

Программирование в **AutoCAD**



Лекция **2**

Функция связи с **AutoCAD**

Функция выполняет команды AutoCAD
(command [аргум1]... [аргумN]) → nil

Аргументы

- Имена команд и опций – STR
- Координаты точек – LIST
- Длины, расстояния – REAL
- "" - <Enter>
- PAUSE

Функция связи с **AutoCAD**

```
(setq y "0,0")
```

```
(command "_line" "10,10" y '(10 15) "")
```

```
(command "_break" pause "_f" pause "@")
```

```
(command "_rectang" pause "@15,25")
```

```
(setq x 10.0 y 3)
```

```
(command "_circle" pause (* x y))
```

Функция связи с **AutoCAD**



В функцию **Command** не могут
быть вложены функции
семейства **Get**,
предназначенные для
интерактивного ввода данных,
так как результат операции не
предсказуем.

Функции обработки списков

- $(\text{Car список}) \rightarrow$ первый элемент списка

$(\text{Car `(x y z)}) \rightarrow x$

$(\text{Car `(a (a1 a2))}) \rightarrow a$

$(\text{Car `((a1 a2) a)}) \rightarrow (a1 a2)$

$(\text{Car ()}) \rightarrow \text{nil}$

$(\text{Car nil}) \rightarrow \text{nil}$

$(\text{Car 10.3}) \rightarrow$

Функции обработки списков

- (Cdr список) → список без первого элемента

(Cdr `(10 20 30)) → (20 30)

(Cdr `((a1 a2) a)) → (a)

(Cdr `(a (a1 a2))) → ((a1 a2))

(Cdr `(a)) → nil

(Cdr ()) → nil

Функции обработки списков

- (C...R список) - композиция функций Car и Cdr
- Порядок применения функций: справа налево
- a – Car, d – Cdr

$(\text{Cadr } '(x\ y\ z)) \rightarrow y$ $(\text{Car } (\text{Cdr } '(x\ y\ z)))$

$(\text{Cadr } '(a\ (a1\ a2))) \rightarrow (a1\ a2)$

$(\text{Caddr } '(x\ y\ z)) \rightarrow z$

Функции обработки списков

- $(\text{Cons НПЭ список}) \rightarrow$ список,
дополненный новым первым элементом
- $(\text{Cons } (a1\ b1)\ (a2\ b2)) \rightarrow ((a1\ b1)\ a2\ b2)$
- $(\text{Setq } a\ 5)\ (\text{Cons } a\ '(+ 7)) \rightarrow (5\ +\ 7)$
- $(\text{Setq } a\ 5.0)\ (\text{Cons } a\ (+\ 7.0\ 2.0)) \rightarrow$
 $(5.0\ .\ 9.0)$ – точечная пара
- $(\text{Car } '(a\ .\ b)) \rightarrow a$ $(\text{Cdr } '(a\ .\ b)) \rightarrow b$

Функции обработки списков

- (Nth n-1 список) → n-ный элемент списка

(Nth 0 '(x y z)) → x

(Nth 1 '(x y z)) → y

(Nth 2 '(x y z)) → z

(Setq pt '(10 20 30))

(Nth 0 pt) → 10

Формирование списков

- (List S-выраж1... S-выражN) → список
- (Setq x 10.0 y 20.0 z 30.0)
- (List x y z) → (10.0 20.0 30.0)
- (Setq a `(a1 a2) b (b1 b2))
- (List a b) → ((a1 a2) (b1 b2))
- (List `a b) → (a (b1 b2))
- (List (Car a) (Cadr b)) → (a1 b2)

Функции обработки списков



- (Append список1... списокN) → объединение нескольких списков в один
- (Last список) → выделение последнего элемента списка
- (Revers список) → запись списка в обратном порядке
- (Length список) → определение количества элементов списка

Функция **Atom**

- (Atom S-выраж) → t или nil

(Setq a '(x y z) a – список, b – ссылка на a
b 'a)

(Atom a) → nil

(Atom b) → t

(Atom 'a) → t

(Atom 'b) → t

Функция **Eq**-физическое равенство

- (Eq S-выраж S-выраж) → t или nil

Проверяет равенство указателей на S-выраж

```
(Setq f1 '(x y z)
      f2 '(x y z)
      f3 f2)
```

```
(Eq f1 f2) → nil
```

```
(Eq f2 f3) → t
```

Функции ввода данных

- Создают паузу для интерактивного ввода информации
 - Подсказка – текст в командной строке
 - Point – координата точки, от которой тянется резиновая нить
 - (Getdist [point] [подсказка]) – ввод расстояния (Real)
- (Setq rad1 (getdist '(0.0 10.0 15.0) "\nВведите значение радиуса"))

Функции ввода данных

- (Getangle [point] [подсказка]) ввод значения угла в радианах (Real)
Point – двухмерная точка в рабочей плоскости
- (Getcorner point [подсказка]) ввод двухмерной координаты точки (List)
Point – двухмерная точка в рабочей плоскости
- (Getint [подсказка]) ввод целого числа (Int)
(Setq num (getint "\nВведите число отверстий"))

Функции ввода данных



- (Getpoint [point] [подсказка]) ввод координаты точки (List)
- (Getreal [подсказка]) ввод действительного числа (Real)
- (Getstring [F] [подсказка]) ввод строки текста (Str)
F – возможность разделить слова пробелами

Геометрические функции

- (Distance point1 point2) – расстояние между двумя точками (Real)
(Distance `(1.0 2.5 3.0) `(7.7 2.5 3.0)) → 6.7
- (Angle point1 point2]) - угол поворота отрезка или его проекции на плоскость XY относительно оси OX в радианах (Real)
- (Inters point1 point2 point3 point4) - точка пересечения двух отрезков (List)

Геометрические функции

- (Polar pt a d) – координата конца вектора, который выходит из точки pt, имеет длину d и составляет угол a (в радианах) с осью OX (LIST).

```
(Setq pt1 (polar '(1.0 1.0 3.5) 0.79 1.414))  
→ (2.0 2.0 3.5)
```

Функция для определения пользовательской функции

- (Defun Имя ([fp1...fprn] [/ lр1...lрm])
Тело функции)
- Имя- название функции
с:имя- новая команда AutoCAD
- fp1...fprn – формальные параметры
- lр1...lрm – локальные параметры
- Тело функции – блок S-выражений

Функция для определения пользовательской функции

- (defun c:breakat ()
(command "_break" pause "f" "@"))
- Расширение файла с LISP – программой
- .lsp
- (Load "имя файла") – загрузка файла
(Load "test1")
- Результат – имя последней функции
или имя файла (при ошибке ввода)

Ввод комментария



- `;;;` - выравнивание комментария по левому краю
- `;;` - уровень вторых скобок
- `;` - сдвиг на 40 символов
- `;` | `|;` - внутренний комментарий

Функции работы с примитивами

- (Entlast) → имя последнего созданного примитива
(Setq E (entlast))
- (Entnext) → имя первого неудаленного примитива
- (Entnext ИП) → имя примитива, следующего за примитивом с именем ИП, в базе данных

Функции работы с примитивами

- (Entsel [подсказка]) - служит для указания примитива
- Если нет подсказки, выводится запрос **Select object:**
- Возвращает список (A Pt)

A – имя указанного примитива в базе данных (ENAME)

Pt – координата точки указания (LIST)

(Setq E (entsel "укажите объект"))

 NameP (nth 0 E)

 Pt (nth 1 E))

Логические функции



- =
- /=
- <
- <=
- >
- >=
- And
- Or
- Not
- Null

Функции организации циклов

- (Repeat K S-выр1... S-вырN)
- Выполняется каждое S-выражение K раз
- K имеет тип INT
- Результат – значение последнего S-выражения
- (Setq n 5 r 10.0)
(Repeat n (Command "_circle" "10,20"
(Setq r (+ r 10.0)))) → nil

Функции организации циклов

- (While условие S-выр1... S-вырN)
- Выполняется каждое S-выражение пока условие истинно (= t)
- Результат – значение последнего S-выражения
- (Setq k 0 m 4)
(While (<= k m) (* (setq k (1+ k)) m)) → 20
- (Setq i 1 factorial 1)
(While (< i n)
(setq i (1+ i) factorial (* factorial i)))

Функции организации разветвлений

- (If условие S-выр-тогда S-выр-иначе)
- Результат – значение последнего S-выражения
- (If (> a 0) (Command "_circle" "0,0" a)
(Prin1 "Неверное значение радиуса"))
- (Setq a (Getdist "Введите радиус"))
(If (> a 0)
(Progn (setq co (Getpoint)) (Command "_circle"
co a))
(Prin1 "Неверное значение радиуса"))

Программа моделирования основы (1)



```
(defun c:osnova ( / h1 h2 bt bt1 c1 c2 v1 r1 r2
  d1 d2 x1 y1 z1)
  (setq bt (getpoint "\nВведите базовую точку")
    r1 (getdist bt "\nВведите радиус фланца")
    h1 (getdist bt "\nВведите высоту фланца"))
  (command "cylinder" bt r1 h1))
```

Программа моделирования основы (2)



```
(setq c1 (entlast)
  x1 (nth 0 bt)
  y1 (nth 1 bt)
  z1 (nth 2 bt)
  bt1 (list x1 y1 (+ z1 h1))
  r2 (getdist bt1 "\nВведите радиус цилиндра")
  h2 (getdist bt1 "\nВведите высоту цилиндра")
  d1 (/ r2 10.0)
  d2 (- h2 d1)
  bt2 (list x1 y1 (+ z1 h1 d2)))
```


Программа моделирования основы (3)



```
(command "cylinder" bt1 r2 d2 )
```

```
(setq c2 (entlast))
```

```
(command "circle" bt2 r2 "extrude" (entlast)  
  "" d1 45 "union" c1 c2 (entlast) ""))
```



```
( defun c:masotv ( / rotv hotv rmas bt n n1 k x1
  y1 z1 e1)
(while (null (setq e1 (entsel "\nУкажите
обрабатываемую деталь"))))
(setq bt (getpoint "\nВведите базовую точку")
  e1 (nth 0 e1)
  n (getint "\nВведите количество отверстий")
  rmas (getdist bt "\nВведите радиус
массива")
  n1 (1- n)
  x1 (nth 0 bt)
  y1 (nth 1 bt)
```



