AutoLISP (1)

Pripremio Dragan Cvetković

AutoLISP je poseban deo LISP programskog jezika, koji se isporučuje unutar programskog paketa za projektovanje AutoCAD. AutoLISP omogućava korisnicima AutoCAD-a da pišu i kreiraju makroe i funkcije u višem programskom jeziku, što je izuzetno povoljno za grafičke aplikacije. LISP je lagan za učenje i korišćenje i veoma je fleksibilan.

LISP je predstavnik aplikativnih ili funkcijskih programskih jezika. U okviru njega ne postoje komande, već se sve svodi na izračunavanje raznih funkcija. Evo jednostavnog primera. Ako treba, u fortranu, da se napiše program koji izračunava zbir dva broja, onda je njegova sintaksa:

```
read*, x,y
z=x+y
print*, z
```

Kada treba da se izračuna zbir dva broja, startuje se program, unose se vrednosti za *x* i *y* promenljive i na ekranu će se pojaviti traženi rezultat.

U AutoLISP-u je procedura drugačija. Najpre, interpreteru LISP-a treba definisati novu funkciju (na primer, pod nazivom ZBIR):

```
(DEFUN ZBIR (X Y)
          (+ X Y))
```

Definiše se funkcija pod nazivom ZBIR, koja ima dva argumenta i čija je vrednost zbir navedenih argumenata, što je definisano korišćenjem operatora "+" iz LISP interpretera.

Kada se prethodni tekst otkuca LISP interpreteru, on je time "naučio" da računa funkciju ZBIR i ona je sada ravnopravna sa njegovim ostalim funkcijama. To znači da može da se koristi za dalja izračunavanja ili prilikom definisanja novih funkcija. Na primer, ako treba da se izračuna 4+5, otkucaće se:

```
Command: (ZBIR 4 5)
```

i interpreter će izračunati vrednost funkcije koja je zadata i ispisaće rezultat 9 u komandnoj liniji (oblasti) AutoCAD-a.

Leksička konvencija

Ulaz AutoLISP-a može da se unese sa tastature u okviru komandne linije AutoCAD-a, može da se učita iz ASCII fajla ili iz string varijable. Prilikom definisanja ulaza moraju se poštovati sledeća pravila:

- sledeći karakteri ne mogu da učestvuju u definisanju naziva simbola ili numeričkih konstanti
  ( ) , " ; (blanko) (kraj linije)
- višestruki blanko (prazan prostor) između karaktera je isto što i jedan blanko;
- celobrojne (engleski *integer*) konstante mogu da počinju sa "+" i "-" opcionim karakterima i njihov opseg je od –32768 do +32767;
- realne konstante se sastoje od jedne ili više cifara za kojima sledi decimalna tačka, a iza nje slede decimalne cifre (".5" se ne prepoznaje kao korektna vrednost, tj. kao realan broj, nego je "0.5" korektna vrednost);
- sledeći kôdovi predstavljaju: \\ je *backslash* (\), \e simulira pritisak na taster **Esc**, \n predstavlja oznaku za novu liniju, \r simulira pritisak na taster **Enter**, \t predstavlja pritisak na taster **Tab** i \nnn predstavlja karakter čiji je oktalni kôd *nnn*.

Tipovi podataka u AutoLISP-u

AutoLISP podržava nekoliko vrsta podataka, i to: liste, simbole, tekstualne stringove, realne i *integer* brojeve, opise fajlova, AutoCAD-ove nazive fajlova, AutoCAD-ove selekcione setove, kao i odgovarajuće funkcije koje omogućavaju bazično programiranje za dvodimenzionalne i trodimenzionalne grafičke aplikacije. Da bi se dobile grafičke koordinate, usvojena je sledeća konvencija za 2D i 3D tačke:

- **2D tačke** – predstavlja listu od dva realna broja (X Y), kao (2.34 4.56), gde je prvi broj vrednost X koordinate, a drugi broj predstavlja Y koordinatu;
- **3D tačke** – predstavlja listu od tri realna broja (X Y Z), kao (4.56 6.76 3.66), gde je prvi broj vrednost X koordinate, drugi broj je vrednost Y koordinate, a treći broj predstavlja Z koordinatu.

Učitavanje programa

Neke rutine se automatski učitavaju kada se izaberu (markiraju) iz standardnih menija AutoCAD-a, a ostale rutine koje su potrebne korisniku moraju da se učitaju pomoću komande APPLOAD. Do ove komande se dolazi putem ***Tools*** > ***AutoLISP*** > ***Load*** i kao rezultat toga se pojavljuje okvir za dijalog, koji je prikazan ispod.

Slika

Mnogi od AutoLISP programa koriste funkciju DEFUN da kreiraju neku funkciju. Posle učitavanja fajla pomoću komande APPLOAD, ta nova funkcija može da se pozove kao i svaka druga AutoCAD-ova komanda, kucajući njen naziv u komandnoj liniji. Na primer, ako je korisnik kreirao funkciju pod nazivom C1 (koristeći u njenoj sintaksi i funkciju DEFUN), onda se ta funkcija poziva, nakon učitavanja, sa:

```
Command: C1
```

Svakoj funkciji se dodeljuje novi naziv (novo ime), bez obzira da li se naziv zadaje malim ili velikim slovima, kao prvi element u listi, sa argumentima te funkcije (ako argumenti postoje), kao sledećim elementima u listi.

Operatori

Ovde će biti nabrojano i razjašnjeno 13 (trinaest) operatora unutar AutoLISP-a.

(+ [broj1] [broj2] [broj3] ...)

Ovaj operator prikazuje sumu (zbir) svih *[broj]*-eva. Ako su svi brojevi celobrojni (*integer*), onda će i rezultat (zbir) biti celobrojan. Ako je neki od *[broj]*-eva realan, u nizu celobrojnih, onda će se svi integer brojevi pretvoriti u realne, tako da će i rezultat biti realan.

(+ 1 2)	rezultat je	3
(+ 1 2 3 4.6)	rezultat je	10.6
(+ 1 2 3 4.0)	rezultat je	10.0

(- [broj1] [broj2] [broj3] ...)

Ovaj operator oduzima drugi *[broj]* od prvog i prikazuje razliku. Ako su data tri broja, onda se, najpre, oduzima drugi broj od prvog, a onda se od njihove razlike oduzima treći broj i taj rezultat se prikazuje. Ako je dat samo jedan broj, onda se taj broj oduzima od 0 (nule) i taj rezultat se prikazuje. Ovaj operator može da koristi i celobrojne i realne brojeve, sa već objašnjenom procedurom pretvaranja rezultata, kao kod operatora za sabiranje.

(- 50 40)	rezultat je	10
(- 50 40.0 3)	rezultat je	7.0
(- 50 40.0 3.5)	rezultat je	6.5
(- 7)	rezultat je	-7

(* [broj1] [broj2] [broj3] ...)

Ovaj operator prikazuje proizvod svih *[broj]*-eva. I ovaj operator koristi i celobrojne i realne brojeve sa već objašnjenom procedurom pretvaranja rezultata.

(* 2 4)	rezultat je	8
(* 2 4 6.0)	rezultat je	48.0
(* 3 –4.5)	rezultat je	-13.5

(/ [broj1] [broj2] [broj3] ...)

Ovaj operator prikazuje količnik dva *[broj]*-a. Ako su data tri broja, onda se najpre deli prvi *[broj]* sa drugim, a onda njihov količnik sa trećim *[broj]*-em, da bi se na kraju prikazao rezultat. I ovaj operator koristi i celobrojne i realne brojeve, sa već objašnjenom procedurom pretvaranja rezultata.

(/ 100 2)	rezultat je	50
(/ 100 2.0)	rezultat je	50.0
(/ 100 20.0 2)	rezultat je	2.5
(/ 100 20 2)	rezultat je	2
(/ 135 360)	rezultat je	0
(/ 135 360.0)	rezultat je	0.375

(= [atom1] [atom2] [atom3] ...)

Ovo je *equal to* (jednako) relacioni operator. Kao rezultat se pojavljuje ili ***T***, ako su specificirani *[atom]*-i numerički jednaki, ili se pojavljuje ***nil*** u obrnutom slučaju. Ovaj operator je primenljiv i na brojeve i na tekstualne stringove.

(= 4 4.0)	rezultat je	T
(= 20 388)	rezultat je	nil
(= 3.6 3.6 3.6)	rezultat je	T
(= 399 399 400)	rezultat je	nil
(= "ja" "ja")	rezultat je	T
(= "ja" "on")	rezultat je	nil

(/= [atom1] [atom2])

Ovo je not equal to (nije jednako) relacioni operator. Kao rezultat se pojavljuje ili ***T***, ako *[atom1]* nije numerički jednak *[atom2]*, ili ***nil*** ako su *[atom1]* i *[atom2]* numerički jednaki. Ovaj operator je primenljiv i na brojeve i na tekstualne stringove, i ne prihvata više od dva argumenta, znači samo *[atom1]* i *[atom2]*.

(/= 4 4.0)	rezultat je	nil
(/= 20 21)	rezultat je	T
(/= "ja" "ja")	rezultat je	nil
(/= "ja" "on")	rezultat je	T

(< [atom1] [atom2] [atom3] ...)

Ovo je *less than* (manje od) relacioni operator. Kao rezultat se pojavljuje ili ***T***, ako je *[atom1]* manji od *[atom2]*, ili se pojavljuje ***nil*** u obrnutom slučaju. Ako je dato više *[atom]*-a, rezultat će biti ***T*** ako je svaki *[atom]* manji od onog sa svoje desne strane.

(< 10 20)	rezultat je	T
(< "b" "c")	rezultat je	T
(< 357 355.2)	rezultat je	nil
(< 2 5 7)	rezultat je	T
(< 2 3 4 5 5)	rezultat je	nil

(<= [atom1] [atom2] [atom3] ...)

Ovo je *less than or eqaul to* (manje od ili jednako) relacioni operator. Kao rezultat se pojavljuje ili ***T***, ako je *[atom1]* manji ili jednak drugom, ili se pojavljuje ***nil*** u obrnutom slučaju. Ako je dato više *[atom]*-a, rezultat će biti ***T*** ako je svaki *[atom]* manji ili jednak onom sa svoje desne strane.

(<= 10 20)	rezultat je	T
(<= "b" "b")	rezultat je	T
(<= 357 355.2)	rezultat je	nil
(<= 2 5 7 7)	rezultat je	T
(<= 2 3 4 5 4)	rezultat je	nil

(> [atom1] [atom2] [atom3] ...)

Ovo je *greater than* (veće od) relacioni operator.Kao rezultat se pojavljuje ili ***T***, ako je prvi *[atom1]* numerički veći od drugog, ili se pojavljuje ***nil*** u obrnutom slučaju. Ako je dato više *[atom]*-a, rezultat će biti ***T*** ako je svaki *[atom]* veći od onog sa svoje desne strane.

(> 20 10)	rezultat je	T
(> "c" "b")	rezultat je	T
(> 355.1 355.2)	rezultat je	nil
(> 7 5 3 2)	rezultat je	T
(> 7 5 3 3)	rezultat je	nil

(>= [atom1] [atom2] [atom3] ...)

Ovo je *greater than or equal to* (veće od ili jednako) relacioni operator.Kao rezultat se pojavljuje ili ***T***, ako je prvi *[atom1]* numerički veći ili jednak drugom, ili se pojavljuje ***nil*** u obrnutom slučaju. Ako je dato više *[atom]*-a, rezultat će biti ***T*** ako je svaki *[atom]* veći ili jednak onom sa svoje desne strane.

(>= 20 19)	rezultat je	T
(>= "c" "c")	rezultat je	T
(>= 355.1 355.2)	rezultat je	nil
(>= 7 5 3 3)	rezultat je	T
(>= 7 5 3 4)	rezultat je	nil

(~ [broj])

Ovaj operator prikazuje prvi komplement *[broj]*-a. Prvi komplement je broj u binarnom označavanju, dobijen iz drugog binarnog broja jednostavnom promenom vrednosti svake cifre. Treba posebno napomenuti da *[broj]* mora biti dat kao *integer* broj.

(~ 7)	rezultat je	-8
(~ 100)	rezultat je	-101
(~ -7)	rezultat je	6
(~ -100)	rezultat je	99

(1+ [broj])

Ovaj operator povećava *[broj]* za 1. Ovaj *[broj]* može biti i celobrojan i realan.

(1+ 7)	rezultat je	8
(1+ -16.5)	rezultat je	-15.5

(1- [broj])

Ovaj operator smanjuje *[broj]* za 1. Ovaj *[broj]* može biti i celobrojan i realan.

(1- 7)	rezultat je	6
(1- -16.54)	rezultat je	-17.54

AutoLISP (16)

Pripremio Dragan Cvetković

AutoCAD je softverski paket firme Autodesk, Inc. za crtanje tehničkih i drugih crteža, a može se koristiti na personalnim računarima. Što se crtanja tiče, AutoCAD omogućava crtanje tehničkih crteža za sve grane tehnike (normalno je da je za neke grane tehnike pogodniji, a za neke grane tehnike manje pogodan). Ovaj programski paket omogućava korišćenje programskog jezika AutoLISP, pomoću koga je moguće definisati skoro svaku funkciju koja korisniku zatreba. Ovaj rad daje kratak osvrt na šest dodatnih funkcija:

- funkcija koja omogućava da se markira kružnica ili luk, i poluprečnik markirane kružnice ili luka setuje kao *fillet* poluprečnik, tj. poluprečnik zaobljenja,
- funkcija koja omogućava ispisivanje teksta po zakrivljenoj putanji poli-linije,
- funkcija koja omogućava pridruživanje dve prekinute kolinearne linije,
- funkcija koja omogućava automatizovanu višestruku primenu komande *Offset*,
- funkcija koja omogućava izračunavanje površine zatvorene konture, automatski oduzimajući površine otvora ili rupa u toj zatvorenoj konturi i
- funkcija koja omogućava kreiranje lučnih segmenata na osnovu definisanih pravolinijskih segmenata polilinije.

Definisanje željenog poluprečnika zaobljenja

Često se dolazi u situaciju da korisniku treba zaobljenje identično već postojećem na crtežu. Da bi se skratio i pojednostavio postupak oko ovog definisana je funkcija **ZAOBLJENJE**, koja omogućava da se markira kružnica ili luk, i poluprečnik markirane kružnice ili luka se setuje kao *fillet* poluprečnik. Sledi listing ove funkcije u sintaksi programskog jezika AutoLISP:

```
(defun C:CC3 (code ename)
```

```
   (cdr (assoc code (entget ename)))
)
 (defun C:ZAOBLJENJE (/ oldech oldsnp olderr xpt esel
                        xent etype newcir)
(setq oldech (getvar "CMDECHO")
        oldsnp (getvar "OSMODE")
        olderr *ERROR*
   )
(defun *ERROR* (msg)
     (princ " \n") (princ msg)
     (setvar "CMDECHO" oldech)
     (setvar "OSMODE"  oldsnp)
     (setq   *ERROR*   olderr)
     (princ)
   )
(setvar "CMDECHO" 0)
   (setvar "OSMODE" 512)
(initget 1)
   (setq xpt (getpoint "\nIzaberi krug ili luk: "))
(cond ((setq esel (nentselp xpt))
         (setq xent (car esel))
        )
(T
    (command ".CIRCLE" xpt pause)
    (setq xent (entlast)
          newcir T)
   )
   )
(cond ((not (or (= "CIRCLE"
              (setq etype (c:cc3 0 xent)))
                 (= "ARC" etype)
              )
         )
         (command ".CIRCLE" xpt pause)
         (setq xent (entlast)
               newcir T)
        )
   )
(setvar "FILLETRAD" (c:cc3 40 xent))
(princ "\nNovi FILLET radijus: ")
   (princ (getvar "FILLETRAD"))
(setvar "CMDECHO" oldech)
   (setvar "OSMODE"  oldsnp)
   (setq   *ERROR*   olderr)
   (princ)
)
```

Tekst i zakrivljena putanja

Često treba ispisati tekst po zakrivljenoj putanji, što predstavlja određen problem unutar AutoCAD-a. Rešenje je dato definisanjem funkcije **PUTANJA**, koja omogućava da se tekst ispiše po putanji koja mora biti ili *Polyline* ili *Spline*. Sledi sintaksa ove funkcije:

```
(defun C:PUTANJA (/ ent1 pt1 txt tleng txtlst entlst
```

```
                loc ang loclst count ctx cht )
(Setvar "cmdecho" 0)
(if (NOT (TBLSEARCH "BLOCK" "$CHAR"))
  (command "point" (getvar "viewctr")
         "block" "$char" (getvar "viewctr") "l" "")
)
(setq ent1 (entsel "\nIzaberi path za tekst: "))
(setq pt1 (cadr ent1))
(setq txt (getstring T "\nUnesi tekst: "))
(princ "Sacekajte")
(setq tleng (strlen txt))
(command "divide" pt1 "b" "$char" "" (1+ tleng))
(setq txtlst '())
(setq count tleng)
(while (> count 0)
 (setq entlst (entget (entlast)))
 (setq loc (cdr (assoc 10 entlst)))
 (setq ang (* 57.2958(cdr (assoc 50 entlst))))
 (setq loclst (append loclst (list (cons loc ang))))
 (entdel (entlast))
 (princ ".")
 (setq count (1- count))
)
(setq count 1)
(setq loclst (reverse loclst))
(setvar "blipmode" 0)
 (while (< count (1+ tleng))
 (setq loc (car (car loclst)))
 (setq ang (cdr (car loclst)))
 (setq loclst (cdr loclst))
 (setq ctxt (getvar "textstyle"))
 (setq cht (cdr (assoc 40 (tblsearch "style" ctxt))))
 (if (= cht 0.0)
   (command "text" "c" loc "" ang (substr txt count 1))
   (command "text" "c" loc ang (substr txt count 1))
 )
 (setq count (1+ count))
)
(setvar "blipmode" 1)
(PRINC)
)
```

Kolinearne linije

Često se sreće korisnik sa problemom prekinute linije. Da bi se nastavila takva linija, trebalo bi,najpre, u krajnjoj tački jedne linije nacrtati novu liniju, a zatim drugu liniju produžiti (primena komande *Extend*) do nove linije. Na kraju treba obrisati docrtanu liniju. Ovaj postupak je skraćen definisanjem funkcije **KOLINEAR**, koja omogućava nastavljanje kolinearnih linija, bez pomoćnih koraka. Sledi sintaksa ove funkcije:

```
(defun C:KOLINEAR (/ a b ai va vb dst1 dst2 a1 b1
                     a2 b2 ent1 ent2
            nd1 end2 dst3 dst4 oldlst newlst)
 (defun mid (w z)
```

```
(LIST (/ (+ (CAR w) (CAR z))2) (/
                    (+ (CADR w) (CADR z))2))
)
  (setvar "cmdecho" 0)
  (setq a (cadr (entsel "\nIzaberi dve
                     linije koje treba spojiti: ")))
  (setq b (cadr (entsel)))
  (setq ai (mid a b))
  (setq va (ssget a))
  (setq vb (ssget b))
  (setq a1 (cdr (assoc 10 (entget
                (setq ent1 (ssname va 0))))))
  (setq b1 (cdr (assoc 11
                    (entget (ssname va 0)))))
  (setq a2 (cdr (assoc 10 (entget
                (setq ent2 (ssname vb 0))))))
  (setq b2 (cdr (assoc 11
                    (entget (ssname vb 0)))))
    (setq dst1 (distance ai a1))
    (setq dst2 (distance ai b1))
     (if (< dst1 dst2)(setq end1 b1)
                              (setq end1 a1))
    (setq dst3 (distance ai a2))
    (setq dst4 (distance ai b2))
    (if (< dst3 dst4)(setq end2 b2)
                              (setq end2 a2))
(setq oldlst (entget ent1))
(setq newlst (subst (cons 10 end1)
                    (assoc 10 oldlst) oldlst))
(setq newlst (subst (cons 11 end2)
                    (assoc 11 newlst) newlst))
(entdel ent2)
(entdel ent1)
(entmake (cdr newlst))
(princ)
)
```

Automatizovana višestruka primena komande *Offset*

Komanda *Offset* programa AutoCAD-a je veoma moćna, ali ima jedan nedostatak: pravi samo jednu kopiju željenog objekta. Problem je rešen definisanjem funkcijom **MOFFSET**. Sledi sintaksa ove funkcije:

```
(defun C:MOFFSET( / ent spt dist)
  (setq #mdist
    (udist 1 "" "OFFSET rastojanje " #mdist nil))
  (while
    (not (setq ent (entsel
                  "\nIzaberi objekat za OFFSET: "))))
  (setq spt (upoint 1 "" "Izberi stranu" nil (cadr ent))
        #mnum (uint 5 "" "Koliko puta? " #mnum)
  )
  (setq dist #mdist)
  (repeat #mnum
     (command "_.offset" dist ent spt "")
```

```
    (setq dist (+ dist #mdist))
  )
  (princ)
)
(princ)
 (defun udist (bit kwd msg def bpt / inp)
  (if def
    (setq msg (strcat "\n" msg "<" (rtos def) ">: ")
          bit (- bit (boole 1 bit 1))
    )
(if (= " " (substr msg (strlen msg) 1))
      (setq msg (strcat "\n"
            (substr msg 1 (1- (strlen msg))) ": "))
(setq msg (strcat "\n" msg ": "))
  ) )
  (initget bit kwd)
  (setq inp
    (if bpt
      (getdist msg bpt)
      (getdist msg)
  ) )
  (if inp inp def)
)
 (defun uint (bit kwd msg def / inp)
  (if def
    (setq msg (strcat "\n" msg "<" (itoa def) ">: ")
          bit (- bit (boole 1 bit 1))
)
    (if (= " " (substr msg (strlen msg) 1))
      (setq msg (strcat "\n"
            (substr msg 1 (1- (strlen msg))) ": "))
(setq msg (strcat "\n" msg ": "))
  ) )
  (initget bit kwd)
  (setq inp (getint msg))
  (if inp inp def)
)
 (defun upoint (bit kwd msg def bpt / inp)
  (if def
    (setq pts (strcat
                (rtos (car def))
                "," (rtos (cadr def))
                (if
                  (and (caddr def) (= 0
                           (getvar "FLATLAND")))
                  (strcat "," (rtos (caddr def)))
                  ""))
msg (strcat "\n" msg "<" pts ">: ")
          bit (- bit (boole 1 bit 1)))
 (if (= " " (substr msg (strlen msg) 1))
      (setq msg (strcat "\n"
          (substr msg 1 (1- (strlen msg))) ": "))
(setq msg (strcat "\n" msg ": "))
  ) )
```

```
  (initget bit kwd)
  (setq inp
    (if bpt
      (getpoint msg bpt)
      (getpoint msg)
  ) )
  (if inp inp def)
)
```

Površina neke konture

Program AutoCAD ima komandu pod nazivom AREA, koja omogućava izračunavanje površine markirajući granične tačke. Definisana funkcija POVRSINA omogućava izračunavanje površine zatvorene konture (markirajući konture kao objekte), automatski oduzimajući površine otvora ili rupa. Sledi sintaksa ove funkcije:

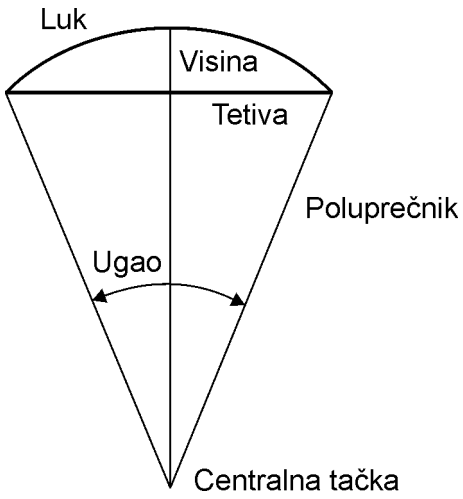
```
(defun C:POVRSINA ( / plate ss1 count emax)
  (while
    (not (setq plate
           (entsel "\nIzaberi spoljnu granicu: "))))
  (prompt "\nIzaberi sve rupe i odsecke...")
  (setq ss1 (ssget))
  (ssdel (car plate) ss1)
  (command "_.area" "_a" "_e" plate "" "_s" "_e")
  (setq count 0
        emax (sslength ss1)
  )
  (while (< count emax)
    (command (ssname ss1 count))
    (setq count (1+ count)))
(command "" "")
  (prompt (strcat "\nKonacna povrsina je "
                  (rtos (getvar "AREA") 2)
                  " sq."))
  (princ)
)defun
(princ)
```

Bulge lukovi

Ova funkcija omogućava korisniku da kreira lukove umesto pravolinijskih delova polilinije po sledećem pravilu:

$$Bulge = 2\ Visina / Tetiva = tg\ (Ugao / 4)$$

gde je objašnjenje oznaka dato na slici 1.



Slika 1. Potrebni parametri za određivanje luka

Sledi sintaksa OBLAK funkcije:

```
(defun C:OBLAK ( / head hdata bulge en ed)
  (if (and
       (setq en (entsel "\nIzaberi poliliniju: "))
       (= (dxf 0 (setq hdata
                 (entget (car en)))) "POLYLINE")
      )
      (progn
        (entmod (subst '(70 . 1) '(70 . 0) hdata))
        (setq bulge (list (cons 42 0.5)))
        (setq en (dxf -1 hdata))
        (while (and (setq en (entnext en))
                    (setq ed (entget en))
                    (/= "SEQEND" (dxf 0 ed))
               )
               (setq ed (append ed bulge))
               (entmod ed)
        )
        (entupd en)
  ) )
  (princ)
)
(princ)
```

Ovo su samo neke dodatne funkcije koje korisnik može da kreira unutar programskog jezika AutoLISP. Ako korisnik solidno barata sa ovim programskim jezikom, onda može da napravi dosta toga i može sebi dosta da olakša život.

AutoLISP (2)

Pripremio Dragan Cvetković

Sledi spisak ostalih funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi.

(abs [broj])

Ova funkcija predstavlja apsolutnu vrednost *[broj]*-a. Ovaj *[broj]* može da bude i celobrojan i realan.

(abs 10) rezultat je 10
(abs –10) rezultat je 10
(abs –43.2) rezultat je 43.2

(and [izraz] ...)

Ova funkcija "vraća" logički AND liste izraza, tj. Zaustavlja evoluciju i vraća vrednost *nil* ako bilo koji izraz evoluira ka *nil*. U obrnutom slučaju je *T*. Na primer, za sledeće parametre:

```
(setq a 102)
(setq b nil)
(setq c "string")
```

sledi:

(and 1.2 b c) rezultat je T
(and 1.2 a b c) rezultat je nil

(angle [pt1] [pt2])

Ova funkcija određuje ugao linije koja prolazi kroz tačke *[pt1]* i *[pt1]*. Ugao se meri od pozitivnog smera X ose odgovarajuće ravni u radijanima (ugao raste u smeru koji je suprotan smeru kretanja kazaljke na satu). Ako se definišu 3D tačke (zadaju se sve tri dimenzije) , onda se tačke projektuju na odgovarajuću ravan.

(angle '(1.0 1.0) '(1.0 4.0)) rezultat je 1.5708
(angle '(5.0 1.33) '(2.4 1.33)) rezultat je 3.14159

(angtof [string] [mod])

Ova funkcija konvertuje i prikazuje vrednost *[string]*-a, koji predstavlja vrednost ugla koju treba pretvoriti u radijane. Format prikazivanja *[string]*-a određuje se uz pomoć *[mod]* argumenta:

0 – decimalni stepeni;
1 – stepeni/minuti/sekunde;
2 – gradi;
3 – radijani; i
4 – posebne (katastarske) jedinice.

Ako argument *[mod]* nije deklarisan, onda će se iskoristiti tekuća vrednost AUNITS sistemske promenljive. Izgled različitih formata je prikazan u sledećoj tabeli.

[mod]	AUNITS	Primer
0	Decimalni stepeni	45.0000
1	Stepeni/minute/sekunde	45d0'0"
2	Gradi	50.0000g
3	Radijani	0.7854r
4	Katastarske jedinice	N 45d0'0" E

(angtos [angle] [mod [preciznost]])

Ova funkcija uređuje vrednost ugla *[angle]* (koja je realna i izražena u radijanima) u string koji može da se setuje opcijama *[mod]* i *[preciznost]*, dimenzionom promenljivom DIMZIN i sistemskom promenljivom UNITMODE. Opcija *[mod]* je ista kao kod **angtof** funkcije. Opcije *[mod]* i *[preciznost]* su u vezi sa sistemskim promenljivama AUNITS i AUPREC programskog paketa AutoCAD. Ako se ove opcije eksplicitno ne navedu, promenljive AUNITS i AUPREC će biti setovane na uobičajene (podrazumevane) vrednosti. Na primer, za DIMZIN=0 i za sledeće vrednosti:

```
(setq pt1 '(5.0 1.33))
(setq pt2 '(2.4 1.33))
(setq a (angle pt1 pt2))
```

sledi:

(angtos a 0 0)	rezultat je	"180"
(angtos a 0 4)	rezultat je	"180.0000"
(angtos a 1 4)	rezultat je	"180d0'0""
(angtos a 3 4)	rezultat je	"3.1416r"
(angtos a 4 2)	rezultat je	"W"

Ova funkcija prihvata i negativne *[angle]* argumente, ali ih uvek redukuje na pozitivnu vrednost između 0 i 2π radijana:

(angtos 0.785398 0 4)	rezultat je	"45.0000"
(angtos –0.785398 0 4)	rezultat je	"315.0000"

Treba napomenuti i da sistemska promenljiva UNITMODE ima uticaja na prikazivanje stringa, i to kada se aktiviraju katastarske jedinice (*[mod]=4*). Na primer, ako sistemska promenljiva UNITMODE ima vrednost 0, onda je dozvoljen prikaz praznog prostora u tekstualnom stringu (na primer "N 45d E"), a ako sistemska promenljiva UNITMODE ima vrednost 1, onda nije dozvoljen prazan međuprostor u tekstualnom stringu (na primer "N45dE").

(append [izraz] ...)

Ova funkcija "uzima" bilo koji broj iz liste *[izraz]* i radi sa svim brojevimakao sa jednom listom.

(append '(a b) '(c d))	rezultat je	(A B C D)
(append '((a) (b)) '((c) (d)))	rezultat je	((A) (B) (C) (D))

(apply [funkcija] [lista])

Ova funkcija omogućava izvršavanje funkcije koja je specificirana u delu *[funkcija]* sa argumentima koji su specificirani u delu pod nazivom *[lista]*. Dva primera su prikazana ispod, s tim što je u drugom primeru korišćena **strcat** funkcija koja nadovezuje stringove, jedan na drugi.

(apply '+ '(1 2 4))	rezultat je	7
(apply 'strcat '("a" "b" "c"))	rezultat je	"abc"

(ascii [string])

Ova funkcija vrši konverziju prvog karaktera *[string]*-a u njegov ASCII karakter (celobrojna vrednost).

(ascii "A")	rezultat je	65

(ascii "a")	rezultat je	97
(ascii "BRAVO")	rezultat je	66

(assoc [item] [list])

Ova funkcija vrši pretragu liste *[list]* za argumentom *[item]*, kao ključnim elementom, i prikazuje sadržinu (vrednost) *[item]*-a. Ako se *[item]* ne pronađe u listi, onda ova funkcija prikazuje vrednost *nil*. Na primer, ako je lista "DDD" definisana kao:

```
Command: (setq ddd '((ime Kutija) (sirina 3.3) (velicina 4.736) (dubina 6)))
((IME KUTIJA) (SIRINA 3.3) (VELICINA 4.736) (DUBINA 6))
```

sledi:

(assoc 'sirina ddd)	rezultat je	SIRINA 3.3
(assoc 'ime ddd)	rezultat je	IME KUTIJA
(assoc 'tezina ddd)	rezultat je	nil

(atan [broj1] [broj2])

Ova funkcija izračunava arkustangens *[broj]*-a i rezultat prikazuje u radijanima. Ako vrednost *[broj2]* nije zadata, onda će funkcija atan prikazivati arkustangens vrednosti *[broj2]*, u radijanima. Vrednost *[broj]* može biti i negativna, što znači da će vrednost ugla biti u opsegu od $-\pi$ do $+\pi$ radijana.

(atan 0.5)	rezultat je	0.463648
(atan 1.0)	rezultat je	0.785398
(atan –1.0)	rezultat je	-0.785398
(angtos (atan –1.0) 0 4)	rezultat je	"315.0000"

Ako su date obe vrednosti, i *[broj1]* i *[broj2]*, onda ova funkcija prikazuje arkustangens *[broj1]/ [broj2]* u radijanima. Ako je *[broj2]=0*, onda će vrednost biti +1.570796 ili –1.570796 radijana ($+90^o$ ili -90^o), što zavisi od znaka *[broj1]*.

(atan 2.0 3.0)	rezultat je	0.588003
(angtos (atan 2.0 3.0) 0 4)	rezultat je	"33.6901"
(atan 2.0 –3.0)	rezultat je	2.553590
(angtos (atan 2.0 -3.0) 0 4)	rezultat je	"146.3099"
(atan –2.0 3.0)	rezultat je	-0.588003
(atan –2.0 –3.0)	rezultat je	-2.553590
(atan 1.0 0.0)	rezultat je	1.5708
(angtos (atan 1.0 0.0) 0 4)	rezultat je	"90.0000"
(atan –0.5 0.0)	rezultat je	-1.5708
(angtos (atan –0.5 0.0) 0 4)	rezultat je	"270.00"

(atof [string])

Ova funkcija vrši konverziju *[string]*-a u realni broj.

(atof "89.3")	rezultat je	89.3
(atof "7")	rezultat je	7

(atoi [string])

Ova funkcija vrši konverziju *[string]*-a u celobrojni broj, s tim što treba napomenuti da ova funkcija jednostavno briše decimalni nastavak, tj. sve iza decimalne tačke.

(atoi "79")	rezultat je	79
(atoi "5")	rezultat je	5
(atoi "6.4")	rezultat je	6

(atom [item])

Ova funkcija prikazuje vrednost nil ako je *[item]* lista, a prikazuje vrednost T u obrnutom slučaju. Ako je dato:

```
(setq a '(x y z))
(setq b 'a)
```

tada je:

(atom 'a)	rezultat je	T
(atom a)	rezultat je	nil
(atom 'b)	rezultat je	T
(atom b)	rezultat je	T
(atom '(a b c))	rezultat je	nil

AutoLISP (3)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što će ovde biti reči o funkcijama koje počinju na slovo A.

(acad_colordlg broj [marker])

Ova funkcija prikazuje standardni okvir za dijalog AutoCAD-a za izbor boja. Vrednost *broj* označava broj boje koja će biti aktivirana. Ovaj broj može da ima vrednosti od 0 do 256. Ako je izabrana vrednost 0, onda se aktivira *ByBlock* boja, a ako je izabrana vrednost 256, onda se aktivira *ByLayer* boja. Vrednost *[marker]* omogućava ili onemogućava upotreba **ByLayer** i **ByBlock** tastera. Ako je vrednost *[marker]*-a izabrana da bude nil, onda se onemogućava upotreba **ByLayer** i **ByBlock** tastera. U svim ostalim slučajevima omogućava se upotreba ovih tastera. Na primer, ako se otkuca:

```
Command: (acad_colordlg 2)
```

pojaviće se *Color Select* okvir za dijalog i biće aktivirana žuta (engleski *yellow*) boja.

(acad_helpdlg fajl topic)

Ova funkcija je zaostatak prethodnih verzija AutoCAD-a, a od verzije 2002 izbačena je iz upotrebe jer je uključena u standardan *Help* ovog programskog paketa. Omogućavala je čitanje pomoćnih informacija iz eksternih fajlova (u sintaksi je to *fajl*), a pretraga je poboljšana i mogućnošću da se traži određeni naziv (u sintaksi je to *topic*).

(acad_strlsort lista)

Ova funkcija omogućava da se parametri u listi poređaju po abecedi. Ako neki parametar u listi nije u redu ili ako nema dovoljno memorije da se parametri sortiraju, onda ova funkcija prikazuje *nil* kao odgovor. Na primer, ako se formira lista pod nazivom Avion:

```
Command: (setq avion '("F-16" "A-10" "B-1" "Orao" "G-4" "F-15"))
("F-16" "A-10" "B-1" "Orao" "G-4" "F-15")
```

i kada se startuje ova funkcija, onda se dobija rezultat:

```
Command: (acad_strlsort avion)
("A-10" "B-1" "F-15" "F-16" "G-4" "Orao")
```

(acdimenableupdate T ¦ nil)

Ova funkcija kontroliše automatsko ažuriranje dimenzionih promenljivih (promenljivih za kotiranje). Ako se aktivira *nil* argument asocijativna kota neće biti ažurirana (čak i ako se geometrija menjala), sve dok se ne aktivira DIMREGEN komanda. Ako se aktivira *T* argument, onda je omogućeno automatsko ažuriranje asocijativnih kota kada se geometrija menja.

(acdimenableupdate T)	rezultat je	Omogućava automatsko ažuriranje
(acdimenableupdate nil)	rezultat je	Onemogućava automatsko ažuriranje

(acet-attsync blok)

Ova funkcija omogućava ažuriranje instanci naznačenog bloka sa tekućim definicajama atributa u tekućem crtežu. Ova funkcija je identična komandi ATTSYNC programskog paketa AutoCAD. Naziv *blok* u sintaksi označava naziv bloka koji treba ažurirati. Dozvoljena je primena specijalnih karaktera (reč je o zvezdici – *) u definisanju naziva bloka. Ako je ažuriranje uspešno, onda se kao rezultat pojavljuje *T*, a u suprotnom pojavljuje se *nil*. Na primer, naredna sintaksa pokazuje da je izvršeno uspešno ažuriranje atributa bloka pod nazivom Prozvod:

```
Command: (acet-attsync "Proizvod")
T
```

Naredna sintaksa pokazuje da je izvršeno uspešno ažuriranje atributa blokova čiji nazivi počinju sa Proi:

```
Command: (acet-attsync "Proi*")
T
```

(acet-layerp-mode [status])

Ova funkcija omogućava da se vidi vrednost LAYERPMODE promenljive i da se ta vrednost podesi prema potrebama korisnika. Argument *[staus]* može da ima dve vrednosti: ako se definiše *T*, onda se promenljiva LAYERPMODE aktivira, a ako se definiše *nil*, onda se ova promenljiva deaktivira. Ako se ovaj argument u sintaksi izostavi, onda ova funkcija vraća stanje promenljive LAYERPMODE na prvobitno stanje. Kao rezultat se pojavljuje ili *T* (ako je status promenljive – uključeno) ili *nil* (ako je status ove promenljive – isključeno). Na primer, provera tekućeg statusa promenljive LAYERPMODE se vrši sa:

```
Command: (acet-layerp-mode)
T
```

Pošto se vidi da je promenljiva uključena (aktivna), onda se njeno isključivanje vrši sa:

```
Command: (acet-layerp-mode nil)
nil
```

Provera novog statusa ove promenljive se vrši sa:

```
Command: (acet-layerp-mode)
```

nil

(acet-layerp-mark [status])

Ova funkcija postavlja markere za početak i za kraj snimanja izmena na slojevima, kako bi mogla da se aktivira LAYERP (engelski *Layer Prevoius*) komanda. Ova funkcija omogućava korisniku da grupiše višestruke izmene slojeva ili na slojevima u jedinstveni zapis, i aktiviranjem komande LAYERP može sve to da vrati na prvobitno stanje odjednom. Da bi ovo funkcionisalo promenljiva LAYERPMODE mora da bude uključena. Argument *[staus]* može da ima dve vrednosti: *T* startuje zapisivanje, a *nil* završava zapisivanje, a ujedno i uklanja marker za početak zapisivanja. Ako se ovaj argument izostavi u sintaksi, onda ova funkcija vraća vrednosti na prvobitno stanje. Pošto promenljiva LAYERPMODE mora da bude uključena, onda tu od pomoći može da bude sledeća sintaksa:

```
(defun TestLayerP ()
  (if (not (acet-layerp-mode))
    (acet-layerp-mode T))
```

koja aktivira ovu promenljiva, ako nije uključena. Nakon toga se menja boja sloja pod nazivom 0 u žutu:

```
(command "_.layer" "_color" "yellow" "0" "")
```

Sada se aktivira ova funkcija i nakon toga sledi nekoliko izmena na ovom sloju:

```
(acet-layerp-mark T)
(command "_.layer" "_color" "green" "0" "")
(command "_.layer" "_thaw" "*" "")
(command "_.layer" "_unlock" "*" "")
(command "_.layer" "_ltype" "center" "0" "")
(command "_.layer" "_color" "red" "0" "")
(acet-layerp-mark nil)
```

Na kraju, ako se aktivira komanda LAYERP, onda se vrednosti sloja 0 vraćaju na prvobitno stanje, što znači da je aktivna žuta boja, kako je definisano na početku.

(acet-laytrans crtez [setovanje])

Ova funkcija omogućava prevođenje slojeva tekućeg crteža na neke standardne vrednosti koje su definisane u nekom drugom crtežu ili u nekom drugom fajlu. Naziv *crtez* u sintaksi označava destinaciju i ime fajla koji se koristi za prevođenje vrednosti slojeva. Argument *[setovanje]* predstavlja vrednost kôda koji definiše koje opcije treba izvršiti. Kôdovi mogu biti grupisani u bilo kojoj kombinaciji, s tim što je vrednost između 0 i 15. Ako se ovaj argument izostavi u sintaksi, onda će sve opcije biti uključene, što je isto kao da je definisana vrednost 15. Ovaj argument može da ima sledeće vrednosti:

- 0 Isključuje sve opcije
- 1 Menja boju entiteta (objekata) u BYLAYER
- 2 Menja tip linije entiteta (objekata) u BYLAYER
- 4 Prevodi objekte u blokove
- 8 Zapisuje sve što je urađeno

Ako je prevođenje bilo uspešno u komandnoj liniji AutoCAD-a se pojavljuje *T*, a ako nešto nije u redu, onda se pojavljuje *nil*. Na primer, sledeća sintaksa prevodi slojeve tekućeg crteža koristeći setovanje koje je smešteno u *SlojMapa.dwg* crtežu, sve opcije su aktivirane sem zapisivanja:

```
Command: (acet-laytrans "e:/Moji crtezi/Standard/SlojMapa.dwg" 7)
T
```

(acet-ms-to-ps [vrednost] [viewport])

Ova funkcija konvertuje realne vrednosti jedinica prostora modela u jedinice prostora papira. Ako su u sintaksi definisana oba argumenta, i *[vrednost]* i *[viewport]*, onda se konvertovanje vrši upotrebom upravo tog *viewport*-a. Ako je definisan samo argument *[vrednost]*, onda se primenjuje tekući *viewport*. Ako je tekući prostor prostor modela, onda nema aktivnog *viewport*-a, tako da će funkcija odgovoriti sa *nil*, tj. prekinuće rad. Ako je tekući prostor prostor papira, onda će funkcija da traži da se specificira *viewport* (ako ih ima više) ili će da iskoristi tekući *viewport* (ako je aktivan samo jedan). Ako argumenti nisu definisani, onda će funkcija tražiti da se unesu vrednosti i konvertovaće ih, ako je to moguće. Kao rezultat rada pojaviće se konvertovana vrednost ili će se pojaviti *nil*, ako nije uspelo konvertovanje.

(acet-ps-to-ms [vrednost] [viewport])

Ova funkcija konvertuje realne vrednosti jedinica prostora papira u jedinice prostora modela. Ako su u sintaksi definisana oba argumenta, i *[vrednost]* i *[viewport]*, onda se konvertovanje vrši upotrebom upravo specificiranog *viewport*-a. Ako je definisan samo argument *[vrednost]*, onda se primenjuje tekući *viewport*. Ako je tekući prostor prostor modela, onda nema aktivnog *viewport*-a, tako da će funkcija odgovoriti sa *nil*, tj. prekinuće rad. Ako je tekući prostor prostor papira, onda će funkcija da traži da se specificira *viewport* (ako ih ima više) ili će da iskoristi tekući *viewport* (ako je aktivan samo jedan). Ako argumenti nisu definisani, onda će funkcija tražiti da se unesu vrednosti i konvertovaće ih, ako je to moguće. Kao rezultat rada pojaviće se konvertovana vrednost ili će se pojaviti *nil*, ako nije uspelo konvertovanje.

AutoLISP (4)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se završavaju funkcije čiji naziv počinje slovom A.

(action_tile ključ akcija)

Ova funkcija omogućava korisniku da startuje neku komandu ili opciju kada aktivira odgovarajući taster ili naziv unutar izabranog okvira za dijalog. Opcija *akcija* suspenduje podazumevanu radnju aktiviranog okvira za dijalog i aktivira željenu komandu ili opciju. Ova sintaksa ima par opcija:

$value	aktivira uobičajenu vrednost koja je dodeljena izabranom nazivu
$key	dodeljuje novi naziv pod kojim se aktivira željena opcija ili komanda
$data	aktivira specificirane podatke, ako to izabrani okvir za dijalog zahteva
$x i $y	dodeljuje koordinate, ako je izbaran specificirani taster

Napomena: *Pomoću ove funkcije ne mogu da se pozivaju (aktiviraju) AutoLISP funkcije.*

Opcija *ključ* ($key) dodeljuje naziv koji aktivira željenu opciju ili komandu. Ako je aktivirana funkcija uspešno izvršena, onda se u komandnoj liniji pojavljuje *T* kao odgovor, a ako nešto nije u redu u komandnoj liniji se pojavljuje *nil* kao odgovor.

(add_list string)

Ova funkcija dodaje ili menja string u listi tekućeg aktivnog okvira za dijalog. Pre nego što se aktivira ova funkcija, korisnik mora da otvori postojeću listu i da je inicijalizuje pomoću start_list funkcije. U zavisnosti

od toga šta je specificirano pomoću funkcije start_list, naznačeni *string* će se dodati u izabranu listu ili će zameniti željeni deo liste. Ako je bila uspešna akcija, onda se u komandnoj liniji pojavljuje naziv stringa koji je dodat listi, ili se pojavljuje *nil* ako akcija nije bila uspešna. Naredna sintaksa inicijalizuje listu i dodaje tekstualne stringove u *lllist*-u:

```
(setq lllist '("prva linija" "druga linija" "treca linija"))
(start_list "longlist")
(mapcar 'add_list lllist)
(end_list)
```

Nakon definisanja liste i njenog sadržaja, naredna sintaksa menja tekst u drugoj liniji liste sa stringom "2.linija":

```
(start_list "longlist" 1 0)
(add_list "2.linija")
(end_list)
```

(alert string)

Ova funkcija prikazuje okvir za dijalog koji sadrži poruku upozorenja ili poruku o nekoj greški. Definisana opcija *string* definiše tekst koji se pojavljuje u pomenutom okviru za dijalog. Ako se uradi sve kako treba, onda se u grafičkoj oblasti *AutoCAD*-a pojavljuje okvir za dijalog sa naslovom *AutoCAD Message* i unutar njega se prikazuje tekst koji je definisan *string* opcijom. Ako nešto nije urađeno kako treba, onda se u komandnoj liniji pojavljuje *nil* kao odziv.

Ako korisnik hoće da prikaže tekstualni string *Funkcija nije dostupna!!* u pomenutom okviru za dijalog, onda je sintaksa za to:

```
(alert "Funkcija nije dostupna!!")
```

Ako korisnik hoće da prikaže željeni tekstualni string u više redova, onda bi trebalo aktivirati karakter koji signalizira da je reč o početku nove linije, i ta sintaksa bi mogla da ima oblik:

```
(alert "Zeljena funkcija \nnije dostupna!!")
```

Napomena: *Dužina tekstualnog stringa (teksta) i broj linija teksta zavisi od platforme, uređaja i definisanog okvira. AutoCAD podešava dužinu teksta kako bi ispunio definisani okvir za dijalog.*

(alloc broj)

Ova funkcija setuje veličinu segmenta koji će biti iskorišćen od strane *expand* funkcije. Opcija *broj* u sintaksi je celobrojni broj koji definiše veličinu memorije koja će biti upotrebljena od strane pomenute funkcije. Ovaj broj predstavlja broj simbola, tekstualnih stringova, ćelija, i slično. Kao rezultat, posle aktiviranja ove funkcije, u komandnoj liniji se pojavljuje prethodna vrednost ove funkcije.

```
Command: (alloc 249)
256
```

arx

Ova funkcija prikazuje listu učitanih ObjectARX aplikacija. Kao rezultat primene ove funkcije, u komandnoj liniji se prikazuje spisak imena ObjectARX aplikacija, s tim što u taj odziv nisu uključene i putanje do fajlova koji su učitani.

```
Command: (arx)
("acadapp.arx" "acdblclkedit.arx" "acdblclkeditpe.arx" "acdim.arx"
```

"aceplotx.arx" "acetlodr.arx" "achapi15.dbx" "achlnkui.arx" "acpltstamp.arx"
"actoday.arx" "oleaprot.arx" "vl.arx")

(arxload aplikacija [otkaz])

Ova funkcija omogućava učitavanje ObjectARX aplikacije. Opcija *aplikacija* omogućava specificiranje naziva aplikacije koju treba učitati. Treba napomenuti da mora da se ukuca *arx* ekstenzija prilikom učitavanja, kao i celokupna putanja do izvršnog fajla aplikacije koja se učitava. Opcija *[otkaz]* definiše šta korisnik mora da odradi, ako učitavanje specificirane aplikacije nije izvršeno.

Kao odziv ove funkcije prikazuje se naziv ObjectARX aplikacije, ako je učitavanje bilo uspešno. Ako korisnik pokuša da učita aplikaciju koja je već učitana, onda će ova funkcija prikazati poruku o greški. Zbog toga je najbolje da se, najpre, funkcijom *arx* proveri šta je od ObjectARX aplikacija već učitano. Ako korisnik hoće da učita aplikaciju MojaAplikacija.arx iz instalacionog direktorijuma AutoCAD-a, onda je sintaksa:

Command: (arxload "c:/Program Files/AutoCAD 2002/MojaAplikacija.arx")
"c:/Program Files/AutoCAD 2002/MojaAplikacija.arx"

(arxunload aplikacija [otkaz])

Ova funkcija omogućava poništavanje učitavanja ObjectARX aplikacije. Opcija *aplikacija* omogućava specificiranje naziva aplikacije koju treba izbaciti iz radnog prostora AutoCAD-a. Treba napomenuti da mora da se ukuca *arx* ekstenzija prilikom poništavanja učitavanja, kao i celokupna putanja do izvršnog fajla aplikacije kojoj se poništava učitavanje. Opcija *[otkaz]* definiše šta korisnik mora da odradi, ako poništavanje učitavanja specificirane aplikacije nije izvršeno.

Kao odziv ove funkcije prikazuje se naziv ObjectARX aplikacije, ako je funkcija uspešno izvršena. Ako korisnik pokuša da poništi aplikaciju koja je već poništena, onda će ova funkcija prikazati poruku o greški. Treba napomenuti da zaključanim ObjectARX aplikacijama ne može da se poništi učitavanje. Sintaksa za poništavanje učitavanja aplikacije koja je učitana u prethodnoj funkciji je:

Command: (arxunload "MojaAplikacija.arx")
"MojaAplikacija.arx"

(atoms-family format [simboli])

Ova funkcija omogućava prikazivanje liste već definisanih simbola. Opcija *format* ima celobrojne vrednosti 0 ili 1, i ona definiše format kako će ova funkcija prikazati nazive simbola:

- **0** Prikazuje nazive simbola kao listu
- **1** Prikazuje nazive simbola kao listu stringova

Što se opcije *[simboli]* tiče, ona definiše listu stringova koja označava nazive simbola koje ova funkcija traži.

Kao odziv ove funkcije prikazuje se lista simbola. Ako korisnik definiše opciju *[simboli]*, onda ova funkcija prikazuje spisak simbola koji su već definisani, a vraća kao rezultat *nil* ako neki simbol u listi nije definisan. Na primer, ako usledi sintaksa:

Command: (atoms-family 0)

sledi poduži spisak simbola, tako da je ovde prikazan samo manji deo, a ostalo je simbolizovano sa tri tačkice (...), jer je spisak poduži.

ACRX_CMD_REDRAW _VLISP-VERSION MEMBER C:DDPTYPE GET_TILE APPLY ANGTOF WHILE
GRDRAW WRITE-CHAR MIN C:DWFOUT CADAR VPORTS ACCOV-REMOVE ASCII EXPAND
INITSTRING FOREACH VL-BT-ON CADAAR *ERROR* AUTOARXACEDLOAD IMPORT3DS defun-q-

list-ref ~ CDAAAR AND LOADEDP AI_ONOFF VLISP-DCLRES-LOAD-DIALOG C:AI_SELALL COMMAND IMAGEFILE LA_SUPPORT_ASSISTANCE_ALERT T MEM VL-FILENAME-DIRECTORY TRANS C:AI_SPHERE C:AI_CONE _VER VL-FILENAME-BASE > >= = < C:ROTATE3D AI_SHOWEDGE_ALERT XDSIZE PROGN / ENTLAST - LSH VL-INFP + * FITSTR2LEN ACCOV-VISIT SLIDE_IMAGE BPOLY MENUGROUP ACTION_TILE <= MAX SUBSTR ...

Naredna sintaksa proverava da li su simboli CAR, CDR i XYZ definisani i prikazuje izlaznu listu:

```
Command: (atoms-family 1 '("CAR" "CDR" "XYZ"))
("CAR" "CDR" nil)
```

Prikazani odziv pokazuje da su simboli CAR i CDR definisani, ali simbol XYZ nije definisan.

(autoarxload ime lista)

Ova funkcija omogućava da se definiše naziv komande kojom će se startovati pridružena ObjectARX aplikacija (fajl). Kada se prvi put upotrebi naziv komande koji je definisan u opciji *lista*, onda AutoCAD učitava aplikaciju koja je specificirana opcijom *ime*, i nastavlja sa tom komandom. Ako nešto nije u redu sa ukucanom sintaksom, onda će AutoCAD da signalizira da nešto nije u redu.

Opcija *ime* određuje naziv ObjectARX fajla (ima ekstenziju ARX) koji će biti učitan kada se aktivira neka komanda iz opcije *lista*, kada se startuje iz komandne oblasti (komandni prompt, *Command:*). Ako u sintaksi ne postoji specificirana putanja do ObjectARX fajla, onda će AutoCAD taj fajl da traži u definisanoj putanji *Support File Search Path*.

Kao rezultat aktiviranja ove funkcije u komandnoj liniji se pojavljuje *nil* kao odziv. Na primer, ako se definiše da se aktivira aplikacija VISETRUKI_MIRROR.ARX komandama VM1, VM2 i VM3, onda je za to predviđena sintaksa:

```
Command: (autoarxload "VISETRUKI_MIRROR" '("VM1" "VM2" "VM3"))
nil
```

(autoload ime lista)

Ova funkcija omogućava da se definiše naziv komande kojom će se startovati pridružena AutoLISP aplikacija (fajl). Kada se prvi put upotrebi naziv komande koji je definisan u opciji *lista*, onda AutoCAD učitava aplikaciju koja je specificirana opcijom *ime*, i nastavlja sa tom komandom. Ako nešto nije u redu sa ukucanom sintaksom, onda će AutoCAD da signalizira da nešto nije u redu.

Opcija *ime* određuje naziv AutoLISP fajla (ima ekstenziju LSP) koji će biti učitan kada se aktivira neka komanda iz opcije *lista*, kada se startuje iz komandne oblasti (komandni prompt, *Command:*). Ako u sintaksi ne postoji specificirana putanja do AutoLISP fajla, onda će AutoCAD taj fajl da traži u definisanoj putanji *Support File Search Path*.

Kao rezultat aktiviranja ove funkcije u komandnoj liniji se pojavljuje *nil* kao odziv. Na primer, ako se definiše da se aktivira aplikacija VISETRUKI_MIRROR.LSP komandama VM1, VM2 i VM3, onda je za to predviđena sintaksa:

```
Command: (autoload "VISETRUKI_MIRROR" '("VM1" "VM2" "VM3"))
nil
```

AutoLISP (5)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima B i C.

(Boole [func] [int1] [int2] ...)

Ova funkcija predstavlja "pametnu" *Boolean* funkciju. Oznaka *[func]* je broj između 0 i 15 koji predstavlja jednu od 16 mogućih *Boolean* funkcija sa dve promenljive. Argumenti su kombinovani na bazi ove funkcije i na bazi naredne tabele:

Int1	**Int2**	**Func bit**
0	0	8
0	1	4
1	0	2
1	1	1

Svaki bit *[int1]* se uparuje sa odgovarajućim bitom *[int2]* koji se biraju iz gornje tabele. Rezultujući bit je **0** ili **1**, što zavisi od setovanja *[func]***.** Ako je bit setovan u *[func]*, rezultujući bit je 1, u svim ostalim slučajevima rezultujući bit je 0. Neke vrednosti *[func]* su identične standardnim *Boolean* operacijama: AND, OR, XOR i NOT.

Func	*Operacija*	*Rezultujući bit je 1 ako...*
1	AND	su oba ulazna bita jednaka 1.
6	XOR	je samo jedan od dva ulazna bita jednak 1.
7	OR	je ili jedan ili oba ulazna bita jednaka 1.
8	NOT	su oba ulazna bita jednaka 0.

(boundp [atom])

Ova funkcija prikazuje T ako *[atom]* ima graničnu vrednost, a prikazuje nil u suprotnom slučaju. Na primer:

```
(setq a 2)
(setq b nil)
```

sledi

```
(boundp 'a)   rezultat je   T
(boundp 'b)   rezultat je   nil
```

(caar, cadr, cddr, cadar,...)

AutoLISP podržava vezu CAR i CDR kroz četiri nivoa. Na primer:

```
(setq x '((a b) c d))
```

sledi:

```
(caar x)   isto je   (car (car x))   rezultat je   A
(cdar x)   isto je   (cdr (car x))   rezultat je   (B)
```

(cadar x)	isto je	(car (cdr (car x)))	rezultat je	B
(cadr x)	isto je	(car (cdr x))	rezultat je	C
(cddr x)	isto je	(cdr (cdr x))	rezultat je	(D)
(caddr x)	isto je	(car (cdr (cdr x)))	rezultat je	D

U AutoLISP-u se funkcija CADR koristi za dobijanje Y koordinate 2D ili 3D tačke (to je druga cifra od 2 ili 3 broja). Funkcija CADDR se koristi za dobijanje Z koordinate 3D tačke. Na primer:

(setq pt2 '(6.34 2.1)) ovo je za 2D tačku
(setq pt3 '(6.34 2.1 3.1)) ovo je za 3D tačku

sledi:

(car pt2)	rezultat je	6.34
(cadr pt2)	rezultat je	2.1
(caddr pt2)	rezultat je	nil
(car pt3)	rezultat je	6.34
(cadr pt3)	rezultat je	2.1
(caddr pt3)	rezultat je	3.1

(car [list])

Ova funkcija prikazuje prvi element iz *[list]*-e. Ako je *[list]* prazna, onda se kao rezultat prikazuje *nil*.

(car '(a b c))	rezultat je	A
(car '((a b) c))	rezultat je	(A B)
(car '())	rezultat je	nil

(cdr [list])

Ova funkcija prikazuje sve elemente iz *[list]*-e, sem prvog elementa. Ako je *[list]* prazna, onda se kao rezultat prikazuje *nil*.

(cdr '(a b c))	rezultat je	(A B)
(cdr '((a b) c))	rezultat je	(C)
(cdr '())	rezultat je	nil

Ako su argumenti *[list]*-e odvojeni tačkom, onda funkcija CDR prikazuje drugi argument bez njegovog zatvaranja u listi.

(cdr '(a.b))	rezultat je	B
(cdr '(2."tekst"))	rezultat je	"tekst"

(chr [broj])

Ova funkcija pretvara prikazani broj (celobrojni) u string dužine jednog karaktera. Na primer:

(chr 65)	rezultat je	"A"
(chr 66)	rezultat je	"B"
(chr 97)	rezultat je	"a"

(close [fajl])

Ova funkcija "zatvara" fajl i prikazuje nil. Opcija *[fajl]* je opis fajla koji se dobija od OPEN funkcije. Posle primenjene CLOSE funkcije opis fajla se ne menja, ali više nije aktivan. Na primer,

```
(close PROBA)
```

zatvara se PROBA, kao otvoreni opis fajla.

(command [argumenti]...)

Ova funkcija izvršava komande AutoCAD-a iz AutoLISP-a i uvek "vraća" nil. Na primer,

```
(setq pt1 '(3.45 4.55))
(setq pt2 '(getpoint "Unesi drugu tacku: "))
(command "line" pt1 pt2)
(command " ")
```

ovim se aktivira komanda LINE koja iscrtava liniju od tačke *pt1* do tačke *pt2* (koja se naknadno unosi).

(cond [test1] [rezultat1]...)

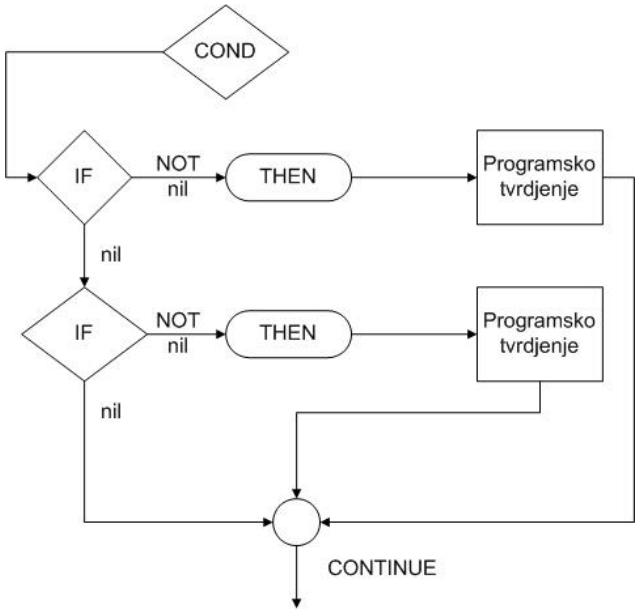
Ova funkcija prihvata bilo koji broj iz liste kao argument. Na primer, naredna upotreba COND funkcije određuje kalkulaciju apsolutne vrednosti:

```
(cond ((minusp a) (-a))
      (t a))
```

Ako je *a* setovan na vrednost –13, prikazaće se vrednost 13. U sledećem primeru korisnik ima simbol *s* kao *[rezultat1]*. Ova funkcija testira argument *[rezultat1]* i prikazuje 1, ako je on *Y* ili *y*, a prikazuje 0 ako je on *N* ili *n*, ili prikazuje *nil* u ostalim slučajevima.

```
(cond ((= s "Y") 1)
      ((= s "y") 1)
      ((= s "N") 0)
      ((= s "n") 0)
      (t nil))
```

Na slici ispod prikazana je struktura ove funkcije.



(cons [novi element] [list])

Ova funkcija vraća *[novi element]* na početak liste *[list]*.

(cons 'a '(b c d))	rezultat je	(A B C D)
(cons '(a) '(b c d))	rezultat je	((A) B C D)

(cos [ugao])

Ova funkcija izračunava kosinus ugla *[ugao]*, gde je *[ugao]* izražen u radijanima.

(cos 0.0)	rezultat je	1.0
(cos pi)	rezultat je	-1.0

AutoLISP (6)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima C, D i E.

(cvunit [vrednost] ”iz jedinice” ”u jedinicu”)

Ova funkcija omogućava prebacivanje jedinice, iz jednih mera u druge. Parametar *[vrednost]* predstavlja numeričku vrednost ili listu koordinata tačaka (2D ili 3D tačaka) koje će biti konvertovane. Parametar *"iz jedinice"* predstavlja naziv jedinice koje se konvertuju, a parametar *"u jedinicu"* predstavlja naziv jedinice u koju se konvertuje. Parametri *"iz jedinice"* i *"u jedinicu"* predstavljaju nazive jedinica koje su definisane unutar fajla ACAD.UNT programa AutoCAD i koji se nalazi unutar instalacionog direktorijuma. Ako je sve u redu sa zadatom sintaksom, onda program prikazuje konvertovanu vrednost, a ako nešto nije u redu onda prikazuje *nil* u komandnoj oblasti. Prikazaće *nil* ako program pokušava da nađe naziv jedinice koja ne postoji unutar ACAD.UNT fajla ili ako dve jedinice nisu kompatibilne (na primer, ako korisnik pokušava da konvertuje kilograme u metre). Evo nekoliko primera i nekoliko odziva:

```
Command: (cvunit 1 "minute" "second")
60.0

Command: (cvunit 1 "gallon" "minute")
nil

Command: (cvunit 1.0 "inch" "cm")
2.54

Command: (cvunit 1 "inch" "mm")
25.4

Command: (cvunit '(1.0 2.5) "ft" "in")
(12.0 30.0)

Command: (cvunit '(1 2 3) "ft" "in")
(12.0 24.0 36.0)
```

(defun [ime] [lista argumenata] [izraz]...)

Ova funkcija definiše novu funkciju pod nazivom *[ime]* sa pripadajućim argumentima:

- (defun mojafunk (x y)...) – funkcija ima dva argumenta,
- (defun mojafunk (/ a c)...) – funkcija ima dva lokalna simbola,
- (defun mojafunk (x / ttmp)...) – funkcija ima jedan argument i jedan lokalni simbol,
- (defun mojafunk ()...) – funkcija nema ni argumente, ni lokalne simbole.

Na primer, sledeća sintaksa:

Command: (defun dodaj10 (x) (ovde je pritisnut taster **Enter**)
(_> (+ 10 x))
DODAJ10

daje novu funkciju pod nazivom DODAJ10. Kada se nova funkcija aktivira, onda se pojavljuju odzivi:

Command: (dodaj10 5)
15

Command: (dodaj10 -6.7)
3.3

Sledi još jedan primer gde se ubacuju tri tačkice (...) između argumenata. Sledeća sintaksa je dobijena kucanjem u komandnoj oblasti programa AutoCAD:

Command: (defun tackice (x y / ttmp) (ovde je pritisnut taster **Enter**)
(_> (setq ttmp (strcat x "...")) (ovde je pritisnut taster **Enter**)
(_> (strcat ttmp y)) (ovde je pritisnut taster **Enter**)
TACKICE

daje novu funkciju pod nazivom TACKICE. Kada se nova funkcija aktivira, odzivi su:

Command: (tackice "x" "y")
"x...y"

Command: (tackice "od" "do")
"od...do"

Postoji još jedna primena ove funkcije da bi se formirala nova komanda unutar programskog paketa AutoCAD. Treba napomenuti, da bi ova funkcija radila kako treba argument *[ime]* definisati da bude u formi ″C:XXX″. Sledeći primer definiše funkciju koja će da iscrta kvadrat koristeći poliliniju. Prikazana sintaksa je dobijena kucanjem u komamndoj oblasti programa AutoCAD.

Command: (defun C:KVADRAT (/ pt1 pt2 pt3 pt4 duz) (ovde je pritisnut taster **Enter**)
(_> (setq pt1 (getpoint "Donji levi ugao: ")) (ovde je pritisnut taster **Enter**)
(_> (setq duz (getdist pt1 "Duzina stranice: ")) (ovde je pritisnut taster **Enter**)
(_> (setq pt2 (polar pt1 0.0 duz)) (ovde je pritisnut taster **Enter**)
(_> (setq pt3 (polar pt2 (/ pi 2.0) duz)) (ovde je pritisnut taster **Enter**)
(_> (setq pt4 (polar pt3 pi duz)) (ovde je pritisnut taster **Enter**)
(_> (command "PLINE" pt1 pt2 pt3 pt4 "C")) (ovde je pritisnut taster **Enter**)
C:KVADRAT

Kada se startuje ova funkcija dobija se odziv:

Command: kvadrat
Donji levi ugao: (koordinata može da se definiše kursorom miša ili da se zada sa tastature)
Duzina stranice: (dužina može da se zada mićem ili da se zada sa tastature)

PLINE
Specify start point:
Current line-width is 0.0000
Specify next point or [Arc/Halfwidth/Length/Undo/Width]:
Specify next point or [Arc/Close/Halfwidth/Length/Undo/Width]:
Specify next point or [Arc/Close/Halfwidth/Length/Undo/Width]:
Specify next point or [Arc/Close/Halfwidth/Length/Undo/Width]: C
Command: nil

(distance [pt1] [pt2])

Ova funkcija omogućava korisniku da izračuna rastojanje između tačaka *[pt1]* i *[pt2]* u prostoru.
Command: (distance '(1.0 2.4 3.0) '(7.7 2.5 3.1))
6.70149

Command: (distance '(1.0 1.0 1.0) '(1.0 1.0 2.0))
1.0

(distof [string] [mod])

Ova funkcija prikazuje vrednost [string]-a kao realan broj. Argument [mod] određuje format prikazivanja. Ako [mod] nije specificiran, onda će se iskoristiti tekuće setovanje LUNITS sistemske promenljive. I sistemska promenljiva UNITMODE ima uticaja na format. Vrednost argumenta [mod] i sistemskih promenljivih prikazane su u sledećoj tabeli.

[mod]	LUNITS	UNITMODE 0	UNITMODE 1
1	Naučni	1.55E+01	1.55E+01
2	Decimalni	15.50	15.50
3	Inženjerski	1′-3.50″	1′3.50″
4	Arhitektonski	1'-3 1/2"	1'3-1/2"
5	Razlomci	15 1/2	15-1/2

(eq [izraz1] [izraz2])

Ova funkcija određuje kada su *[izraz1]* i *[izraz2]* identični. Funkcija EQ prikazuje *T*, kada su izrazi identični, a prikazuje *nil*, u svim ostalim slučajevima. Ako se dodeli:

Command: (setq f1 '(a b c))
(A B C)
Command: (setq f2 '(a b c))
(A B C)
Command: (setq f3 f2)
(A B C)

sledi:

Command: (eq f1 f3)
nil (jer nemaju iste liste)

Command: (eq f3 f2)
T (jer imaju iste liste)

(equal [izraz1] [izraz2] [razlika])

Ova funkcija određuje kada su izrazi *[izraz1]* i *[izraz2]* jednaki, tj. kada teže istoj stvari (vrednosti). Prethodni primer,

```
Command: (setq f1 '(a b c))
(A B C)
Command: (setq f2 '(a b c))
(A B C)
Command: (setq f3 f2)
(A B C)
```

daje sledeće odgovore:

```
Command: (equal f1 f3)
T (jer teže istoj stvari)

Command: (equal f3 f2)
T (jer imaju iste liste)
```

Ovo samo potvrđuje da dve liste mogu da budu EQUAL, ali ne moraju da budu EQ. U definiciji funkcije postoji numerički argument *[razlika]* koji predstavlja maksimalnu brojnu vrednost za koliko mogu da se razlikuju dva broja, koji se upoređuju, a da budu EQUAL. Na primer,

```
Command: (setq aa 1.1234)
1.1234
Command: (setq bb 1.1236)
1.1236
```

i sledi:

```
Command: (equal aa bb)
nil

Command: (equal aa bb 0.0002)
T
```

(*error* [string])

Ova funkcija omogućava korisniku da refeniše grešku. Argument *[string]* prima poruku od AutoLISP-a, koja je opisana kao greška.

(eval [izraz])

Ova funkcija prikazuje rezultat priloženog *[izraz]*-a, gde je *[izraz]* bilo koji LISP izraz. Na primer,

```
Command: (setq ccc 121)
121
Command: (setq xx 'ccc)
CCC
```

sledi:

Command: (eval 4.0)
4.0

Command: (eval (abs -34.54))
34.54

Command: (eval ccc)
121

Command: (eval xx)
121

(exp [broj])

Ova funkcija prikazuje rezultat $e^{[broj]}$ i to je neki realan broj. Trebalo bi napomenuti da je *e* osnova prirodnog logaritma i to je vrednost 2.71828.

Command: (exp 1.0)
2.71828

Command: (exp 2.4)
11.0232

Command: (exp -0.56)
0.571209

(exp [baza] [broj])

Ova funkcija prikazuje rezultat $[baza]^{[broj]}$. Ako su oba argumenta celobrojni brojevi, onda će i rezultat biti celobrojan. U drugim slučajevima rezultat je realan broj. Evo nekoliko primera:

Command: (expt 2 2)
4

Command: (expt 3 4.0)
81.0

Command: (expt 3.5 4.5)
280.741

Command: (expt 3.0 2)
9.0

AutoLISP (7)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima F i G.

(findfile [naziv fajla])

Ova funkcija omogućava korisničkim fajlovima da pretražuju biblioteku AutoCAD-a u potrazi za specificiranim fajlom. Kao rezultat, pojavljuje se ili putanja do fajla ili *nil*, ako fajl ne postoji na disku.

```
Command: (findfile "abc.lsp")
"C:\\Program Files\\AutoCAD 2004\\abc.lsp"

Command: (findfile "xyz.txt")
"C:\\Program Files\\AutoCAD 2004\\xyz.txt"

Command: (findfile "proba")
nil
```

(fix [broj])

Ova funkcija pretvara *[broj]* u celobrojnu vrednost. Parametar *[broj]* može da ima i celobrojnu i realnu vrednost. Ako je ovaj parametar realan broj, onda ga funkcija pretvara u celobrojnu vrednost, i to tako što, jednostavno, zanemari decimale.

```
Command: (fix 34)
34

Command: (fix 34.567)
34

Command: (fix -34.456)
-34
```

(float [broj])

Ova funkcija pretvara parametar *[broj]* u realan broj. Ovaj parametar može da ima i realnu i celobrojnu vrednost.

```
Command: (float 44)
44.0

Command: (float 44.34)
44.34

Command: (float -23)
-23.0
```

(float [naziv] [lista] [izraz]...)

Ova funkcija prolazi kroz listu koja je definisana parametrom *[lista]* i dodeljuje svakom argumentom *[naziv]* i onda izvršava *[izraz]* za svaki element u listi. Na primer,

```
Command: (foreach n '(a b c) (print n))

A
B
C C
```

je ekvivalentno sledećoj sintaksi (bez upotrebe funkcije FOREACH),

```
(print a)
(print b)
(print c)
```

gde funkcija FOREACH prikazuje rezultat samo zadnjeg izraza.

(gc)

Ova funkcija neutrališe nepotrebne stvari (stvari koje su aktivirane ili učitane u radno okruženje programa AutoCAD, a nisu nikada upotrebljene), i na taj način se oslobađa radna memorija.

(gcd [broj1] [broj2])

Ova funkcija prikazuje najveći zajednički delilac brojeva *[broj1]* i *[broj2]*. Treba napomenuti da ovi brojevi moraju da imaju celobrojne vrednosti.

```
Command: (gcd 234 344)
2

Command: (gcd 120 18)
6

Command: (gcd 57 81)
3
```

(getangle [pt] [prompt])

Ova funkcija pravi pauzu da bi korisnik ukucao drugu koordinatu, kako bi se odredio ugao između početne tačke (koja je definisana unutar sintakse) i druge tačke. Argument *[prompt]* je opcioni string koji se pojavljuje u komandnoj oblasti, dok argument *[pt]* predstavlja 2D tačku. Vrednost izračunatog ugla je izražena u radijanima.

```
Command: (getangle '(1.1 3.5))
0.697045

Command: (getangle '(0 0) "Koji pravac?")
Koji pravac?0.71731

Command: (getangle '(0 0) "Koji pravac?")
Koji pravac?2,2
0.785398
```

(getcorner [pt] [prompt])

Ova funkcija prikazuje tačku u odgovarajućem UCS, i to na taj način što se argument *[pt]* tretira kao početna tačka i počinje da se iscrtava pravougaonik iz te tačke. Ako se ukucaju 3D koordinate, Z koordinata će biti ignorisana. Argument *[prompt]* je opcioni string koji se pojavljuje u komandnoj oblasti.

```
Command: (getcorner '(1 1))
(199.959 224.313 0.0)
```

Command: (getcorner '(2 3) "Druga tacka?")
Druga tacka?@24,34
(26.0 37.0 0.0)

(getdist [pt] [prompt])

Ova funkcija izračunava i prikazuje rastojanje između dve tačke. Prva tačka je definisana argumentom *[pt]*, a druga tačka se unosi sa tastature ili kursorom miša. Argument *[prompt]* je opcioni string koji se pojavljuje u komandnoj oblasti, kao eventualno pitanje ili podsetnik šta korisnik treba da uradi. Ovde nema ograničenja po pitanju 2D ili 3D tačaka.

Command: (getdist '(0 0))
@23<65
23.0

Command: (getdist '(0 0) "(Unesi drugu tacku)")
(Unesi drugu tacku)@34,56
65.5134

(getenv [naziv])

Ova funkcija prikazuje vrednost stringa koja je dodeljena promenljivoj okruženja. Argument *[naziv]* je string koji specificira naziv promenljive koju treba pročitati. Ako zadata promenljiva ne postoji, onda se pojavljuje rezultat *nil*.

Command: (getenv "ACAD")
"D:\\Documents and Settings\\Administrator\\Application Data\\Autodesk\\AutoCAD 2004\\R16.0\\enu\\support;D:\\Program Files\\AutoCAD 2004\\support;D:\\Program Files\\AutoCAD 2004\\fonts;D:\\Program Files\\AutoCAD 2004\\help;D:\\Program Files\\AutoCAD 2004\\support\\color;"

Command: (getenv "proba")
nil

Command: (getenv "MaxArray")
"100000"

AutoLISP (8)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima G i I.

(getint [odziv])

Ova funkcija pravi pauzu kako bi korisnik ukucao celobrojni broj, da bi se taj broj prikazao kao integer. Vrednost boja je u dijapazonu od –32768 do +32767.

Command: (getint)
3457890

Requires an integer between -32768 and 32767.
3457
3457

Command: (getint)
-345
-345

(getkword [odziv])

Ova funkcija funkcija trađi odgovor od korisnika koji on treba da ukuca sa tastature. Lista osnovnih odgovara setuje se (podešava) pomoću funkcije INITGET, o kojoj će malo kasnije biti reči. Funkcija GETKWORD prikazuje odgovor korisnika sa tastature kao tekstualni string u komandnoj liniji.

Command: (initget 1 "Da Ne")
nil

Command: (setq x (getkword "Da li ste sigurni? (Da ili Ne)"))
Da li ste sigurni? (Da ili Ne)da
"Da"

(getorient [tačka] [odziv])

Unutar AutoLISP-a uglovi se uvek predstavljaju u radijanima, gde je nulti (početni) pravac pozitivan smer X ose, i uglovi rastu (u pozitivnom smislu) u smeru suprotnom od smera kretanja kazaljke na satu. Kada je potrebno da se izabere neki drugi pravac i drugi smer rasta ugla, onda se koristi GETORIENT funkcija. Funkcija GETANGLE se koristi kada treba da se radi sa relativnim uglovima, dok se funkcija GETORIENT koristi kada treba raditi sa apsolutnim uglovima.

Ulaz (stepeni)	GETANGLE	GETORIENT
0	0.0	1.5708
-90	1.5708	3.14159
180	3.14159	4.71239
90	4.71239	0.0

(getpoint [tačka] [odziv])

Ova funkcija pravi pauzu da bi korisnik ukucao koordinate tačke. Parametar TAČKA je opciono 2D ili 3D tačka u odgovarajućem koordinatnom sistemu, a ODZIV je opcioni string koji se pojavljuje u komandnoj liniji. Na primer,

Command: (getpoint)
2,3
(2.0 3.0 0.0)

Command: (setq p (getpoint "Gde? "))
Gde? 34,56
(34.0 56.0 0.0)

Command: (setq vv (getpoint '(1.4 2.4) "Druga tacka: "))
Druga tacka: 34,56
(34.0 56.0 0.0)

(getreal [odziv])

Ova funkcija pravi pauzu da bi korisnik ukucao realan broj i da bi se taj broj prikazao kao realan. Na primer,

Command: (getreal)
34.56
34.56

Command: (setq broj (getreal "Faktor skaliranja je: "))
Faktor skaliranja je: 0.345
0.345

(getstring [cr] [prompt])

Ova funkcija pravi pauzu da bi korisnik ukucao string, kako bi se taj string pojavio u komandnoj liniji. Treba napomenuti da ako je string duži od 132 karaktera, onda će se prikazati prvih 132 karaktera, a ostalo se zanemaruje. Opcija PROMPT je opcioni string koji se prikazuje kjao odziv u komandnoj oblasti. Na primer,

Command: (getstring)
Pera "Pera"

Command: (setq cc (getstring "Unesite ime: "))
Unesite ime: Mika
"Mika"

(getvar [promenljiva])

Ova funkcija prikazuje vrednost sistemske promenljive AutoCAD-a. Naziv promenljive mora da se nalazi unutar navodnika. Na primer,

Command: (getvar "Ltscale")
1.0

Command: (getvar "osmode")
55

(graphscr)

Ova funkcija omogućava prelaz iz tekstualnog ekrana (koji se aktivira funkcijskim tasterom F2) u grafički ekran. Funkcija TEXTSCR radi suprotnu stvar.

(if [izraz1] [izraz2] [izraz3])

Ova funkcija procenjuje izraze i u zavisnosti od tačnosti izvršava ih. Ako IZRAZ1 nije *nil*, onda se izvršava IZRAZ2, a u ostalim slučajevima se izvršava IZRAZ3. Ako IZRAZ3 nedostaje u specifikaciji, a IZRAZ1 je *nil*, onda funkcija IF prikazuje *nil*.

Command: (if (= 1 3) "Da" "Ne")
"Ne"

Command: (if (= 2 (+ 1 1)) "Da!!!!")
"Da!!!!"

Command: (if (= 2 (+ 3 6)) "Da")
nil

(inters [pt1] [pt2] [pt3] [pt4] [seg])

Ova funkcija određuje dve linije i prikazuje koordinate presečne tačke ili *nil* ako se linije ne seku. Parametri PT1 i PT2 su krajnje tačke prve linije, a parametri PT3 i PT4 su krajnje tačke druge linije. Parametar SEG je, uglavnom, *nil*.

Command: (setq a '(1.0 1.0) b '(10.0 1.0))
(10.0 1.0)

Command: (setq c '(5.0 0.0) d '(5.0 10.0))
(5.0 10.0)

Command: (inters a b c d)
(5.0 1.0)

Command: (inters a b c d T)
(5.0 1.0)

Command: (inters a b c d nil)
(5.0 1.0)

(itoa [int])

Ova funkcija vrši konverziju celobrojnog broja u tekstualni string.

Command: (itoa 345)
"345"

Command: (itoa -456)
"-456"

AutoLISP (9)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima L i M.

(last [lista])

Ova funkcija prikazuje poslednji element u listi. Treba napomenuti da poslednji element u listi ne može da bude *nil*.

Command: (last '(a b c d e))
E

Command: (last '(a b c (d e f)))
(D E F)

(length [lista])

Ova funkcija prikazuje ukupan broj elemenata u listi.

Command: (length '(a b c d e))
5

Command: (length '(a b c (d e)))
4

Command: (length '())
0

(list [izraz])

Ova funkcija uzima elemente iz izraza i povezuje ih u zajednički string.

Command: (list 'a 'b 'c 'd)
(A B C D)

Command: (list 'a 'b '(c d))
(A B (C D))

Command: (list 1.23 4.56 7.89)
(1.23 4.56 7.89)

(listp [item])

Ova funkcija prikazuje *T* ako je *[item]* lista, a prikazuje *nil* u drugom slučaju.

Command: (listp '(a b c d))
T

Command: (listp 'a)
nil

Command: (listp 4.567)
nil

(log [broj])

Ova funkcija prikazuje prirodni algoritam broja kao realni broj.

Command: (log 4.5)

1.50408

Command: (log 1.234)
0.210261

Command: (log 1)
0.0

(mapcar [funkcija] [lista1] ... [listan])

Ova funkcija prikazuje rezultat izvršavanja funkcije sa individualnim elementima od LISTA1 do LISTAn, gde se elementi uzimaju kao argumenti za funkciju. Ako se setuje:

(setq a 10 b 20 c 30 d 40)

onda je:

Command: (mapcar '1+ (list a b c d))
(11 21 31 41)

Ovde treba pomenuti LAMBDA funkciju koja može da specificira nepoznatu funkciju da radi kao MAPCAR. Ovo je veoma korisno kada su argumenti funkcije konstantne veličine. Sledi primer, koji je napravljen u prozoru Visual LISP Console, i koji demonstrira upotrebu LAMBDA funkcije sa MAPCAR funkcijom:

```
_$ (mapcar  '(lambda (x)
         (+ x 3)
         )
        '(10 20 30)
)
nil

(13 23 33)
```

(max [broj1] [broj2] ... [brojn])

Ova funkcija prikazuje najveći broj od datih brojeva. Svaki BROJ može da bude ili realan ili celobrojan, što znači da to ne utiče na rezultat.

Command: (max 3.2345 4.567 12)
12.0

Command: (max -34567 234.345 123)
234.345

(mem)

Ova funkcija pruža informacije korisniku o zauzetosti AutoLISP memorije. Kada se ova funkcija startuje dobijaju se sledeće informacije:

Command: (mem)
; GC calls: 12; GC run time: 100 ms

```
Dynamic memory segments statistic:
PgSz  Used  Free  FMCL  Segs  Type
 128     4   507   506     1  bstack body
4096   209     1     1    14  DMxx memory
4096   122    13    13     9  DM Str
  32   988   995   995     1  ::new
4096   325     5     5    22  CONS memory
 256  3393   289    99    14  bytecode area
 512    73   181   122     2  lisp stacks
Segment size: 65536, total used: 63, free: 0
nil
```

(member [izraz] [lista])

Ova funkcija pretražuje LISTU i prikazuje ostatak liste, startujući sa IZRAZOM kao prvim elementom. Ako ne postoji IZRAZ u LISTI, onda ova funkcija prikazuje *nil*.

```
Command: (member 'c '(a b c d e f))
(C D E F)

Command: (member 'q '(a b c d e f))
nil
```

(min [broj1] [broj2] ... [brojn])

Ova funkcija prikazuje najmanji broj od datih brojeva. Svaki BROJ može da bude ili realan ili celobrojan, što znači da to ne utiče na rezultat.

```
Command: (min 123 234 567 32 -12)
-12

Command: (min 2234 456 34 2 678 1.11)
1.11
```

(minusp [broj])

Ova funkcija prikazuje *T* ako je BROJ realan ili celobrojan i teži negativnoj vrednosti. U drugom slučaju ova funkcija prikazuje *nil*.

```
Command: (minusp -2)
T

Command: (minusp -0.00000023)
T

Command: (minusp 0.0000000345)
nil
```

AutoLISP (10)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima N, O, P i Q..

(not [izraz])

Ova funkcija prikazuje *T* ako je *[izraz] nil*, a prikazuje *nil* ako je drugačije. Na primer, ako se setuje da je:

```
Command: (setq a 123)
123

Command: (setq b "tekst")
"tekst"

Command: (setq c nil)
nil
```

onda, prilikom pozivanja i aktiviranja ove funkcije, sledi odziv:

```
Command: (not a)
nil

Command: (not b)
nil

Command: (not c)
T

Command: (not '())
T
```

(nth [n] [list])

Ova funkcija prikazuje "nth"-i element *[list]*-e, gde je *[n]* redni broj elementa koji treba prikazati. Treba napomenuti da je prvi element u listi broj **0** i da treba voditi računa o tome. Ako je *[n]* veći od broja elemenata u *[list]*-i, onda se prikazuje vrednost *nil*.

```
Command: (nth 3 '(a b c d e))
D

Command: (nth 0 '(a b c d e))
A

Command: (nth 5 '(a b c d e))
nil

Command: (nth 0 '(-2 -3))
-2
```

(null [item])

Ova funkcija prikazuje *T* ako *[item]* teži ka *nil*, a prikazuje *nil* u drugom slučaju. Ako se setuje:

```
Command: (setq a 234)
234

Command: (setq b "string")
"string"

Command: (setq c nil)
nil
```

onda, prilikom pozivanja i aktiviranja ove funkcije, sledi odziv:

```
Command: (null a)
nil

Command: (null b)
nil

Command: (null c)
T

Command: (null '())
T
```

(numberp [item])

Ova funkcija prikazuje *T* ako je *[item]* realan ili integer broj, a prikazuje *nil* u drugom slučaju. Ako je setovano:

```
Command: (setq a 123)
123

Command: (setq b 'a)
A
```

onda, prilikom pozivanja i aktiviranja ove funkcije, sledi odziv:

```
Command: (numberp 5)
T

Command: (numberp 4.1456)
T

Command: (numberp "RAF")
nil

Command: (numberp 'a)
nil

Command: (numberp a)
T

Command: (numberp b)
```

nil

Command: (numberp (eval b))
T

(open [fajl] [mod])

Ova funkcija otvara fajl za pristup I/O funkcijama AutoLISP-a. Argument *[fajl]* predstavlja naziv fajla sa ekstenzijom koji treba otvoriti (učitati), a *[mod]* je *read/write* zastavica. Argument *[mod]* može biti:

- "r" – otvara se fajl za čitanje; ako *[fajl]* ne postoji prikazaće se *nil*.
- "w" – otvara se fajl za pisanje; ako *[fajl]* ne postoji kreiraće se novi fajl i podaci će se upisivati u njega, a ako *[fajl]* postoji, onda će njegovi podaci biti prepisani.
- "a" – otvara se fajl dodavanje podataka; ako *[fajl]* postoji, onda će on biti otvoren i kursor miša će se pozicionirati na kraj postojećih podataka, tako da će se novi podaci dodavati postojećim.

Otvaranje postojećeg fajla bi bilo:

Command: (setq a (open "c:/program files/AutoCAD 2004/help/filelist.txt" "r"))
#<file "c:/program files/AutoCAD 2004/help/filelist.txt">

(or [izraz])

Ova funkcija prikazuje logičko OR izraza. OR pretražuje [izraz]-e sa leve na desnu stranu, tražeći [izraz]-e koji su različiti od nil. Kada pronađe prvi OR, prekida pretragu i prikazuje T. Ako su svi izrazi jednaki, OR prikazuje nil.

Command: (or nil nil 444 '())
T

Command: (or nil '())
nil

Command: (or 4 4 4)
T

(pause)

Ova konstanta se koristi unutar komandne funkcije da bi se sačekao unos podataka od strane korisnika.

(pi)

Ovo nije funkcija AutoLISP-a, nego je konstanta **π** koja prikazuje vrednost **3.1415926**.

(polar [pt] [ugao] [odstojanje])

Ova funkcija prikazuje UCS tačku pod datim uglom (definiše ga argument *[ugao]*) i rastojanjem (definiše ga argument *[odstojanje]*) od UCS tačke (definiše je argument *[pt]*). Argument *[ugao]* je izražen u radijanima od X ose , gde ugao raste u smeru suprotnom od smera kretanja kazaljke na satu. Argument *[pt]* može da bude i 3D tačka, s tim što je onda argument *[ugao]* usaglašen sa odgovarajućim ravnima, kako bi se dobila odgovarajuća i ispravna definicija.

Command: (polar '(1.0 1.0 3.5) 0.78539 1.41421)
(2.00001 1.99999 3.5)

Command: (polar '(1.0 1.0) 3.456 5.678)
(-4.39966 -0.755938)

(prinl [izraz] [fajl])

Ova funkcija štampa [izraz] sa ekrana i ponovo prikazuje [izraz]. Argument [izraz] može da bude bilo koji izraz, ali nije poželjno da bude string. Ako postoji [fajl] (i indeks za fajl koji je otvoren za pisanje), onda će [izraz] biti zapisan u fajl i biće prikazan na ekranu. Ako se setuje:

(setq a 345)
(setq b '(a))

sledi da će:

- (prinl 'a) štampa A i prikazuje A
- (prinl a) štampa 345 i prikazuje 345
- (prinl b) štampa (A) i prikazuje (A)
- (prinl "Zdravo") štampa "Zdravo" i prikazuje "Zdravo"

(princ [izraz] [fajl])

Ova funkcija je identična funkciji PRINL.

(print [izraz] [fajl])

I ova funkcija je identična funkciji PRINL.

(prompt [poruka])

Ova funkcija prikazuje poruku na ekranu u liniji gde se nalazi kursor miša i vraća *nil*. Treba napomenuti da je poruka string. Na primer:

Command: (prompt "Nova vrednost: ")
Nova vrednost: nil

Command: (prompt "Ovo je proba!!!! ")
Ovo je proba!!!! nil

(quit)

Ova funkcija tera tekuću aplikaciju da završi sa radom. Kada se funkcija startuje, prikazuje se propratna poruka i korisnik se vraća u AutoCAD-ovu komandnu liniju.

(quote [poruka])

Ova funkcija prikazuje poruku onako kako je ukucana.

Command: (quote Cvele)
CVELE

Command: (quote (a=b))
(A=B)

AutoLISP (11)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima O, P i Q.

(open [naziv fajla] [mod])

Ova funkcija otvara fajl za pristup I/O funkcijama AutoLISP-a. Argument *[naziv fajla]* predstavlja naziv fajla sa ekstenzijom koji treba otvoriti, a *[mod]* je *read/write* zastavica (marker). Ovaj marker može da ima sledeće vrednosti:

- **r** – otvara se fajl za čitanje; ako *[naziv fajla]* ne postoji prikazaće se *nil*.
- **w** – otvara se fajl za pisanje; ako *[naziv fajla]* ne postoji formiraće se novi fajl i podaci će se zapisivati u njega; ako [naziv fajla] postoji, onda će se postojeći podaci prepisati novim podacima.
- **a** – otvara se fajl za dodavanje podataka; ako *[naziv fajla]* ne postoji, formiraće se novi; a ako *[naziv fajla]* postoji, onda će taj fajl biti otvoren i kursor miša će se pozicionirati na kraj postojećih podataka, tako da će se novi podaci dodavati postojećim.

(or [izraz)

Ova funkcija prikazuje logičko OR datog izraza. Funkcija pretražuje izraz sa leve na desnu stranu, tražeći članove koji su različiti od *nil*. Kada se pronađe prvi takav član, pretraga se prekida i prikazuje se *T*.Ako su svi članovi izraza jednaki *nil*, onda funkcija prikazuje *nil*.

Command: (or nil nil 45 '())
T

Command: (or 3 3 3 3 3 '())
T

Command: (or nil nil nil nil '())
nil

(polar [pt] [ugao] [dist])

Ova funkcija prikazuje koordinate 3D tačke tekućeg koordinatnog sistema na određenom rastojanju i pod određenim uglom u odnosu na definisanu *pt* tačku. Argument *pt* predstavlja 2D ili 3D tačku, argument *ugao* predstavlja vrednost relativnog ugla u radijanima u odnosu na pozitivan smer X ose. Ugao raste u suprotnom smeru od smera kretanja kazaljke na satu, nezavisno od tekuće konstrukcione ravni. Argument *dist* predstavlja rastojanje do specificirane pt tačke. Ova funkcija prikazuje 2D ili 3D koordinate tačke, što direktno zavisi od toga kako je definisana *pt* tačka.

Command: (polar '(1 1 3.5) 0.7865 1.4142)
(1.99889 2.00109 3.5)

Command: (polar '(1 1) 0.7865 1.4142)
(1.99889 2.00109)

Command: (polar '(0 0) 0.55555 2.34567)
(1.99291 1.23713)

(pi)

Ovo nije funkcija AutoLISP-a, nego je konstanta π koja prikazuje vrednost 3.14159.

(prin1 [izraz] [fajl])

Ova funkcija štampa *[izraz]* sa ekrana i ponovo prikazuje *[izraz]*. Argument *[izraz]* može da bude bilo koji izraz, ali nije poželjno da to bude string. Ako postoji argument *[fajl]* (i indeks za fajl koji je otvoren za pisanje), onda će *[izraz]* biti zapisan u fajl i biće prikazan na ekranu. Ako se setuje:

Command: (setq a 234)
234

Command: (setq b '(a))
(A)

sledi štampanje i prikazivanje izraza:

Command: (prin1 'a)
AA

Command: (prin1 a)
234234

Command: (prin1 b)
(A)(A)

Command: (prin1 "Zdravo")
"Zdravo""Zdravo"

(princ [izraz] [fajl])

Ova funkcija je ista kao funkcija PRIN1.

(print [izraz] [fajl])

I ova funkcija je ista kao funkcija PRIN1.

(progn [izraz1] [izraz2]...)

Ova funkcija omogućava da više izraza deluju kao jedan, prikazujući poslednju vrednost. U tu svrhu koristi se IF funkcija, ali treba napomenuti da je funkcija IF limitirana na samo jedan *then* i jedan *else* parametar. Ovo ograničenje uvodi restrikcije u programiranje. Ako korisnik želi da funkcija IF izvrši više *then* i *else* parametara, onda je funkcija PROGN neupotrebljiva. Sledi jednostavan primer:

```
(if TEST
    (progn
        (setvar "orthomode" 1)
        "Yes"
    )
    (progn
        (setvar "orthomode" 0)
        "No"
    )
)
```

Ako je TEST *true*, onda se aktivira ortogonalno crtanje i prikazuje se *Yes* u komandnoj liniji, a ako je TEST *nil*, onda se ukida ortogonalno crtanje i prikazuje se *No* u komandnoj liniji.

(prompt [poruka])

Ova funkcija prikazuje navedenu poruku u komandnoj liniji gde se nalazi kursor miša i završava je sa *nil*. Treba napomenuti da se poruka tretira kao string. Na primer:

```
Command: (prompt "Nova vrednost: ")
Nova vrednost: nil

Command: (prompt "Sta je ovo? ")
Sta je ovo? nil
```

(quit)

Ova funkcija zatvara aktivnu aplikaciju, s tim što korisnik može da dobije poruku o zatvaranju aplikacije. Ova funkcija ima sličnosti sa EXIT funkcijom.

(quote [izraz])

Ova funkcija prikazuje *izraz* onako kako je ukucan. Treba napomenuti da *izraz* može biti napisan kao *'izraz*.

```
Command: (quote Pera)
PERA

Command: (quote Sta je ovo?)
; error: syntax error

Command: (quote (Sta je ovo?))
(STA JE OVO?)
```

Treba napomenuti da *'izraz* neće da radi ako se ovako ukuca direktno u AutoCAD-ovom komandnom promptu. To znači da ovakva sintaksa može da se koristi u LSP fajlovima, koji će se učitati i startovati pomoću određene funkcije.

AutoLISP (12)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovom R.

(read [string])

Ova funkcija prikazuje prvu listu ili prvi član iz *[string]*-a. Ono što treba napomenuti *[string]* ne može da sadrži prazan (blanko) prostor.

Command: (read "Halo")
HALO

Command: (read "Sta je ovo?")
STA

Command: (read "Sta_je_ovo?")
STA_JE_OVO?

(read-char [fajl)

Ova funkcija čita jedinstveni karakter koji se ukucava sa tastature ili iz fajla koji je specificiran kao *[fajl]* i prikazuje ASCII kôd koji predstavlja učitani karakter. Ako *[fajl]* nije specificiran, onda se posle sintakse

(read-char)

čeka na unos podataka sa tastature.

Command: (read-char)
A
65

(read-line [fajl)

Ova funkcija prikazuje string koji se ukucava sa tastature ili iz fajla koji je specificiran kao *[fajl]*.

Command: (read-line)
Sta je ovo?
"Sta je ovo?"

Command: (read-line)
Prikazuje sve sto se otkuca!!!!
"Prikazuje sve sto se otkuca!!!!"

(redraw [naziv] [mod])

Ova funkcija omogućava osvežavanje odgovarajućih entiteta unutar grafičke oblasti, gde je u sintaksi *[naziv]* naziv entiteta, a *[mod]* predstavlja brojeve od 1 do 4, što je definisano narednim nabrajanjem:

- **Mod = 1** – osvežava entitet sa eventualnim izmenama
- **Mod = 2** – osvežava entitet bez izmena
- **Mod = 3** – osvežava osvetljeni (naznačeni) entitet
- **Mod = 4** – osvežava neosvetljeni (nenaznačeni) entitet

(regapp [ime])

Ova funkcija registruje *[ime]* kao naziv aplikacije unutar programskog paketa AutoCAD. Program će da smesti aplikaciju unutar APPID tabele simbola. Ako je registrovanje završeno uspešno AutoCAD prikazuje naziv aplikacije, u suprotnom prikazuje *nil*.

```
Command: (regapp "DESIGNER-v2.1-124753")
"DESIGNER-v2.1-124753"

Command: (regapp "ADESK_4153322344")
"ADESK_4153322344"
```

(rem [broj1] [broj2] ...)

Ova funkcija deli BROJ1 brojem BROJ2 i prikazuje ostatak. Funkcija REM može da koristi i celobrojne i realne brojeve uz standardna pravila pretvaranja. Ako u sintaksi ima više od dva broja, onda ova funkcija deli prvi broj drugim, a onda njihov količnike deli trećim, i tako redom.

```
Command: (rem 12 4)
0

Command: (rem 12.0 4.)
0.0

Command: (rem 4 12)
4

Command: (rem 100 5 4.)
0.0

Command: (rem 26 7 2)
1
```

(repeat [broj] [izraz])

Ova funkcija izračunava IZRAZ onoliko puta koliki je BROJ i prikazuje vrednost poslednjeg izraza. BROJ predstavlja bilo koji pozitivan celobrojni broj. Na primer ako se A dodeli vrednost 10, a B dodeli vrednost 100 sintaksom

```
Command: (setq a 10)
10

Command: (setq b 100)
100
```

onda se sintaksom dobija rezultat 500 (posle izračunavanja A ima vrednost 50 , a B ima vrednost 500 i ta vrednost se prikazuje u komandnoj liniji).

```
Command: (repeat 4 (setq a (+ a 10)) (setq b (+ b 100)))
500
```

Ova funkcija radi isto i sa stringovima:

Command: (repeat 1003 "Ja" "on" "SVI")
"SVI"

(reverse [lista])

Ova funkcija obrće redosled u LISTI.

Command: (reverse '((A) b c))
(C B (A))

(rtos [broj] [mod] [preciznost])

Ova funkcija prikazuje string koji je difinisan kao BROJ i koji je usaglašen sa setovanim MOD-om, koji se prikazuje sa odgovarajućom PRECIZNOŠĆU i koji je usaglašen sa sistemskim promenljivama UNITMODE, DIMZIN, LUNITS, i LUPREC. Podržavane vrednosti MOD-a su:

1. naučni
2. decimalni
3. inženjerski (stope i decimalni inči)
4. arhitektonski (stope i fraktalni inči)
5. proizvoljne fraktalne jedinice

Vrednosti MOD i PRECIZNOST odgovaraju vrednosti sistemskih promenljivih LUNITS i LUPREC. Ako se izostave ovi parametri onda će biti upotrebljene odgovarajuće setovane vrednosti za ove promenljive.

Command: (rtos 17.5 1 4)
"1.7500E+01"

Command: (rtos 17.5 2 4)
"17.5"

Command: (rtos 17.5 3 4)
"1'-5.5\""

Command: (rtos 17.5 4 4)
"1'-5 1/2\""

AutoLISP (13)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovom S.

(set [simbol] [izraz])

Ova funkcija setuje vrednost SIMBOL-a, gde je SIMBOL navedeni naziv simbola, i prikazuje tu vrednost.

Command: (set 'a 6.8)

6.8

(setq [simbol1] [izraz1] [simbol2] [izraz2]...)

Ova funkcija setuje vrednosti SIMBOL1 na IZRAZ1, SIMBOL2 na IZRAZ2, itd. Treba napomenuti da se uvek prikazuje poslednji IZRAZ u nizu.

Command: (setq a 3)
3

Command: (setq a 3 b 4 c 5)
5

Command: (setq s "opisno")
"opisno"

(sin [ugao])

Ova funkcija prikazuje sinus ugla kao realan broj, gde je UGAO izražen u radijanima.

Command: (sin 1.)
0.841471

Command: (sin 0.0)
0.0

(sqrt [broj])

Ova funkcija prikazuje kvadratni koren BROJ-a kao realan broj.

Command: (sqrt 25)
5.0

Command: (sqrt 34.567)
5.87937

(strcase [string] [kako])

Ova funkcija čita string koji je dat STRING parametrom, i prikazuje kopiju sa svim alfabetskim karakterima pretvorenim u velika ili mala slova, što zavisi od drugog argumenta KAKO. Ako argument KAKO ne postoji ili teži ka nil, onda će svi karakteri biti pretvoreni u velika slova. Ako argument KAKO postoji i nije nil, svi karakteri će biti pretvoreni u mala slova.

Command: (strcase "Primer")
"PRIMER"

Command: (strcase "Primer" T)
"primer"

(strcat [string1] [string2]...)

Ova funkcija prikazuje string koji se dobija nadovezivanjem, jedan na drugi, stringova STRING1, STRING2, itd.

```
Command: (strcat "Ka" "rak" "ter")
"Karakter"

Command: (strcat "Pa" "ra" "me" "tar")
"Parametar"
```

(strlen [string])

Ova funkcija prikazuje dužinu STRING-a u karakterima, kao celobrojnu vrednost.

```
Command: (strlen "abcdefghs")
9

Command: (strlen "waesrweqarfdweFWAefwsaeWFaeWRfwsAwf")
35

Command: (strlen " ")
1

Command: (strlen "")
0
```

(subst [novo] [staro] [list])

Ova funkcija pretražuje LIST-u u potrazi za argumentom STARO i prikazuje kopiju LIST-e sa pridodatim argumentom NOVO na svakom mestu gde je pronađen argument STARO. Ako se ne pronađe argument STARO u LIST-i, ova funkcija prikazuje nepromenjenu listu.

```
Command: (setq Primer '(a b (c d) b))
(A B (C D) B)

Command: (subst 'qq 'b Primer)
(A QQ (C D) QQ)

Command: (subst 'qq 'z Primer)
(A B (C D) B)

Command: (subst '(qq ww ee) '(c d) Primer)
(A B (QQ WW EE) B)
```

(subst [string] [start] [duzina])

Ova funkcija prikazuje podstring STRING-a, počinjući od pozicije START karaktera i nastavljajući za DUZINA karaktera. Ako argument DUZINA nije specificiran, onda se podstring nastavlja do kraja STRING-a. Argumenti START i DUZINA (ako je prisutan) moraju imati pozitivne celobrojne vrednosti.

```
Command: (substr "abcdefg" 2)
"bcdefg"

Command: (substr "abcdefg" 2 4)
```

"bcde"

Command: (substr "abcdefg" 2 1)
"b"

AutoLISP (14)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima T i V.

(terpri)

Ova funkcija prelazi u novi red na ekranu i prikazuje nil. Funkcija TERPRI se ne koristi za I/O fajlove. Ako treba ubaciti novu liniju u fajl, onda je bolje da korisnik koristi funkcije PRINT, PRINC ili PRIN1.

(textpage)

Ova funkcija prebacuje grafički ekran u tekstualni, na sistemima sa jednim monitorom, osvežava sadržinu ekrana (uklanja kontrolne krstiće ako su odgovarajuće sistemske promenljive aktivne), i postavlja kursor miša na vrh ekrana. Kada se sve ovo završi, na ekranu se pojavljuje *nil*.

(textscr)

Ova funkcija prebacuje grafički ekran u tekstualni, na sistemima sa jednim monitorom. Kada se sve ovo završi, na ekranu se pojavljuje *nil*.

(trans [pt] [iz] [u] [disp])

Ova funkcija prevodi koordinate tačke (ili rastojanja) iz jednog koordinatnog sistema u drugi. Argument PT je lsita 3 realna broja koji mogu da predstavljaju 3D tačku ili 3D rastojanje (vektor). Argument IZ je kôd koji indicira iz kog koordinatnog sistema potiče PT, a argument U predstavlja kôd koji specificira odredišni koordinatni sistem. Opcioni DISP argument, ako je prisutan i različit od *nil*, specificira PT radije kao 3D rastojanje nego kao 3D tačku. Argumenti IZ i U ,mogu da budu:

- **0 – WCS** – to je referentni koordiantni sistem i svi ostali koordinatni sistemi se definišu relativno u odnosu na njega. WCS je jedini koordinatni sistem koji se nikada ne menja;
- **1 – UCS** – to je radni koordinatni sistem koji je usvojio korisnik radi lakšeg rada; i
- **2 – DCS** – Displaz Coordinate System je koordinatni sistem u kome se izvrše sve izmene u crtežu pre nego što se prikažu.

Command: (trans '(1.0 2.0 3.0) 0 1)
(2.0 -1.0 3.0)

Command: (trans '(1.0 2.0 3.0) 1 0)
(-2.0 1.0 3.0)

(type [item])

Ova funkcija prikazuje tip ITEM-a, gde tip može da bude jedan od sledećih:

- REAL – koordinate tačaka,
- FILE – indeks fajla,
- STR – string,
- INT – celobrojna vrednost,
- SYM – simboli,
- LIST – lista i funkcije korisnika,
- SUBR – interne AutoLISP funkcije,
- USUBR – funkcije AutoLISP-a koje je kreirao korisnik i učitao iz eksternog fajla,
- PICKSET – AutoCAD setovi za selekciju,
- ENAME – AutoCAD-ovi nazivi entiteta,
- PAGETB – tabele funkcija.

Na primer, ako se definiše:

```
(setq a 123 r 3.45 s "Halo!" x '(a b c))
(setq f (open "naziv" "r")
```

sledi:

```
Command: (type 'a)
SYM

Command: (type a)
INT

Command: (type s)
STR

Command: (type r)
REAL

Command: (type x)
LIST

Command: (type +)
SUBR
```

(ver)

Ova funkcija prikazuje string koji sadrži oznaku o verziji AutoLISP-a, Na primer, na sintaksu

```
Command: (ver)
```

sledi odgovor:

```
"Visual LISP 2004 (en)"
```

AutoLISP (15)

Pripremio Dragan Cvetković

Sledi nastavak funkcija unutar AutoLISP-a, s tim što treba napomenuti da će biti poređane po abecedi i da se nastavljaju funkcije čiji naziv počinje slovima W, X i Z. Ovim delom se završava priča koja je vezana za programski jezik AutoLISP.

(while [test-izraz] [izraz] ...)

Ova funkcija prevodi izvršava TEST-IZRAZ i, ako nije *nil*, onda izvršava IZRAZ i, posle toga, ponovo izvršava TEST-IZRAZ. Ovo se vrti u krug sve dok TEST-IZRAZ ne postane nil. Tada WHILE funkcija prikazuje poslednju vrednost. Na primer, ako je dato

```
(setq a 1)
```

tada će se:

```
(while (<= a 10)
    (funkcija a)
    (setq a (1+ a)) )
```

pozivati korisnikova FUNKCIJA deset puta, gde će se A setovati od 1 do 10. Na kraju će se prikazati vrednost 11 kao poslednja dobijena vrednost izraza.

(write-char [broj] [indeks fajla])

Ova funkcija upisuje jedan karakter na ekran ili u fajl koji je speificiran pomoću INDEKS FAJLA. Parametar BROJ je ASCII kôd karaktera koji će biti prikazan i ta vrednost će biti vraćena od strane funkcije.

```
Command: (write-char 67)
C67

Command: (write-char 88)
X88

Command: (write-char 77)
M77
```

(write-line [string] [indeks fajla])

Ova funkcija upisuje STRING na ekran ili u otvoreni fajl koji je opisan pomoću INDEKS FAJLA. Ako se ispisuje STRING na ekranu, mora se staviti između navodnika, a ako se ispisuje u fajl, navodnici nisu potrebni. Na primer, ako je FF odgovarajući INDEKS FAJLA za otvoreni fajl, tada će

```
(write-line "Test" f)
```

ispisati Test u fajl i vratiće "Test".

(xdroom [naziv])

Ova funkcija prikazuje veličinu slobodne memorije koja je dostupna za podatke entiteta, ili prikazuje *nil*.

(xdsize [list])

Ova funkcija prikazuje veličinu memorije koju je zauzila LIST-a.

(zerop [broj])

Ova funkcija prikazuje T, ako je BROJ ima realnu ili celobrojnu vrednost i teži ka 0, a u svakom drugom slučaju prikazuje *nil*.

Command: (zerop 0)
T

Command: (zerop 0.000)
T

Command: (zerop 0.00001)
nil

AutoLISP je poseban deo LISP programskog jezika, koji se isporučuje unutar programskog paketa za projektovanje AutoCAD. AutoLISP omogućava korisnicima AutoCAD-a da pišu i kreiraju makroe i funkcije u višem programskom jeziku, što je izuzetno povoljno za grafičke aplikacije. LISP je lagan za učenje i korišćenje i veoma je fleksibilan. LISP je predstavnik aplikativnih ili funkcijskih programskih jezika. U okviru njega ne postoje komande, već se sve svodi na izračunavanje raznih funkcija.