однойменне вікно. У цьому вікні приводиться перелік останніх виконаних операцій (рис. А.4).

У верхній частині вікна розташовані кнопки:

Освіжити (Refresh) - очищає вміст вікна;

Копіювати у вікно трасування/журнал (Copy to Trace/Log) - копіює текст в спеціальне вікно Трасування (Trace).

Якщо у вікні Стек трасування (Trace Stack) виділити який-небудь рядок, то за допомогою правої кнопки миші можна викликати контекстне меню з додатковими операціями над елементами стека трасування (об'єм контекстного меню може мінятися): Огляд (Inspect), Друк (Print), Символ функції (Function Symbol), Копіювати (Copy), Локальні змінні (Local Variables), Положення в тексті (Source Position), Позиція виклику функції (Call Point Source).

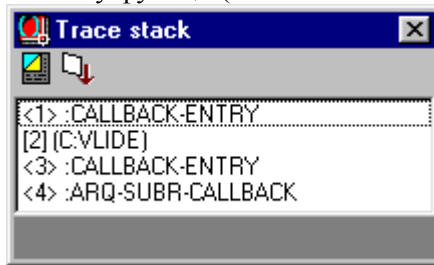


Рисунок А.4 - Вікно Стек трасування

Пункт *Пошук помилок (Error Trace)* падаючого меню Вигляд (View) викликає однойменне вікно (рис. А.15) з інформацією про помилки, що виникли при виконанні програми.

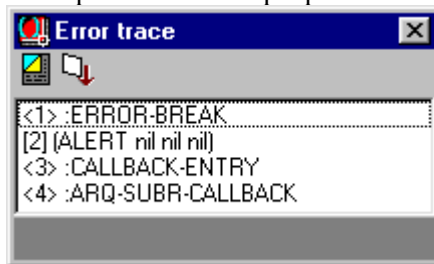


Рисунок А.5 - Вікно Пошук помилок

Якщо виділити в цьому вікні рядок, то натисненням правої кнопки миші можна викликати контекстне меню для отримання додаткових відомостей про елемент вікна Пошук помилок (Error Trace). Контекстне меню має ті ж пункти, що і контекстне меню вікна Стек трасування (Trace Stack). На рис. А.6 показаний приклад додаткового вікна, викликаного за допомогою пункту Показати повідомлення (Show Message), яке з'являється в контекстному меню для рядка SYNTAX-ERROR.

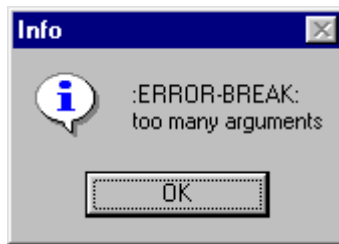


Рисунок А.6 - Вікно “Інфо” з додатковою інформацією про помилку

Пункт *Інспектор символів (Symbol Service)* падаючого меню Вигляд (View) відкриває однойменне вікно (рис. А.7), яке виконує функції, схожі на функції вікна Огляд (Inspect), але тільки для символів AutoLISP.



Рисунок. А.7 - Вікно Інспектор символів

Вікно має три поля: Ім'я (Name), Значення (Value) і Прапорці (Flags). Призначення перших двох полів очевидно (ім'я символу і значення символу), а поле Прапорці (Flags) містить чотири прапорці, які можуть бути встановлені або скинуті для вибраного символу:

Трасування (Trace) - відстежування за допомогою трасування;

Відладка на вході (Debug on Entry) - відладка при вході у функцію;

Захист від привласнення (Protect Assign) - захист спеціальних символів від випадкового привласнення (наприклад, t, pi і ін.);

Експорт в AutoCAD (Export to AutoCAD) - експорт функції в AutoCAD, щоб зробити її доступною для застосунків OBJECTARX.

Вікно Інспектор символів (Symbol Service) у верхній частині має чотири кнопки (зліва направо):

Контрольне значення (Watch) - додає символ до вікна Контрольне значення (Watch);

Оглянути (Inspect) - відкриває вікно Огляд (Inspect) для даного символу;

Показати опис (Show Definition) - викликає вікно текстового редактора і виділяє в ньому текст символу, що є призначеною для користувача функцією;

Довідка (Help) - викликає довідку по Visual LISP з описом символу, що є вбудованою функцією.

*Пункт Вікно контрольних значень (Watch Window)* падаючого меню Вигляд (View) виводить на екран вікно Контрольне значення (Watch) (рис. А.8).

Вікно відображає значення вибраних символів в ході виконання програми. Додавання символів до цього вікна проводиться або за допомогою вікна Інспектор символів (Symbol Service) (кнопкою Контрольне значення (Watch)), або за допомогою першої (лівою) з чотирьох кнопок (Додати контрольне значення (Add Watch)), розташованих у верхній частині вікна Контрольне значення (Watch). Перелічимо ці кнопки:

Додати контрольне значення (Add Watch)

Очистити вікно (Clear Window)

Сортувати вирази (Sort Expressions)

Копіювати у вікно трасування/журнал (Copy to Trace/Log)

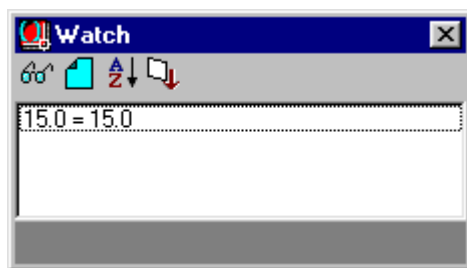


Рисунок А.8 - Вікно Контрольне значення

*Пункт Параметри довідкового списку (Apropos Window)* падаючого меню Вигляд (View) виводить на екран вікно Параметри довідкового списку (Apropos) (рис. А.9).

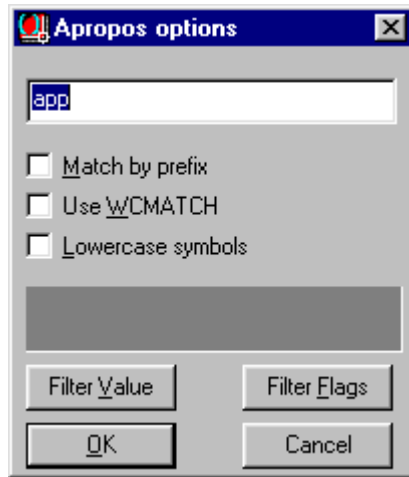


Рисунок А.9 - Вікно Параметри довідкового списку

### Зауваження

Оскільки при роботі в середовищі Visual LISP система AutoCAD жорстко "тримає" розмір вікна, а найменування вікон в російській версії можуть складатися з більшої кількості символів, чим в англійській, то система AutoCAD в цьому випадку може скорочувати найменування вікон, закінчуючи їх багатокрапкою. Наприклад, на рис. А.9 вікно Параметри довідкового списку насправді називатиметься як "Параметри довідкового сп..."

Це вікно дозволяє проглядати список зарезервованих символів (в першу чергу, імен вбудованих функцій). Введена у верхньому полі послідовність букв (на рис. А.9 введена одна буква "a") є основним критерієм формування списку імен, що виводяться (ці букви повинні бути обов'язково присутніми в іменах всіх символів списку). Користувач може за допомогою прапорців вказати, чи повинна ця послідовність обов'язково бути присутньою на початку імені або може зустрітися в іншій частині символу, а також чи можливе використання спеціальних знаків типу "\*" або "#".



Пункт Вікно точок останову (*Breakpoints Window*) падаючого меню Вигляд (*View*) відкриває вікно Точки останову (*Breakpoints*) (рис. А.10), використовуване у відладці програм.

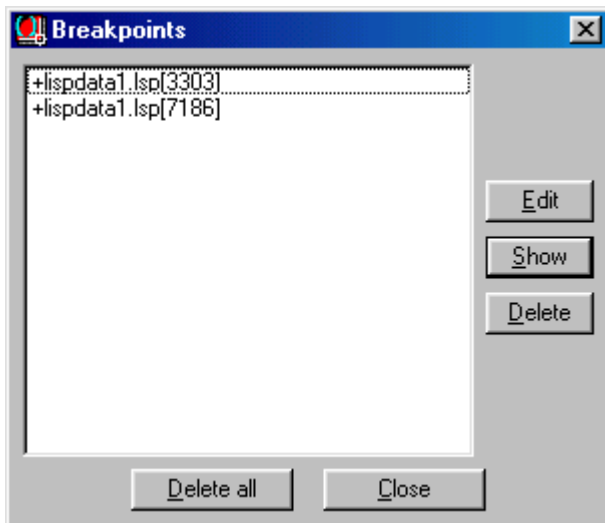


Рисунок А.10 - Вікно Точки останову знайдених символів

Пункт Вікно виводу (*Output Window*) дозволяє відкрити вікно <Повідомлення збірки> (<Builder Output>) для виведення повідомлень, що генеруються в процесі розробки застосунків.

**Пункт Консоль LISP (*LISP Console*) падаючого меню Вигляд (*View*)** викликає дуже важливе вікно, зване Консоль Visual LISP (*Visual LISP Console*) (рис. А.11), яке в середовищі Visual LISP є аналогом зони командних рядків системи AutoCAD. В ньому можна вводити вирази AutoLISP, завантажувати і виконувати функції, а також читати значення змінних. Сюди ж Visual LISP виводить свої повідомлення.

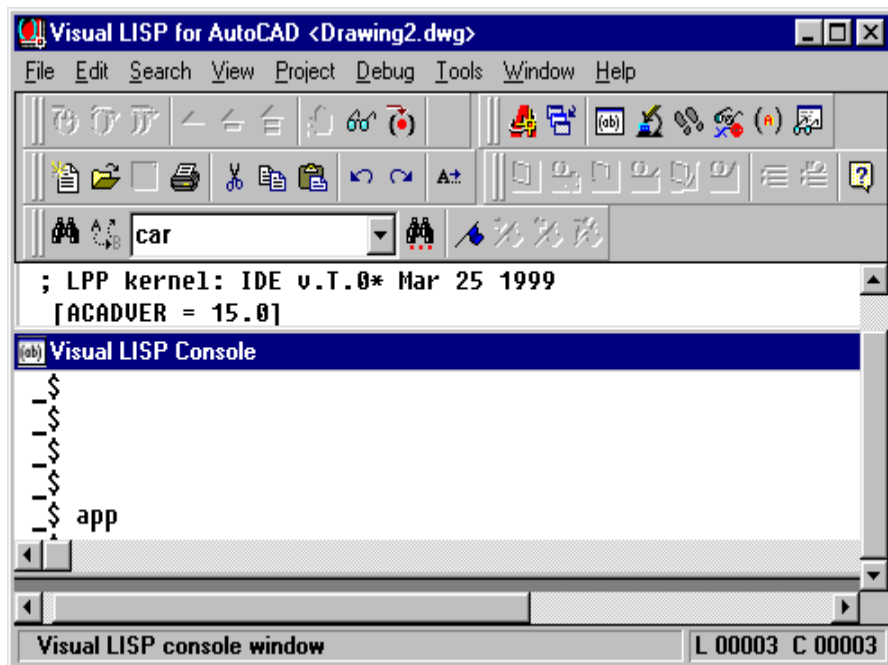


Рисунок А.11 - Вікно Консоль Visual LISP

*Пункт Огляд бази даних рисунка (Browse Drawing Database)* падаючого меню Вигляд (View) має підпункти Всі об'єкти (Browse All Entities), Таблиці (Browse Tables), Блоки (Browse Blocks), Набір (Browse Selection) і Розширені дані (Inquire Extended Data...), при виборі яких викликаються відповідні вікна для проглядання бази даних графічних примітивів і неграфічних об'єктів поточного рисунка.

*Пункт Панелі інструментів (Toolbars)* падаючого меню Вигляд (View) викликає діалогове вікно Панелі інструментів (Toolbars), яке управляє видимістю п'яти панелей середовища Visual LISP.

Прапорці цього вікна повинні бути встановлені біля імен тих з п'яти панелей (Стандартна (Standard), Пошук (Search), Сервіс (Tools), Відладка (Debug), Вигляд (View)), які необхідно вивести на екран. Зміни вносяться за допомогою кнопок Приховати все (Hide All),

Показати все (Show All), Застосувати (Apply). Кнопка Закрити (Close) закриває діалогове вікно Панелі інструментів (Toolbars).

**Падаюче меню Проект (Project)** дає можливість працювати з проектами.

Проект - це об'єднання групи LISP-програм в застосунок, що працює в середовищі системи AutoCAD і компілює в один файл з розширенням fas (можливий варіант, коли кожною з програм створюється свій FAS-файл, хоча це менш зручно). Меню Проект (Project) має такі пункти:

*Новий проект (New Project)*

*Відкрити проект (Open Project)*

*Закрити проект (Close Project)*

*Властивості проекту (Project Properties)*

*Завантажити FAS-файл проекту (Load Project FAS File)*

*Завантажити початкові файли проекту (Load Project Source Files)*

*Зібрати FAS-файл проекту (Build Project FAS)*

*Перекомпілювати FAS-файл проекту (Rebuild Project FAS)*

Останні (додаткові) рядки в меню показують імена відкритих проектів. Активний проект помічений прапорцем.

**Падаюче меню Відладка (Debug)** дозволяє відлажувати програми в середовищі Visual LISP.

Це меню має такі пункти:

*Крок із заходом (Step Into)*

*Крок з обходом (Step Over)*

*Крок з виходом (Step Out)*

*Продовжити (Continue)*

*Скидання до верхнього рівня (Reset to Top Level)*

*Вихід з поточного рівня (Quit Current Level)*

*Додати контрольне значення (Add Watch)*

*Результат останнього обчислення (Watch Last Evaluation)*

*Точка останову Вкл/відкл (Toggle Breakpoint)*

*Видалити всі точки останову (Clear All Breakpoints)*

*Причина останнього останову (Last Break Source)*

*Команда трасування (Trace Command)*

*Останов (Stop Once)*  
*Останов на помилці (Break On Error)*  
*Анімація (Animate)*  
*Перервати обчислення (Abort Evaluation)*

**Падаюче меню Сервіс (Tools)** призначено для допоміжних операцій.

Якщо в середовищі Visual LISP активне вікно Консоль Visual LISP (Visual LISP Console), то меню Сервіс (Tools) має такі пункти:

*Режим AutoCAD (AutoCAD Mode)*  
*Оформлення вікна (Window Attributes)*  
*Параметри середовища (Environment Options)*  
*Зберегти параметри (Save Settings)*

*Пункт Режим AutoCAD (AutoCAD Mode)* включає або відключає режим використання вікна Консоль Visual LISP (Visual LISP Console) як командний рядок системи AutoCAD (з'являється підказка Команда: (Command:)). Це дає можливість вводити команди AutoCAD прямо у вікні консолі. Якщо такий режим встановлений, то в меню зліва від даного пункту стоїть прапорець.

*Пункт Оформлення вікна (Window Attributes)* складається з п'яти підпунктів:

Кольори синтаксичних елементів (Syntax Coloring)  
 Налаштувати поточне (Configure Current)  
 Оформлення поточного вікна за прототипом (Set Current Window to Prototype)  
 Оформлення всіх вікон за прототипом (Set All Windows to Prototype)  
 Шрифти (Fonts)

Підпункт Кольори синтаксичних елементів (Syntax Coloring) доступний тільки в тому випадку, якщо в активному вікні знаходиться текст файлу, який має розширення, відмінне від lsp. Тоді викликається вікно Колірний стиль (Color Style), в якому можна вибрати колірну гамму текстового редактора, що відповідає одному з пропонувананих варіантів.

Підпункт Налаштувати поточне (Configure Current) відкриває вікно Оформлення вікна (Window Attributes), в якому можна підкоригувати колірні установки різних елементів тексту, що діють. Елементи, що оформляються, перелічені в списку, що розкривається.

Підпункт Оформлення поточного вікна за прототипом (Set Current Window to Prototype) міняє установки активного вікна на установки прототипу, а підпункт Оформлення всіх вікон за прототипом (Set All Windows to Prototype) виконує ту ж операцію над всіма вікнами.

Підпункт Шрифти (Fonts) викликає стандартне діалогове вікно Вибір шрифту (Select Font) для вибору використовуваних у вікнах шрифтів.

*Пункт Параметри середовища (Environment Options) падаючого меню Сервіс (Tools) має три підпункти:*

Загальні параметри (General Options)

Параметри форматування Visual LISP (Visual LISP Format Options)

Параметри розмітки сторінки (Page Setup Options)

Підпункт Загальні параметри (General Options) викликає діалогове вікно Загальні параметри (General Options), що має вкладки Загальні (General) і Діагностичні (Diagnostic), які дозволяють міняти різні настройки середовища Visual LISP.

Підпункт Параметри форматування Visual LISP (Visual LISP Format Options) розкриває діалогове вікно Опції форматування (Format options), що управляє установками форматування для текстового редактора Visual LISP.

Третій підпункт - Параметри розмітки сторінки (Page Setup Options) пункту Параметри середовища (Environment Options) викликає діалогове вікно Параметри сторінки (Page Setup) для операцій виведення текстів на друк з середовища Visual LISP.

*Пункт Зберегти параметри (Save Settings) падаючого меню Сервіс (Tools) дозволяє зберігати настройки робочого столу Visual LISP (останні відкриті вікна, їх параметри і т. п.). Можна задати автоматичне збереження настройок, встановивши відповідні прапорці*

в полі Робочий Стіл (Desktop) вкладки Загальні (General) діалогового вікна Загальні параметри (General Options).

Якщо в середовищі Visual LISP активне вікно з текстом редагованої програми, то меню Сервіс (Tools) має такі додаткові пункти:

- Завантажити виділений фрагмент (Load Selection)*
- Завантаження тексту в редактор (Load Text in Editor)*
- Перевірити виділений фрагмент (Check Selection)*
- Перевірити текст в редакторі (Check Text in Editor)*
- Форматувати у фрагменті (Format code in Selection)*
- Форматувати код в редакторі (Format code in Editor)*
- Інструменти інтерфейсу (Interface Tools)*

Перші шість з перелічених пунктів дозволяють завантажувати для виконання, перевіряти або форматувати як текст програми, так і його виділену частину.

Пункт Інструменти інтерфейсу (Interface Tools) має два підпункти:

- Проглядання DCL з фрагмента (Preview DCL in Selection)
- Проглядання DCL з редактора (Preview DCL in Editor)

Ці підпункти дозволяють завантажувати для виконання редагований DCL-файл або його виділену частину.

**Падаюче меню Вікно (Window)** по структурі аналогічно меню з аналогічним найменуванням інших застосунків Windows.

Воно дозволяє управляти взаємним розташуванням декількох одночасно відкритих вікон в середовищі Visual LISP. Пункт Активізувати AutoCAD (Activate AutoCAD) даного меню згортає діалогове вікно Visual LISP для AutoCAD (Visual LISP for AutoCAD) і робить тимчасово доступним основне вікно системи AutoCAD (наприклад, для виконання в рисунку яких-небудь побудов). У нижній частині падаючого меню Вікно (Window) приводиться список відкритих вікон і прапорцем помічено активне вікно.

Останнє **падаюче меню Довідка (Help)** складається з двох пунктів і виконує довідкові функції.

## Огляд панелей інструментів

В оболонку Visual LISP входять п'ять панелей інструментів:

**Стандартна (Standard)**

**Пошук (Search)**

**Сервіс (Tools)** (у плаваючому положенні вона називається Інструменти (Tools))

**Відладка (Debug)**

**Вигляд (View)**

Як видно з даного переліку, найменування панелей схожі на найменування розглянутих вище падаючих меню. Призначення панелей також аналогічно призначенню відповідних падаючих меню.

У панель інструментів **Стандартна (Standard)** (рис. А.12) входять такі кнопки (зліва направо):

*Новий файл (New file)*

*Відкрити файл (Open file)*

*Зберегти файл (Save file)*

*Друк (Print)*

*Виізати (Cut)*

*Копіювати (Сміття)*

*Вставити (Paste)*

*Відмінити (Undo)*

*Повторити (Redo)*

*Дописати слово (Complete word)*



Рисунок А.12 - Панель інструментів Стандартна

Дев'ять перших кнопок цієї панелі аналогічні однойменним кнопкам панелі інструментів Стандартна (Standard) системи AutoCAD. Остання кнопка Дописати слово (Complete word) відповідає пункту

Дописати слово із списку (Complete Word by Apropos) падаючого меню Пошук (Search), що дозволяє закінчити почате слово елементом списку стандартних символів.

Панель інструментів **Пошук (Search)** виконує функції, аналогічні функціям падаючого меню Пошук (Search).

У неї входять сім наступних кнопок:

*Знайти (Find)*

*Замінити (Replace)*

*Пошук рядка на панелі інструментів (Find toolbar string)*

*Закладка Вкл/відкл (Toggle bookmark)*

*Наступна закладка (Next bookmark)*

*Попередня закладка (Previous bookmark)*

*Видалення всіх закладок (Clear all bookmarks)*

Призначення всіх цих кнопок, окрім третьої, аналогічно призначенню відповідних пунктів і підпунктів падаючого меню Пошук (Search). Кнопка Пошук рядка на панелі інструментів (Find toolbar string) працює разом із списком, що розкривається, знаходиться в панелі інструментів Пошук (Search) і виконує пошук символу, ім'я якого вибрано в цьому списку або введено в список з клавіатури.

Панель інструментів **Сервіс (Tools)** (рис. А.13) має дев'ять кнопок:

*Завантаження активного вікна редагування (Load active edit window)*

*Завантаження виділеного фрагмента (Load selection)*

*Перевірка вмісту вікна редактора (Check edit window)*

*Перевірка виділеного фрагмента (Check selection)*

*Форматування вмісту вікна редагування (Format edit window)*

*Форматування виділеного фрагмента (Format selection)*

*Закоментувати блок (Comment block)*

*Розкоментувати блок (Uncomment block)*

*Довідка (Help)*

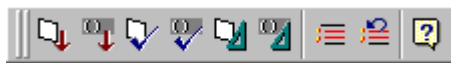


Рисунок А.13 - Панель інструментів Сервіс



Перші шість пунктів панелі аналогічні першим шести пунктам падаючого меню Сервіс (Tools), якщо користувач за допомогою редактора працює з текстом програми або її фрагментом.

Пункти Закоментувати блок (Comment block) і Розкоментувати блок (Uncomment block) дозволяють легко вставляти (за допомогою крапок з комою) або знімати коментарі у виділених рядках тексту програми. Пункт Довідка (Help) служить для отримання довідкової інформації.

Панель інструментів **Відладка (Debug)** (рис. А.14) має такі кнопки:

*Крок із заходом (Step into)*

*Крок з обходом (Step over)*

*Крок з виходом (Step out)*

*Продовжити (Continue)*

*Вийти (Quit)*

*Скидання (Reset)*

*Точка останову Вкл/відкл (Toggle breakpoint)*

*Додати контрольне значення (Add watch)*

*Останній останов (Last break)*



Рисунок А.14 - Панель інструментів Відладка

Крім того, в процесі виконання програми в налагоджувальному режимі, з точками остановів, в панелі з'являється остання кнопка (на рис. 12.42 вона показана, але в звичайному режимі ця кнопка не відображається). Кнопка показує, де знаходиться курсор під час поточного останову (перед дужками або після).

У панель інструментів **Вигляд (View)** (рис. А.15) входять такі кнопки:

- Активізація AutoCAD (Activate AutoCAD)*
- Вибрати вікно (Select window)*
- Консоль Visual LISP (Visual LISP Console)*
- Вивчити (Inspect)*
- Трасування (Trace)*
- Вікно відладки символів (Symbol service)*
- Довідковий список (Apropos)*
- Вікно контрольних значень (Watch window)*



Рисунок А.15 - Панель інструментів Вигляд

Найменування кнопок говорять про те, що вони виконують операції, закріплені за розглянутими вище різними пунктами падаючих меню: Режим AutoCAD (AutoCAD Mode) меню Сервіс (Tools), вибір вікна за допомогою списку вікон меню Вікно (Window), Консоль LISP (LISP Console) меню Вигляд (View), Оглянути (Inspect) меню Вигляд (View), Стек трасування (Trace Stack) меню Вигляд (View), Інспектор символів (Symbol Service) меню Вигляд (View), Параметри довідкового списку (Apropos Window) меню Вигляд (View), Вікно контрольних значень (Watch Window) меню Вигляд (View).

## **Написання програм в текстовому редакторві середовища Visual LISP**

У середовищі Visual LISP можна відкривати кожний в своєму вікні і редагувати текстові файли таких типів: LISP-програми (з розширенням lsp), DCL-файли діалогових вікон (з розширенням del), SQL-файли операцій з базами даних (з розширенням sql), файли програм на мові C++ (з розширенням c). Включений до складу

середовища текстовий редактор надає суттєву допомогу в написанні файлів цих типів, оскільки вже в процесі введення тексту виконує синтаксичний аналіз, виділяючи символи, відповідні дужкам, операторам, операндам, коментарям і так далі. Результати свого синтаксичного аналізу редактор відображає кольором букв набраного тексту. Вибір настройки редактора відповідно до типу файлу можна зробити за допомогою пункту *Оформлення вікна (Window Attributes)*, *Кольори синтаксичних елементів (Syntax Coloring)* падаючого меню Сервіс (Tools). За умовчанням редактор налаштований на роботу з текстами LISP - програм.

Колірна гамма, зазвичай використовувана для синтаксичного аналізу програм на мові LISP, наступна:

синій - зарезервовані символи AutoLISP (імена вбудованих функцій і т. п.);

фіолетовий - текстові рядки, обрамлені подвійними лапками;

зелений - цілі і речові числа;

фіолетовий (на сірому фоні) - коментарі;

червоний - відкриваючі і закриваючі дужки;

чорний - інші символи (змінні користувача та ін.).

За бажанням можна внести зміни до колірної палітри за допомогою пункту *Оформлення вікна (Window Attributes)*, *Налагодити поточне (Configure Current)* падаючого меню Сервіс (Tools).

*Пункт Параметри середовища (Environment Options)* падаючого меню Сервіс (Tools) теж може допомогти у встановленні великої кількості зручностей для конкретного користувача.

Незамінним засобом швидкого коментування виділеної ділянки тексту і зняття коментарів є кнопки Закоментувати блок (Comment block) і Розкоментувати блок (Uncomment block) панелі інструментів Сервіс (Tools).

Синтаксична перевірка тексту всього відкритого файлу або виділеного фрагмента може бути виконана за допомогою пунктів *Перевірити виділений фрагмент (Check Selection)* і *Перевірити текст в редакторі (Check Text in Editor)* падаючого меню Сервіс (Tools)

або аналогічних кнопок в панелі Сервіс (Tools). Для перевірки відповідності закриваючих і відкриваючих дужок добре користуватися пунктом *Перевірка дужок (Parentheses Matching)* падаючого меню Правка (Edit).

При перезапису файлу з розширенням `lsp` Visual LISP створює копію попереднього стану файлу (копії привласнюється розширення `_ls`).

## Додаток Б

### Розробка застосунків

За допомогою Visual LISP зручно писати не окремі програми, а цілі застосунки. Застосунки тільки з LSP-файлів можна створювати за допомогою механізму проєктів. Якщо в застосунок необхідно включити DCL-файли діалогових вікон або допоміжні текстові файли, то можна користуватися інструментом розробки складних застосунків.

### Засоби наладки програм

Спеціальне вікно Консоль Visual LISP (Visual LISP Console) замінює користувачеві в середовищі Visual LISP зону командних рядків системи AutoCAD. З консолі можна вводити вирази AutoLISP, завантажувати і виконувати функції, в нього ж виводяться повідомлення Visual LISP і системи AutoCAD.

Для завантаження LISP-програми (аналогічно завантаженню файлу за допомогою функції load мови AutoLISP), що знаходиться у відкритому вікні текстового редактора, слід застосувати пункт *Завантаження тексту в редактор (Load Text in Editor)* падаючого меню Сервіс (Tools) або кнопку Завантаження активного вікна редагування (Load active edit window) панелі інструментів Сервіс (Tools). На рис. Б.1 показано повідомлення, що генерується Visual LISP, про завантаження файлу.

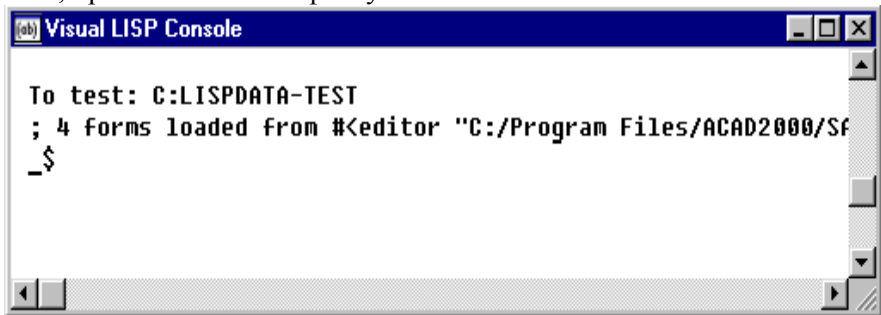
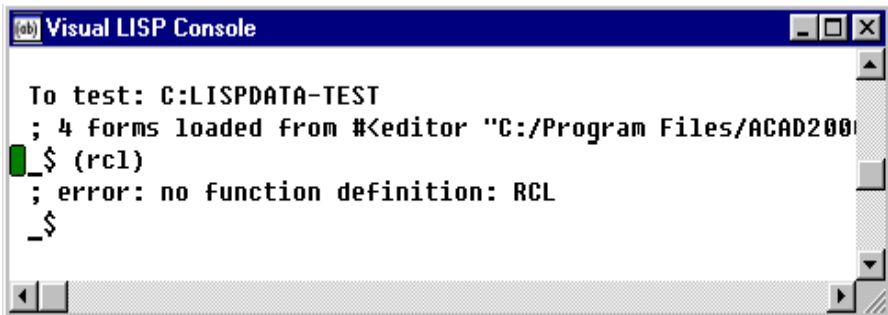


Рисунок Б.1 - Повідомлення про завантаження файлу у вікні Консоль Visual LISP

Можливе покрокове виконання програми, коли користувач по черзі виділяє і завантажує фрагменти тексту (використовуються пункт *Завантажити виділений фрагмент (Load Selection)* падаючого меню Сервіс (Tools) або відповідна кнопка панелі інструментів Сервіс (Tools)).

У вікні Консоль Visual LISP (Visual LISP Console) як запрошення користувача до введення застосовуються символи "\_\$". Так, наприклад, якщо на попередньому кроці користувач завантажив файл rcl.lsp з однойменною програмою, то він може виконати її, ввівши в консолі "(rcl)". Інтерпретатор Visual LISP виконає програму і, якщо в ній немає помилок, виведе результат на консоль (рис. Б.2).



```

Visual LISP Console
To test: C:LISPDATA-TEST
; 4 forms loaded from #<editor "C:/Program Files/ACAD2000
_$ (rcl)
; error: no function definition: RCL
_$

```

Рисунок Б.2 - Повідомлення про виконання програми у вікні Консоль Visual LISP

Прямо з консолі можна читати значення глобальних змінних. Для цього потрібно у відповідь на запрошення ввести з консолі ім'я змінної. Visual LISP виведе її поточне значення (на. рис. Б.3 показано значення змінної list\_rad).

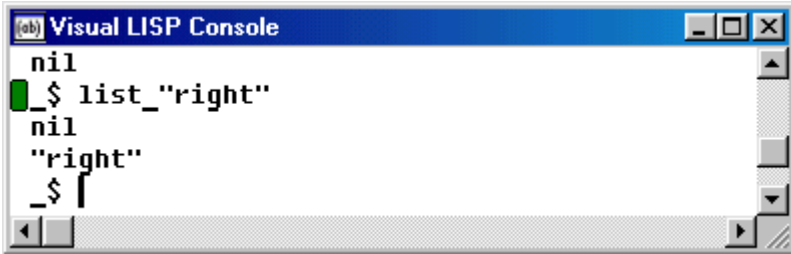


Рисунок Б.3 - Читання значення глобальної змінної у вікні Консоль Visual LISP

### Зауваження

У середовищі Visual LISP при читанні значення змінної не потрібне введення перед ім'ям змінної знаку оклику, як це необхідно в системі AutoCAD.

При наладці програми зручно стежити за значеннями вибраних змінних. Для цього використовуються пункти *Оглянути (Inspect)*, *Інспектор символів (Symbol Service)*, *Вікно контрольних символів (Watch Window)*, розглянуті при вивченні падаючого меню *Вигляд (View)*. Виклик за допомогою цих пунктів відповідних вікон показує поточні значення вибраних змінних або виразів. Вміст цих вікон може бути скопійований у вікно *Трасування (Trace)* для подальшого узагальнення.

За наявності помилок слід також користуватися пунктами *Стек трасування (Trace Stack)* і *Пошук помилок (Error Trace)*, що виводять відповідні однойменні вікна.

Дієвим засобом відладки є розстановка в програмі точок її переривання (останову). Це виконується за допомогою пункту *Точка останову Вкл/відкл (Toggle Breakpoint)* падаючого меню *Відладка (Debug)* або за допомогою кнопки *Крапка останову Вкл/відкл (Toggle breakpoint)* панелі інструментів *Відладка (Debug)*. В цьому випадку при виконанні програма зупиняється, дійшовши до точки переривання, і чекає вказівки від користувача про можливий варіант продовження роботи. Для продовження є пункти *Крок із заходом (Step*

*Into*), *Крок з обходом (Step Over)*, *Крок з виходом (Step Out)*, *Продовжити (Continue)*, *Скидання до верхнього рівня (Reset to Top Level)*, *Вихід з поточного рівня (Quit Current Level)* падаючого меню Відладка (Debug) або аналогічні ним кнопки панелі інструментів Відладка (Debug).

Особливо ефективне поєднання точок переривання з відкритим вікном контрольних значень. Виконуючи свою програму по частинах, ви стежитимете за тим, як при цьому змінюються значення контрольних змінних.

*Пункт Анімація (Animation)* падаючого меню Відладка (Debug) дає можливість стежити за ходом виконання програми по початковому тексту, оскільки Visual LISP по черзі виділяє ті вирази програми, які обробляються у нинішній момент часу.

## Проекти

Декілька LISP-програм, що виконуються спільно, можна об'єднувати в проекти. Кожен такий проект є самостійним застосунком, який може працювати в початкових текстах або у файлах, що компілюють, з розширенням fas. Крім того, проект можна скомпілювати в єдиний FAS-файл, що дасть переваги за швидкістю завантаження і виконання.

Операції з проектами зібрані в падаюче меню Проект (Project). Для створення нового проекту використовується пункт *Новий проект (New Project)*, який відкриває діалогове вікно Новий проект... (New Project...) (рис. Б.4).



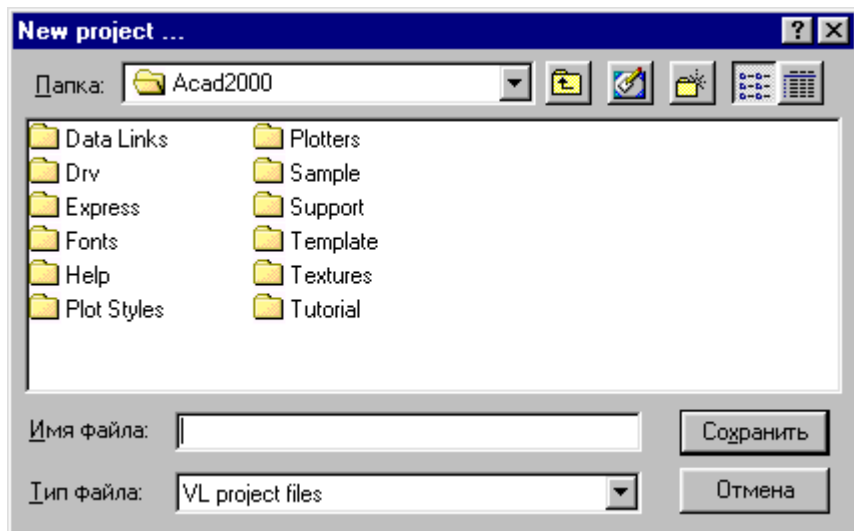


Рисунок Б.4 - Діалогове вікно Новий проект...

У цьому вікні потрібно ввести ім'я файлу даними про проект. Файл утворюється автоматично, в нього заноситься інформація про вхідні в нього програми і режими компіляції. Розширення файлу проекту prj. Після введення імені проекту з'являється вікно Властивості проекту (Object Properties) (рис. Б.5).

Діалогове вікно має дві вкладки: Файли проекту (Project Files) і Параметри збірки (Build Options). У першій з них потрібно в полі Шукати в (Look in) вибрати теку з LSP-файлами і в списку вікна виявлених файлів, що з'явився, за допомогою кнопки > перенести потрібні імена в праву частину. Якщо файли проекту знаходяться в різних теках, то можна поміняти ім'я теки і вибрати з неї наступну частину файлів, що включаються. Для файлів, включених з різних тек, за допомогою натиску правої кнопки миші на імені файлу можна викликати контекстне меню, де при виборі пункту Імена і розміри файлів (Log filenames and size) виводиться в нижню частину діалогового вікна для кожного виділеного імені його повне ім'я і розмір (рис. Б.6).

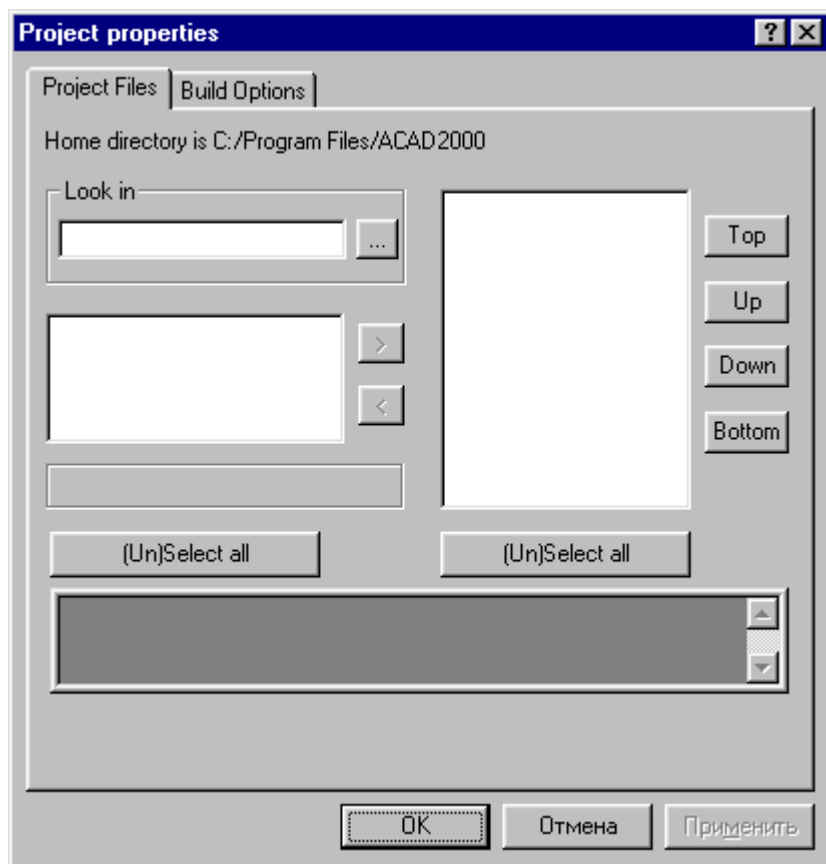


Рисунок Б.5 - Діалогове вікно Властивості проекту, вкладка Файли проекту

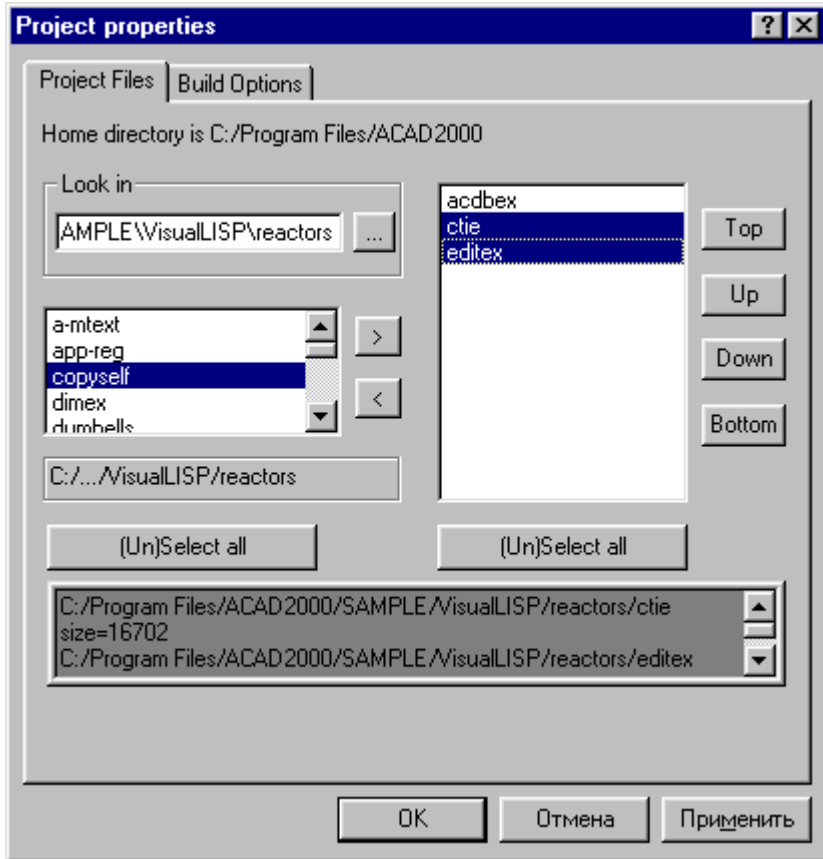


Рисунок Б.6 - Повні імена і розміри файлів, включених в проект

Після закінчення формування повного списку компонент нового проекту можна їх упорядкувати за допомогою кнопок Початок (Top), Вгору (Up), Вниз (Down), Нижня (Bottom). Найчастіше таке впорядкування не потрібне, але іноді воно потрібне (наприклад, якщо є спеціальні файли ініціалізації, то їх треба винести вгору, щоб вони завантажувалися першими).

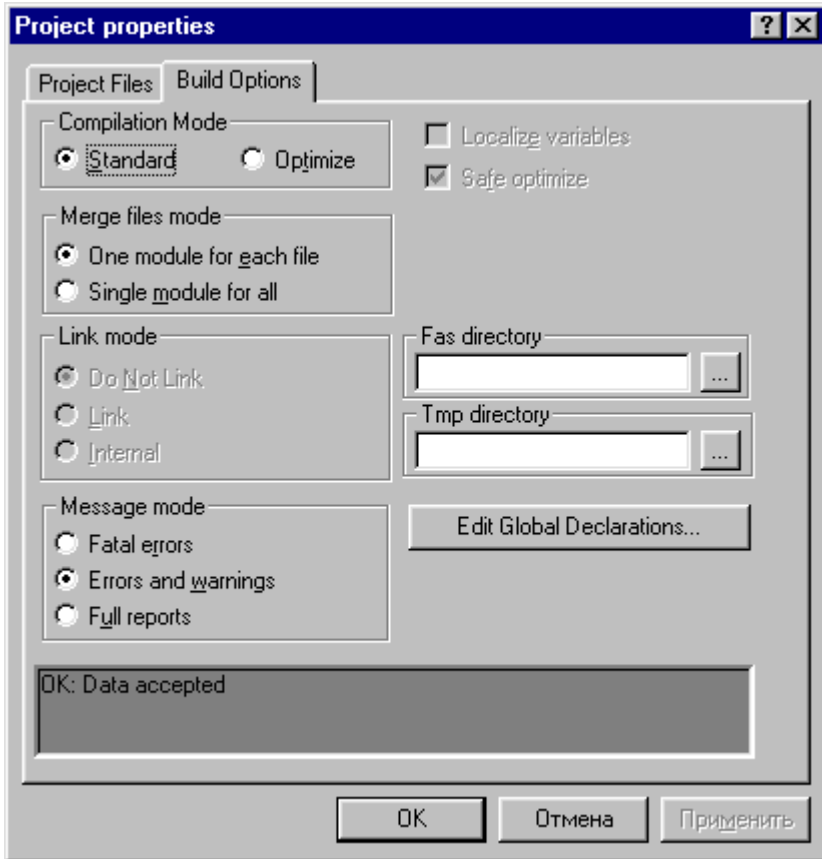


Рисунок Б.7 - Діалогове вікно Властивості проекту, вкладка Параметри збірки

Вкладка Параметри збірки (Build Options) вікна Властивості проекту (Object Properties) (рис. Б.7) встановлює режими компіляції і зв'язку. Зокрема, поле Режим злиття файлів (Merge files mode) встановлює, чи будуть FAS - файли створюватися для кожного LSP - файлу окремо або проект компілюватиметься в єдиний файл (FAS - застосунок). Тут же можна задати теки для розміщення FAS-файлів і тимчасових файлів.

Закінчення роботи у вікні завершується натиском кнопки Застосувати (Apply). Кнопка ОК (OK) закриває діалогове вікно, після чого на екрані з'являється вікно відкритого проекту. На рис. Б.8 показаний приклад проекту *newp* з трьох файлів.



Рисунок Б.8 - Вікно проекту

У верхній частині цього вікна є п'ять кнопок операцій з проектом (зліва направо):

*Властивості проекту (Project Properties)*

*Завантаження FAS-файлов проекту (Load Project FAS)*

*Завантаження початкових файлів (Load Source Files)*

*Збірка FAS-файлу проекту (Build Project FAS)*

*Перекомпіляція FAS-файлу проекту (Rebuild Project FAS)*

Найменування кнопок аналогічні найменуванням пунктів падаючого меню *Проект (Project)*. У створені раніше проекти можна вносити зміни, їх можна завантажувати як у компільованому вигляді, так і вигляді початкових текстів.