

POUŽITÍ JAZYKA AUTOLISP

Ing. Karel Mareš

AutoLISP je programovací jazyk, který vychází z LISPu, přesněji z Common LISPu. Zachovává jeho syntaxi, základní používané struktury i standardní funkce. Rozdílná je však oblast aplikace. AutoLISP je programovou podporou grafického systému AutoCAD, do něhož je integrován. S tím souvisí následující odlišnosti - uživateli je dána k dispozici nepoměrně větší množina standardních funkcí, jsou rozšířeny typy proměnných (např. typ file-descriptor, selection set up ap.).

AutoLISP nelze provozovat samostatně, jako interpretační jazyk je k dispozici až po spuštění grafického editoru AutoCADu. Pro svoji činnost tedy vyžaduje počítač se systémem AutoCAD a 640 Kbytů paměti, kam jsou ukládány proměnné, standardní a uživatelem definované funkce.

Při spuštění AutoLISPu jsou alokovány dvě paměťové oblasti. První, nazývaná "heap" (hromada), se používá pro ukládání všech funkcí a proměnných; čím máte víc proměnných a funkcí nebo čím jsou funkce složitější, tím je zaplněno více paměti "heap". Druhá oblast, nazývaná "stack" (zásobník), ukládá argumenty funkce a mezivýsledky. Zaplnění této paměti tedy závisí na složitosti našich definovaných funkcí (např. počtu argumentů nebo značné rekurze).

Bez znalosti systému AutoCAD nemá použití AutoLISPu praktický význam. Klíčovou úlohu přitom má pojem entita, což je základní geometrický útvar (bod, úsečka, kružnice, kruhový oblouk atd.), s kterým AutoCAD jako vektorový systém pracuje. Každý výkres v AutoCADu se skládá z takových entit, přičemž informace o těchto entitách se ukládají do grafické databáze, což je datová struktura, kterou si AutoCAD vytváří v paměti. Záznam o entitě v grafické databázi se skládá z jejího geo-

metrického popisu (např. x,y-ové souřadnice obou konců úsečky) a z jejích vlastností (barva, typ čáry, kterou je kreslena, hladina výkresu, ve které leží).

Standardní funkce AutoLISPu můžeme rozdělit na dvě základní skupiny. První je tvořena obecnými funkcemi, převzatými z LISPu, druhá obsahuje speciální funkce, známé jen v AutoLISPu. Do první skupiny patří např. funkce aritmetické, logické, relační, goniometrické, funkce pro práci s řetězci, funkce pro práci se seznamy, konverzní funkce, funkce vstupu a výstupu, funkce pro ladění programu, funkce pro větvení programu, funkce pro vytváření cyklů.

Druhá skupina obsahuje funkce, umožňující přístup do grafické databáze a manipulaci s existujícími entitami, generování nových entit, funkce pro usnadnění interakce s operátorem, funkce pro měření úhlů a délek.

Při vytváření aplikačních programů v AutoLISPu můžeme rozeznat dva hlavní směry. Prvním je práce nad hotovým výkresem, resp. nad jeho grafickou databází, kdy se extrahují vlastnosti určitých entit za účelem pořizování kusovníků, soupisek materiálu apod., případně se vlastnosti určitých entit mění substitucí jiných hodnot, čímž dosáhneme potřebných změn na výkrese. Podobně na extrahování geometrie entit můžeme založit program postprocesoru, generujícího s výrobního výkresu nějaké součásti v AutoCADu kód pro NC stroj, vyrábějící tuto součást. Při těchto úlohách se grafická databáze sekvenčně prochází pomocí speciální funkce ENTNEXT, která vrací jména jednotlivých entit. Jméno entity potom použije další speciální funkce ENTGET, která vrací ve formě seznamu všechny informace o dané entitě.

Když tedy například v AutoCADu sestrojíme úsečku z bodu 1,1 do bodu 2,2, vrátí kombinace těchto AutoLISPovských funkcí (ENTGET (ENTNEXT)) následující seznam :

```
((-1 . <Entity name:60000018>) (0 . "LINE") (8 . "0")  
(10 1.0 1.0) (11 2.0 2.0)),
```

kde první subseznam obsahuje pod kódovým označením -1 jméno entity, neboli číslo, pod kterým se tato entita nachází v grafické databázi, druhý subseznam je tečka-číslice, udávající typ entity, třetí udává hladinu, ve které entita leží, čtvrtý a pátý udává souřadnice počátku a konce úsečky. Nad takovým seznamem potom mohou pracovat standardní LISPovské funkce jako CAR, CDR, ASSOC, SUBST atd.

Chceme-li například získat souřadnice počátku takové úsečky, použijeme kombinaci funkcí

```
(cdr (assoc 10 (entget (entnext))))
```

která vrací seznam (1.0 1.0).

Druhým směrem je přímo generování nových entit AutoLISPovským programem. Za tímto účelem je definována funkce COMMAND, jejímiž argumenty jsou přímo příkazy AutoCADu tak, jako kdyby byly dávány ručně operátorem. Například výraz (COMMAND "LINE" "1,1" "2,2" "") sestrojí úsečku z bodu 1,1 do bodu 2,2. Tímto způsobem může tedy AutoLISPovský program automaticky kreslit. Rozvinutí této techniky vede k tzv. parametrickému programování, což je vytváření programů pro kreslení tvarů se stejnou geometrií, ale různými rozměry, závislými na vstupních parametrech.

S AutoLISPem můžeme pracovat jednak tak, že výrazy vkládáme přímo z klávesnice na návěští Command: AutoCADu, přičemž stisknutí klávesy ENTER způsobí ohodnocení výrazu. Levá otevřená závorka odliší výraz AutoLISPu od příkazu AutoCADu. Lze tak pohodlně provádět aritmetické výpočty nebo ověřovat činnost určitých funkcí.

Druhou možností je použít textového editoru pro psaní a uložení AutoLISPovského programu do souboru. Tento soubor je potom nutno načíst do AutoCADu pomocí funkce LOAD.

Velmi těsnou vazbu mezi AutoLISPem a AutoCADem je možno ukázat na některých speciálních velmi výkonných funkcích, jako jsou např. funkce pro vytváření tzv. výběrových množin VM

(selection set). Entity AutoCADu, obsažené ve VM, jsou přímo přístupné pomocí speciálních výrazů AutoLISPu.

Použití VM si ukážeme na několika AutoLISPovských aplikacích. Jednou z nich může být generování jednoduché soupisky materiálu a verifikace obsahu výkresu. Nejdříve stručný přehled základních informací o VM, které budeme potřebovat.

VM je množina (souhrn) entit AutoCADu, vytvořená pro účely manipulace s nimi. Příkazový systém AutoCADu pak může používat VM jako celkem např. při editování této skupiny entit. S hlediska AutoLISPu mohou být VM vázány k symbolům, s nimiž pracují jak běžné příkazy AutoCADu, tak speciální výrazy AutoLISPu, určené pro práci s VM.

VM tvoří speciální datový typ AutoLISPu, nejsou to tedy běžné seznamy a lze s nimi manipulovat jedině pomocí speciálních vnitřních funkcí AutoLISPu. Každá taková funkce začíná na "ss" (selection set). Tyto základní VM funkce mohou být různě kombinovány pro vytvoření složitějších funkcí, definovaných uživatelem pro řešení složitějších problémů.

Pro vytvoření nových VM se v AutoLISPu použije výrazu (SSADD) nebo (SSGET). Podle parametrů výrazu (SSADD) se buďto vytvoří nová VM nebo se přidá entita k existující VM. (SSGET) vytvoří novou VM a vrátí hodnotu pro vázání k symbolu. Pro vázání vrácené hodnoty k symbolu je třeba použít SETQ. Např. výraz

```
(SETQ SS1 (SSGET))
```

vytvoří VM č. 1.

(SSGET) vytváří nové VM různým způsobem v závislosti na specifikovaných parametrech. Jestliže není dodán žádný, (SSGET) žádá operátora o dodání objektů pro VM zobrazením výběrového kurzoru (object pickbox) a dotazem "Select object:". Současně jsou k dispozici všechny výběrové módy "window, cross, pick, snap, remove, add". Pokud místo dodání dat stiskneme ENTER, (SSGET) proces se ukončí a je vráceno jméno VM.

Když pracuje s parametry, (SSGET) automaticky vytvoří novou VM. U všech parametrů kromě jednoho se vyžaduje dodání seznamu bodů, které definují oblast pro výběr objektů. Tyto body musí dodat operátor dříve, než započne vlastní automatický výběr objektů, které připadnou do VM. Tyto body lze zadat v rámci specifikace parametrů např. pomocí výrazů (GETPOINT) a (GETCORNER). Příklad aplikace těchto výrazů je uveden ve Výpisu 1.

Dvě funkce v tomto příkladu spolupracují při vytvoření nové VM, která bude obsahovat entity, jejichž typ zadá operátor. Obě makra tedy slouží k manipulaci s VM a lze je použít obecně i v jiných aplikacích. Předpokládá se dostatek místa pro dočasné vytvoření dvou nových VM.

První funkcí, definovanou ve výpisu 1, je "ENTHERE". Po zadání souřadnic bodu ve formě seznamu jako její jediný parametr vrací "ENTHERE" jméno entity, umístěné ve specifikovaném bodu nebo nil, pokud se zde žádná entita nenalézá. Funkce začíná vytvořením dočasné VM použitím (SSGET) výrazu s volbou jednoho bodu. Dále se testuje, zda (SSGET) vrací nil. V případě, že není nalezena entita, podmínka vrací nil a tím i celá funkce vrací nil. Jinak se získá jméno entity, která je prvním členem dočasné VM a je vráceno jako výsledek funkce "ENTHERE".

Druhá funkce "SSGETP" vytváří VM podobně jako to dělá subr (SSGET) a jediným rozdílem. V našem příkladě funkce je poskytován pouze přímý výběr entit. (window a crossing není dovoleno). "SSGETP" začíná s (SSADD) výrazem pro vytvoření nové VM a (SETQ) pro stanovení inicializačního dotazu na volby parametrů. (INITGET) výraz potom inicializuje vstupní systém pro vstup znaků "R" a "U". Tyto volby budou použity k nastavení běžného módu systému buď pro přidávání entit k VM nebo k jejich odstraňování.

Cyklus (WHILE) se spustí a bude se opakovat vždy, když operátor vybere bod nebo vloží platné klíčové slovo. K říze-

ní testování a ohodnocení vstupu operátora je inicializován výraz (COND) se sérií testovaných možností. Jestliže byla operátorem vložena volba "R", změní se dotaz na "Remove" a seznam voleb je modifikován, aby akceptoval volbu "A". Tímto způsobem je běžný mód systému přestaven na odstraňování entit z VM. Jestliže je operátorem vybrána volba "A", dotaz se opět změní na "Select" a seznam vstupních voleb se nastaví na "R" a "U". Běžný mód je potom přestaven na přidávání entit k VM. Pokud byla vložena volba "U", (SSGETP) funkce zruší předchozí výběr a odstraní entitu z VM. Je to dosaženo sérií vzájemně vnořených subrutin (SSxxx), které provádí tuto operaci.

Jestliže vstup od operátora není žádná z možností "R", "A", "U", výraz (GETPOINT), použitý uvnitř výrazu (WHILE) vrátí seznam souřadnic bodu, data tohoto bodu jsou dodána funkcí "ENTHERE", která vrací jméno entity, která je přidána nebo odstraněna z VM. Výraz (COND) určí, zda je běžný mód nastaven na odstraňování entit nebo jejich přidávání. Pokud je nastaven na odstraňování, výraz (SSDEL) se provede, aby odstranil danou entitu z VM. Jestliže je běžný mód nastaven na přidávání (add), výraz (SSADD) se provede, aby přidal entitu do VM. V případě, že nebyla vybrána žádná entita, zobrazí se operátorovi chybové hlášení. Po dokončení všech testů se provede výraz (INITGET) pro nastavení vstupních systémů přijímací voleb (ze seznamu voleb OPTS) pro další průchod cyklem (WHILE). Když je vybrána a zpracována poslední entita, cyklus (WHILE) se ukončí a VM je vrácena jako výsledek funkce.

```

; ++++++ VÝBĚROVÉ MNOŽINY, Výpis č. 1 ++++++
;
; Pomocné funkce pro manipulaci s entitami a výběrovými
; množinami.
;   ENTHERE vrací jméno entity v zadaném bodě.
;   SSGETP vrací výběrovou množinu, vytvořenou pouze přímým
;       výběrem entit.
; ++++++
(defun ENTHERE (PNT / see)
  (setq sse (pset PNT))
  (cond
    ((null sse) nil)

```

```

      (t (ssname sse 0))))
;
(defun SSGET ( / Ssp pl opts en tmp)
  (setq ssp (ssadd) tmp "Select:" opts "R U")
  (initget 0 opts)
  (while )setq Fl )getpoint tmp))
  (cond
    ((=pl "R") (setq tmp "Remove:" opts "A"))
    ((=pl "A") (setq tmp "Select:" opts "R U"))
    ((=pl "U")
      (ssdel (ssname ssp (1- (sslength ssp))) ssp))
    (t
      (setq en (enthere pl))
      (cond
        ((and en (= tmp "Remove:"))(ssdel en ssp))
        ((and en (= tmp "Select:"))(ssadd en ssp))
        (t (prompt */nNe entity found! "))))))
  (initget 0 opts)
  (setq ssp ssp)
)

```

AutoCAD verze 9 a vyšší poskytuje nový rozměr funkcí (SSGET), kterým je možnost zvolit tzv. SEZAM FILTRŮ (mód "I"). Zadáním volby "I" v seznamu parametrů může být grafická databáze stávajícího výkresu v AutoCADu rychle prohlížena a nová VM může být vytvořena z určitých entit na výkresu. Abychom specifikovali typy entit, které mají být přidány do nové VM, musíme seznam vlastností entit jako parametr (SSGET) dodat. Tento seznam se často nazývá seznam filtrů, protože slouží jako filtr, propouštějící ty objekty, které aplikace vyžaduje.

Jednoduchým použitím volby seznamu filtrů v (SSGET) lze získat výkonný prostředek pro extrahování objektů. Např. následující výraz vytvoří novou VM, která obsahuje pouze entity INSERT se jménem bloku "TEST1":

```
(SETQ SSI (SSGET "I" ((0. "INSERT")(2. "TEST1"))))
```

Je třeba pamatovat na to, že seznam filtrů se musí podobat seznamu vlastností entit, jejichž výběr se provádí. To znamená, že musí být použita stejná kódová čísla (group codes), a to, co je v seznamu vlastností (který vrací ENTGET) např. tečka-dvojicí, musí být tečka-dvojicí i v seznamu filtrů.

Podívejme se na činnost demonstračního příkladu, který využívá seznamu filtrů ve výrazu (SSGET). Jedná se o program ve Výpisu č. 2. Slouží pro požízování výpisů materiálu (kusovníků). Funkce "TYPISUJ" je zde definována, aby prohledávala aktuální výkres a vypsalá počty čar, nalezených na hladině "TYCE". Během zjišťování úhrnných počtů čar jsou jejich délky srovnávány se seznamem předdefinovaných délek. Čára, která má délku jinou, než jsou ty uvedené v seznamu, je přesunuta na hladinu "OPRAV".

TYPISUJ začíná použitím výrazu "SSGET" a volbou "X" pro použití seznamu filtrů, aby vytvořila novou VM. Kritéria pro hledání jsou definována v seznamu parameterů, který má dva komponenty. První znamená, že vybraná entita musí být typu "LINE" a druhý, že tato "LINE" musí být umístěna na hladině "TYCE". Když jsou zadána tato kritéria pro vyhledávání, vytvoří se nová VM a je vázána k symbolu SSL (jako první VM v této aplikaci). Určí se celková délka této VM a celkový počet LINE, které byly nalezeny na výkrese, se zobrazí na monitoru. Před testováním legitimní délky první nalezené LINE se nastaví několik proměnných. V proměnné CNELIST je uložen asociativní seznam, kam se ukládají počty čar jednotlivých délek. Na počátku je inicializován na nulové hodnoty. Každý element asociativního seznamu obsahuje jednak délku čáry a jednak počítadlo výskytu této délky (jsou použity tečka-dvojice a čísla integer). Tím se šetří prostor pro uzly (node space), protože se k jejich reprezentaci vyžaduje méně stavebních buněk (cons cells) a menšího prostoru v paměti (heap space). Další počítadlo CNT se používá pro počítání objektů ve VM a je inicializováno na -1. Poslední počítadlo je v proměnné CNTDEL a je nastaveno na 0.

Cyklus (REPEAT) proběhne tolikrát, kolik entit obsahuje VM. Každý průchod cyklem způsobí, že výraz ENTGET extrahuje další entitu z VM a vrací seznam jejích vlastností. Jméno entity, které je dáno jako element s pořadím CNT + 1 ve VM je dodáno jako parametr výrazu (ENTGET). Výraz (DISTANCE) potom ze seznamu vlastností entity určí vzdálenost mezi jejím počátečním a koncovým bodem. Výchozí a koncový bod LINE mají kódová čísla (group codes) 10 a 11. Délka se převede na integer přičtením malé zaokrouhlovací hodnoty. Tento malý korekční faktor je použit, protože matematika reálných čísel na počítači může být zdrojem chyby zaokrouhlení. AutoCAD totiž pracuje běžně s reálnými čísly a přesností minimálně na 11 míst v závislosti na typu počítače.

Uvnitř cyklu (REPEAT) se inicializuje podmínka, která porovnává vypočtenou délku se seznamem předdefinovaných délek. Výraz (ASSOC) porovnává hodnoty délek a vrací nil vždy, když vypočtená vzdálenost neodpovídá žádné hodnotě, uložené v proměnné CNTLIST. Když (ASSOC) zjistí shodu, počet výskytů této délky se zvýší. Výraz (SUBST) dosadí nový počet k odpovídající délce. Počítadlo u zjištěné délky se zvýší o 1 a uloží se zpět do seznamu.

Když vypočtená vzdálenost neodpovídá žádné s přednastavených délek v CNTLIST, další výraz (SUBST) změná hladinu dané entity na "OPRAV". Počítadlo CNTDEL se pak zvýší o 1. Když je takto zpracována celá VM, počet výskytů čar jednotlivých délek se zobrazí pomocí cyklu (FOR EACH) a výrazu (PRINT). Nakonec se zobrazí celkový počet entit, přesunutý do "OPRAV".

Tento jednoduchý příklad výpisu materiálů může být samozřejmě dále rozvíjen tak, aby odpovídal potřebám uživatele. Kombinací volby filtrů u výrazu (SSGET) a schopností AutoLISPu ukládat výsledky do souborů, lze např. vytvořit podklady pro systém cenových odhadů nebo pro databázi či tabulkové zpracování dat (spreadsheet). Tímto příkladem jsme pouze nahlédli do nekonečných možností ve světě AutoLISPu.

```

; ++++++ VYBEROVÉ MNOŽINY, výpis č. 2 ++++++
;
; TYPIZUJ prohlíží všechny "LINE" na hladině "TYCE"
; a zjišťuje jejich délku: Když se délka rovná některé
; z předvolených délek, zvýší se počítadlo těchto délek.
; Jinak se tato "LINE" přesune na hladinu "OPRAV", kde
; bude opravena.
;
; ++++++

```

```

(defun C:TYPIZUJ ()
  (setq SSL (ssget "X" ((0 . "LINE") (8 . "TYCE"))))
  (print (sslength SSL))
  (prompt "entit nalezeno")
  (setq CNTLIST ((12 . 0) (24 . 0) (48 . 0) (96 . 0)))
  CNT -1
  CNTDEL 0)
  (repeat (sslength SSL)
    (setq EL (entget (ssname SSL (setq CNT (1+ CNT))))))
    DS (fix (+ 0.000001
              (distance (cdr (assoc 10 EL))
                        (cdr (assoc 11 EL))))))
    (cond
      ((setq AA (assoc DS CNTLIST))
        (setq CNTLIST (subst (cons DS (1+ (cdr AA)))
                              AA CNTLIST)))
      (t (setq EL (subst (cons 8 "OPRAV") (assoc 8 EL) EL)
                    CNTDEL (1+ CNTDEL))
        (entmod EL))))
    (prompt "/n(Delka Pocet)")
    (foreach MBR CNTLIST
      (print MBR)
    )
  (print CNTDEL)
  (prompt "entit presunute na hladinu OPRAVI")
  (princ)
)

```

Literatura :

- /1/ AutoLISP Programmers Reference Manual, Autodesk Ltd**
- /2/ AutoCAD Reference Manual, Autodesk Ltd**
- /3/ CAD User, The Independent Guide to AutoCAD in the UK and Europe (měsíčník)**
- /4/ CADENCE, Using AutoCAD in the Professional Environment (měsíčník, USA Kanada)**
- /5/ LISP, Winston and Horn, Addison-Wesley Publishing Company, second edition**
- /6/ AutoLISP in Plain English, George U. Head, Ventana Press**