

Turbo Pascal

Nyelvi segédkönyv
a

7.0

verzió ismertetésével

REÁL

Csoma – Borbély

**PASCAL NYELVI
SEGÉDKÖNYV**

**A TURBO PASCAL 7.0
FEJLESZTŐI RENDSZER
ISMERTETÉSÉVEL**

**REÁL
1993**

Ellenőrizte: Kis Balázs

Felelős szerkesztő: Borbély Viktor

©Csoma Péter Gergely, Borbély Viktor

A szerzők ezúton szeretnének köszönetet mondani a tatabányai **ALFADAT Kft**-nek azért a számítógépért, amelyen ez a könyv elkészülhetett.

Múzsák Kiadó Kft.

1993

ISBN 963 564 515 5

Felelős kiadó: a Múzsák Kiadó Kft. igazgatója

Készült a Múzsák Kiadó Kft. nyomdájában

Műszaki vezető: Horváth László

Megjelent 10 A/5 ív terjedelemben

TARTALOMJEGYZÉK

• Tartalomjegyzék.....	3
• Bevezető.....	5
• Magyarázatok.....	7
• Turbo Pascal 7.0 a DOS és a védett-DOS üzemmódban.....	13
• A fejlesztői rendszer újdonságai	17
• Fejlesztői rendszer és menürendszer	21
• Szövegszerkesztő	51
• Adattípusok	57
• Fenntartott szavak.....	63
• Fordítási direktívák.....	73
• Unitok.....	79
CRT.....	79
DOS	85
GRAPH.....	99
OVERLAY	132
PRINT	135
SYSTEM.....	136
• Más verziókból átvett UNIT-ok.....	171
• Hibakódok.....	177

BEVEZETŐ

A *Pascal* nyelvű programozás hazánkban a legelterjedtebb programkészítési mód. E programnyelv kiszolgáló piaca nem olyan változatos, mint más – például a C vagy a C++ – nyelveké.

A Turbo Pascal legújabb változata a 7.0 verziószámot viseli. Ezen belül jelenleg két fajtája kapható: a leginkább elterjedt *Turbo Pascal 7.0*, illetve a *Borland Pascal 7.0*. E két programcsomag között jelentős különbség van. Segédkönyvünk a *Turbo Pascal 7.0*-val foglalkozik, ám annak csak alapvető részeivel. (Amennyiben e könyvünk sikeres lesz, a *Turbo Pascal 7.0* többi részét is közre adjuk hasonló, referencia szerű formában.) Ez a programcsomag az eddigi három Pascal fejlesztés (DOS, „védett-DOS”, és Windows) közül kettőt, a DOS-t (TURBO.EXE) és a védett-DOS-t (TPX.EXE) tartalmazza.

Nem célunk e fejlesztőrendszer minősítése; minősíti a *Turbo Pascal*t használók népes tábora. Tény azonban, hogy ez a nagyszerű program is majdhogynem hemzseg a – szerencsére csak zavaró – hibáktól. Ezeket a hibákat (pl. a segítőrendszer hiányosságai és hibás információi) próbáljuk könyvünkkel könnyebben elkerülhetővé tenni.

Könyvünket nem a kezdő, hanem a (közép) haladó programozóknak készítettük. Tehát ebből a könyvből nem lehet megtanulni „*paszkálul*” programozni!!! Ezzel a témával sok más könyv foglalkozik (melyek minősítésére megint csak nem térünk ki). Ez a könyv a *Turbo Pascal 7.0*-át rendszeresen használóknak készült, s ez szerkezetén és nyelvezetén is látszik.

A *Turbo Pascal 7.0* legfontosabb újításai (itt a már ismertebb 6.0-hoz viszonyítunk) nem szembetűnőek, ám lényegesekek; rendszeresen programozóknak nagy segítséget jelenthetnek.

A *Turbo Pascal 7.0* fordítója jóval gyorsabb lett, mint 6.0-ás elődjéé. A lefordított programok sebessége viszont nem (vagy nem sokat) nőtt.

Újdonság a TPX-védett mód, és az ebben használható *ObjectBrowser* (objektum tallózó) és a mindkét programban megtalálható különböző (azonnal elérhető) funkciók saját, lokális menüje.

Ötletes újítás volt a színezhető szövegszerkesztő, amellyel a fordító különböző nyelvi elemeket emel ki már a forrásszöveg gépelése közben, ezzel segítvén a forrásszöveg áttekintését.

Újdonság a külső programokhoz tartozó üzenetablak (*messages*).

Maga a Pascal nyelv is megújult, pontosabban bővült. Néhány alap UNIT (vagy pontosabban eljárás-egység) nagyobb lett, sőt, két új UNIT is megjelent (*Strings*, *WinDOS*). Fontosabb bővítésként a *Turbo Vision 2.0* említhető, mellyel könyvünk ezen kötete nem foglalkozik.

Ez a – talán mondhatjuk azt, hogy első – kötet a *Turbo Pascal 7.0* következő részeivel foglalkozik:

- a *Turbo Pascal 7.0* menürendszere,
- a szövegszerkesztő,
- a pascal nyelv fenntartott szavai,
- a fordítási direktívák,
- adattípusok,
- a legtöbbet használt hat eljárás-egység,
- futási hibakódok.

MAGYARÁZATOK

A SZINTAXIS ÉS REFERENCIA LEÍRÁSÁBAN ÁLTALÁNOSAN ALKALMAZOTT JELÖLÉSEK ÉS RÖVIDÍTÉSEK

Alt

Az *Alt* billentyű. Általában a menüsor aktivizálására használjuk.

Esc

Az *Esc* (*Escape*) billentyű. Általában az éppen futó alkalmazás működésének megszakítására használjuk.

Tab

A *Tab* billentyű. Általában ablakok és bejegyzések között ugrálhatunk vele az óramutató járásával megegyező irányban. Az ellenkező irányú mozgás a **Shift-Tab** billentyűkombinációval érhető el.

End

Az *End* billentyű. A szerkesztett sor utolsó karakterére ugrik.

Home

A *Home* billentyű. A szerkesztett sor első karakterére ugrik.

Backspace

A *Backspace* billentyű. A kurzortól balra eső karaktert törli, a kurzor egy pozícióval balra mozdul. A *Bksp* rövidítés is használatos.

Del

A *Del* (*Delete*) billentyű. A kurzortól jobbra eső karaktert törli, a kurzor a helyén marad.

Ins

Az *Ins* (*Insert*) billentyű. A beszúró és a felülíró üzemmód váltására használjuk.

PgUp

A *PgUp* (*Page Up*) billentyű. Egy oldalnyit lapoz fölfelé az aktuális ablakban.

PgDn

A *PgDn* (*Page Down*) billentyű. Egy oldalnyit lapoz fölfelé az aktuális ablakban.

Shift-x

A Shift és a megadott billentyű együttes lenyomása.

Alt-x

Az Alt és a megadott billentyű együttes lenyomása.

Alt-x y

Az Alt billentyű folyamatos nyomva tartása mellett először az *x*, majd az Alt billentyű elengedése után az *y* billentyűt kell megnyomni.

Ctrl-x

A *Ctrl* (*Control*) billentyű lenyomása mellett nyomjuk meg az adott billentyűt.

Fn

Valamely funkcióbillentyű ($n=2-12$).

F1

Bárhol vagyunk, az F1 funkcióbillentyű megnyomásakor megjelenik a *Turbo Pascal 7.0* segítőrendszere, a *Help* ablak, és benne az aktuális téma leírása angolul.

Ctrl-Alt-Del

A három billentyű együttes leütésével újraindíthatjuk a rendszert (*melegindítás*).

MAGYARÁZATOK A MENÜRENDSZER ÉS A SZINTAXIS LEÍRÁSÁHOZ

MENÜRENDSZER

Go to line number...

Alt-S G

Az állomány adott sorára ugrik. A sor számát az *Enter new line number* sorban kell megadni, amelyben a korábban megadott értékek is előhívhatók.

A menüpont sorában jobb oldalt az adott funkciót elindító billentyűkombináció áll.

A menürendszer leírásában minden angol szöveget (üzeneteket, kifejezéseket, programneveket) *dőlt betűkkel* szedtünk. Dőlt betűsek még a felhasználó által megadható paraméterek nevei is.

SZINTAXIS

A menürendszer leírását kivéve minden fejezet a Turbo Pascal nyelv elemeit ismerteti. A nyelvi objektumok szintaxisának leírásánál az alábbi jelöléseket alkalmaztuk:

Félkövér betű: Az így megadott nyelvi elemet (fenntartott szót, eljárásnevet stb.) változtatás nélkül, a leírt formában kell a programba vagy a DOS-parancssorba írni.

Például: **WRITE**[(*kifejezés*[,*kifejezés*]...)];

Dőlt betűvel szedtük a helyettesíthető paraméterek neveit. Ezek helyére a programban vagy a DOS-parancssorban a megfelelő típusú (formális vagy konkrét) adatot kell írni.

Például: *változónév*:=*kifejezés*;

A szintaxis leírásakor (nem félkövér) **szögletes zárójel** ([]) között tüntettük fel azokat a nyelvi elemeket, amelyeket az adott utasítás, parancs, függvény, fordítási direktíva vagy DOS-parancssor használata-

kor nem kötelező megadni. (A félkövér szögletes zárójeleket és a közöttük levő részeket *be kell írni!*)

Például: **TPX.EXE** [/Cállomány][/D[+|-]] [/Eméret] [/Fméret]
 [/G[+|-]] [/L[+|-]] [/N[+|-]] [/Oméret] [/P[+|-]]
 [/R[+|-]] [/Súf] [/T[+|-]] [/Wméret] [/X[+|-]]
 [/Y[+|-]] [állomány]

A (nem félkövér) **szögletes zárójelek** ([]) között leírt, egymástól függőleges vonallal (|) elválasztott nyelvi elemek közül egyet lehet megadni, de azt sem kötelező (lásd a fenti példát). (A félkövér kapcsos zárójeleket és a közöttük levő részeket *be kell írni!*)

A (nem félkövér) **kapcsos zárójelek** ({ }) között felüntetett, egymástól függőleges vonallal (|) elválasztott nyelvi elemek közül egyet és csak egyet kell beírni az adott utasítás, eljárás, függvény, fordítási direktíva vagy DOS-parancssor használatakor (A félkövér kapcsos zárójeleket és a közöttük levő részeket *be kell írni!*)

A Turbo Pascal beépített vagy könyvtári (szabványos eljárásrendszerben, *unitban* levő) **függvényeinek** leírásakor megadtuk a függvény által visszaadott adat típusát is. Függvények csak kifejezésben, illetve értékadás jobb oldalán szerepelhetnek, önálló utasításként nem (ezt technikai okokból nem jelöltük).

Példa: **PTR** (szegmens, eltolás) : **POINTER**

A DOS-BAN ÉS A TURBO PASCAL 7.0-BAN HASZNÁLHATÓ SZABVÁNYOS PERIFÉRIANEVEK

AUX	Az első soros port.
COM n	Az n -edik soros port.
CON	Bemenete a billentyűzet, kimenete a képernyő.
LPT n	Az n -edik párhuzamos port.
NUL	Üres periféria. (Csak írható.)
PRN	Az első párhuzamos port.

A TURBO PASCAL 7.0 A DOS- ÉS A VÉDETT-DOS-ÜZEMMÓDBAN

A TURBO PASCAL 7.0 A DOS- ÉS A VÉDETT-DOS-ÜZEMMÓDBAN

A *Turbo Pascal 7.0* segítségével a PC valós és (a csak 80286-os vagy fejlettebb gépeken működő) védett DOS-üzemmódban is programozható. (Az *MS-Windows* védett üzemmódjában nem, arra a *Borland Pascal 7.0-t* kell használni.)

Számítógépünk valós üzemmódú programozásához a TURBO.EXE, a védett DOS-üzemmódban a TPX.EXE programot kell elindítani.

A TURBO.EXE elindításához elég a hagyományos 640 kilobyte, a TPX.EXE-hez azonban legalább 80286-os processzor és 2 megabyte memória szükséges.

A TPX.EXE tökéletes működéséhez memóriakezelő segédprogramra is szükség van (DPMILOAD.EXE, illetve DPMI16BI.OVL), melyek telepítése a BIN alkönyvtárból indítható DPMIINST.EXE programmal végezhető el.

A TURBO.EXE és TPX.EXE működése a DOS-parancssorbéli indításkor paraméterezhető. Ezek a paraméterek azonos hatásúak az *Options* menü *Environment* almenüjének *Startup* menüpontjában beállítható opciókkal, illetve néhány más direktívával. A kapcsolók alapértelmezése a bekapcsolt állapot (+).

TURBO.EXE [/Cállomány]/[D[+|-]]/[Eméret]/[G[+|-]]
 [/L[+|-]]/[N[+|-]]/[Oméret]/[P[+|-]]/[R[+|-]]
 [/Sút]/[T[+|-]]/[Wméret]/[X[+|-]][állomány
 [állomány]...]

TPX.EXE [/Cállomány]/[D[+|-]]/[Eméret]/[Fméret]
 [/G[+|-]]/[L[+|-]]/[N[+|-]]/[Oméret]/[P[+|-]]
 [/R[+|-]]/[Sút]/[T[+|-]]/[Wméret]/[X[+|-]]
 [/Y[+|-]][állomány [állomány]...]

(folytatás a következő oldalon)

(folytatás az előző oldalról)

/Cállomány

A megadott nevű konfigurációs állomány betöltése. Alapértelmezés: TURBO.TP (TURBO.EXE hívása esetén), illetve TPX.TP (TPX.EXE hívásakor).

/D[+|-]

Kétképernyős üzemmód (+: bekapcsolva, -: kikapcsolva).

/Eméret

A Turbo Pascal szövegszerkesztője számára fenntartandó memóriaterület nagysága kilobyte-ban (28-128). Az alapértelmezés 28 kilobyte.

/Fméret

Csak a TPX.EXE paramétereként használható. A cserállomány (*swap file*) méretének megadása kilobyte-ban.

/G[+|-]

A grafikus memória elmentésének engedélyezése, illetve megtiltása.

/L[+|-]

Folyadékkristályos (LCD) képernyő használatának jelzése (+: igen, -: nem).

/N[+|-]

Havazás elleni védelem CGA képernyőkön (+: bekapcsolva, -: kikapcsolva).

/Oméret

Az átlapoló (*overlay*) alrendszer számára lefoglalandó memóriaterület nagysága kilobyte-ban (64—256). Az alapértelmezés 112 kilobyte. Bővített (EMS, expanded) memória használatával ez az érték csökkenthető.

(folytatás a következő oldalon)

(folytatás az előző oldalról)

/P[+|-]

Az EGA/VGA képernyő elmentésének engedélyezése, illetve megtiltása.

/R[+|-]

Visszatérés a legutóbbi könyvtárhoz (igen|nem).

/Sút

A csereállomány (*swap file*) állomány helyének (elérési útjának) megadása [*meghajtó:*][*út*] alakban.

/T[+|-]

A rendszer főkönyvtárának betöltése (+: engedélyezve, -: megtiltva).

/Wméret

Az ablakok kezelésére szolgáló memóriaterület nagysága kilobyte-ban (28—64). Az alapértelmezés 32 kilobyte. A pillanatnyilag lefoglalt memória a megnyitott ablakok számával arányos.

/X[+|-]

A bővített (EMS, expanded) memória használatának engedélyezése, illetve megtiltása.

/Y[+|-]

A szimbóluminformációk elmentése (+: engedélyezve, -: megtiltva). (E paraméter csak a TPX.EXE mellett használható.)

állomány [*állomány*]...

A TURBO.EXE, illetve a TPX.EXE indulásakor betöltendő forrásállomány(ok) neve(i) [*meghajtó:*][*út*]*állománynév* alakban. Több állomány megadása esetén a neveket egymástól szóközzel kell elválasztani.

ÚJDONSÁGOK

A TP 7.0 SZERKESZTŐRENDSZERÉNEK ÚJDONSÁGAI

A 7.0-ás TP fejlesztői rendszere az IDE névvel jelölt fejlesztői rendszerek családjába tartozik. Ezen fejlesztői rendszereknek, s így a TP 7.0 fejlesztői rendszerének is legfőbb jellemzője a látványos, ám egyszerű kezelés.

Az IDE rendszerek a DOS alatt is hasonló vezérlést tesznek lehetővé, mint a Windows (menüsor, státuszsor, desktop, ablakok felépítése és kezelése stb.). Ennek ismertetésére (könyvünk stílusából adódóan) nem térhetünk ki.

Ez a felhasználóbarát fejlesztőrendszer már régóta a TP család része. A legnagyobb újítás az eddig egységesen kezelhető menürendszer bővítése. Így most már nemcsak a hagyományos felső (a továbbiakban globális) menüsor aktivizálható, hanem speciális helyzetekben az adott témához leginkább odaillő felső menü menüpontjainak, illetve ezen menüpontok bizonyos funkcióinak kivonatos menüje (a továbbiakban lokális menü) is igénybe vehető.

A globális és a lokális menük között tehát az a legfőbb különbség, hogy a globális menü az F10 funkcióbillentyű megnyomásával bármikor, bárhol hívható, míg a lokális menü csak egy adott helyzetben, ám akkor sokkal egyszerűbben. A lokális menü az Alt-F10 billentyűkombinációval, illetve a jobb egérgombbal hívható. A lokális menü hívásakor – az adott helyzettől függően – az öt lokális menü valamelyike aktivizálódik. A TP 7.0-ban *Edit*, *Browse*, *Watch*, *Messages* és *Help* lokális menü használható.

A forrásprogram megírását a TP szövegszerkesztője segíti. (A szövegszerkesztő funkcióit külön fejezetbe gyűjtöttük.) A szövegszerkesztő jellege alapvetően nem változott. A leglényegesebb változás a színezhető szövegszerkesztő. Így alaphelyzetben a háttérszín kék, a fenntartott szavak fehérek, a megjegyzések szürkék, az assembly betétek zöldek. A többi

szövegkomponens sárga. Ezek az alapbeállítások az *Options* menü *Environment* almenüjének *Colors* menüpontjával meg is változtathatók. A bal oldalon látható *Group* listában a *Syntax* bejegyzéssel teljesen átszínezzhetjük a szövegszerkesztőt.

További (esetleg) lényeges változás az, hogy a 6.0-ás TP 1023 karakteres sorhatára a 7.0-ás TP-ben programsorok esetén 127, s csak egyéb szövegek esetén 1023 karakter.

FEJLESZTŐI KÖRNYEZET ÉS MENÜRENDSZER

A TURBO PASCAL 7.0 MENÜRENDSZERE A TURBO.EXE MENÜJE

- 1. File** Alt-F
Az állományműveletek menüje.
- a) New** Alt-F N
Új állományablak megnyitása.
- b) Open** Alt-F O, F3
Létező állomány betöltése. A *Name* sorban a *Files* ablakban megjelenő lista szűrőjét állíthatjuk be (helyettesítő karakterek: * és ? segítségével). Visszahívhatók a korábban megadott állomány-, illetve állománycsoport-nevek is. Az *Open* kapcsoló hatására kiválasztott állomány új ablakba kerül, míg a *Replace* hatására felülíródik az aktuális nyitott ablak tartalma.
- c) Save** Alt-F S, F2
Az aktuális ablakban levő állomány lemezre mentése eredeti nevén. (Ha az aktuális ablakban név nélküli, korábban nem létező állomány van, a menüpont hatása megegyezik a *Save As...* menüponttal.)
- d) Save As...** Alt-F A
Az aktuális ablakban levő állomány lemezre mentése új néven. A *Save file as* sorban kell megadni az állomány (új) nevét. A sorban korábban megadott nevek visszahívhatók. A *Files* ablakban levő listában létező állomány neve is kiválasztható mentési névként, ekkor a Turbo Pascal (kérdés után) felülírja a kiválasztott állomány tartalmát.
- e) Save aLl** Alt-F L
Az összes nyitott ablak tartalmának lemezre mentése. A *New* menüponttal megnyitott (név nélküli) ablakok elmentésekor a *Save as...* menüpont aktivizálódik.

- f) Change dir...** Alt-F C
Könyvtár- és meghajtóváltás. A *Directory name* sorban az alkönyvtár elérési útját, nevét kell megadni. A korábban megadott alkönyvtárnevek visszahívhatók. A *Directory tree* ablakban ki is lehet választani a kívánt könyvtárt, illetve meghajtótben a munkát folytatni szeretnénk. A *ChDir* funkcióval beleléphetünk a kiválasztott (al)könyvtárba, míg a *Revert* funkció változtatás nélkül kilép a *Change dir...* ablakból.
- g) Print** Alt-F P
Az aktuális ablakban levő állomány vagy a kijelölt blokk kinyomtatása.
- h) pRinter setup...** Alt-F R
A nyomtató beállítása a *Print* menüpont használatához. A *Filter path* sorban a nyomtatóvezérlő program neve, a *Command line* sorban a nyomtatási paraméterek listája adható meg. A *Send highlighting escape codes* kapcsolóval a *dőlt*, a **vas-tag**, és az aláhúzott karakterek nyomtatása is engedélyezhető.
- i) DOS shell** Alt-F D
Kilépés DOS-környezetbe. A Turbo Pascal-hoz a DOS EXIT parancsával lehet visszatérni.
- j) eXit** Alt-F X, Alt-X
Kilép a *Turbo Pascalból*. Ha van még el nem mentett program, figyelmeztető üzenetet küld. Ekkor a megjelenő párbeszédablak *Save* funkciójával elvégezhető a mentés. (Korábban nem létező állomány mentésekor a *Save* hatása azonos a *Save as...* menüpontéval.)

2. Edit Alt-E

A forrásállomány(ok) szerkesztése.

a) Undo Alt-E U, Alt-Bksp

Az utoljára végzett szerkesztési művelet (beszúrás, törlés, mozgatás, stb.) hatásának megszüntetése (lásd az *Options • Environment • Editor* menüpontot).

b) Redo Alt-E R

A legutoljára kiadott *Undo* hatásának megszüntetése.

c) Cut Alt-E T, Shift-Del

A kijelölt blokkot a vágóasztalra másolja, és törli a szerkesztett szövegből. A törölt szöveg visszaállítása az *Undo* funkcióval lehetséges.

d) Copy Alt- E C, Ctrl-Ins

Az aktuális ablakból a kijelölt blokkot a vágóasztalra másolja. Ha a vágóasztalon már van adat, akkor a blokkot a vágóasztal tartalmának végére írja.

e) Paste Alt-E P, Shift-Ins

A vágóasztal tartalmából a kijelölt részt a szerkesztett szövegbe másolja az aktuális kurzorpozíciótól kezdve.

f) Clear Alt-E L, Ctrl-Del

A kijelölt blokk törlése.

g) Show clipboard Alt-E S

A vágóasztal tartalmának megtekintése és szerkesztése külön ablakban.

3. Search

Alt-S

Keresések, cserék és ugrások.

a) Find...

Alt-S F

A *Text to find* sorban megadott szöveg megkeresése az aktuális állományban. Az *Options* mezőben beállítható keresési módok:

Case sensitive

A keresés során különbséget tesz kis- és nagybetűk között.

Whole words only

A megadott szöveget teljes szóként keresi.

Regular expression

A helyettesítő karakterek használatának engedélyezése, illetve megtiltása.

A *Direction* mezőben a keresés iránya választható ki. A *Forward* az állomány eleje felé, a *Backward* az állomány vége felé indítja meg a keresést. A *Scope* mezőben a keresés intervalluma adható meg. A *Global* kapcsoló hatására a Turbo Pascal az egész állományban, a *Selected text* kapcsoló hatására pedig csak a kijelölt részben keres. Az *Origin* mezőben a keresés kezdeti pozíciója jelölhető ki. A *From cursor* kapcsolóval a kurzor helyénél, a *Entire scope* kapcsolóval az ablak első soránál kezdi a keresést.

b) Replace...

Alt-S R

Mint *Find...*, de a keresett szöveg kicserélése is lehetséges. A cserélendő szöveget a *Text to find*, a csereszöveget pedig a *New text* mezőben kell megadni. Az *Options* mezőben levő *Prompt on replace* kapcsoló hatására minden csere előtt megerősítést kér.

- c) Search again** Alt-S S
Az előző keresés vagy csere megismétlése.
- d) Go to line number...** Alt-S G
Az állomány adott számú sorára ugrik. A sor számát az *Enter new line number* mezőben kell megadni. A mezőben korábban megadott számok visszahívhatók.
- e) Show last compiler error** Alt-S C
Az utolsó fordítási hiba helyének és típusának megjelenítése.
- f) Find error...** Alt-S E
Hiba keresése a memóriába fordított programban a megadott címtől kezdődően. A keresés kezdőcímét az *Error adress* sorban kell megadni.
- g) Find procedure...** Alt-S P
Eljárás vagy függvény keresése. Az eljárás vagy függvény nevét a *Procedure name* sorban kell megadni. A korábban megadott nevek visszahívhatók.

4. Run Alt-R

A bevitt Pascal nyelvű program futtatása.

a) Run Alt-R R, Ctrl-F9

A Pascal program fordítása, szerkesztése (link) és futtatása.

b) Step over Alt-R S, F8

A program lépésenkénti végrehajtása az aktuális programszinten. Ez azt jelenti, hogy az adott programszintről hívott eljárásokat, függvényeket a Turbo Pascal nem hajtja végre lépésenként. Az aktuális programszintet az adja meg, hogy a kurzor melyik eljárásban áll.

c) Trace into Alt-R T, F7

A program lépésenkénti végrehajtása. Minden utasításon végiglép, a programszinttől függetlenül.

d) Go to cursor Alt-R G, F4

A program futása a kurzor helyéig folyamatos, a kurzornál megáll. Innen akár a *Step over*, akár a *Trace into*, akár a *Run* menüponttal folytatható a program futtatását.

e) Program reset Alt-R P, Ctrl-F2

Az aktuális program futásának megszüntetése a *Go to cursor*, a *Step over*, a *Trace into* vagy a *Debug/Breakpoints* menüpontok korábbi használata esetén.

f) Parameters... Alt-R A

Az aktuális program indítási paramétereinek beállítása. A paramétereket a *Parameter* mezőben kell megadni a program indítására szolgáló DOS-parancssor formájának megfelelően. A korábban megadott paraméterlisták visszahívhatók.

- 5. Compile** Alt-C
Pascal program fordítása és szerkesztése (link).
- a) Compile** Alt-C C, Alt-F9
Az aktuális pascal program fordítása memóriába vagy lemezre (lásd *Destination: Memory/Disk*).
- b) Make** Alt-C M, F9
Az elsődleges állomány lefordítása (lásd *Primary file...*). Ha nincs ilyen, akkor csak az aktuális ablakban levő programot fordítja le.
- c) Build** Alt-C B
Az elsődleges állomány lefordítása (lásd *Primary file...*). Ha nincs ilyen, akkor az aktuális ablakban levő programot fordítja le az összes onnan hívott forrásállománnyal együtt.
- d) Destination Memory/Disk** Alt-C D
Annak beállítása, hogy a Turbo Pascal a programot a memóriába vagy lemezre (.EXE állományba) fordítsa.
- e) Primary file...** Alt-C P
Az elsődleges állomány beállítása. A megjelenő párbeszédablak felépítése megegyezik a *File* menü *Open* menüpontjával, kivéve, hogy az *Open* menüpont *Replace* kapcsolója helyén egy *Clear* kapcsoló van. A *Clear* funkcióval az elsődleges állomány kiválasztása törölhető.
- f) Clear primary file** Alt-C L
Az elsődleges állomány beállításának törlése.
- g) Information...** Alt-C I
A lefordított program legfontosabb adatainak (sorok száma, memóriakiosztás, kódméret stb.) táblázatba rendezett megjelenítése.

6. Debug Alt-D

A Pascal program futásának figyelése.

a) Breakpoints... Alt-D B

Műveletek a töréspontokkal. (Töréspontokat az *Add breakpoint* menüponttal vagy a Ctrl-F8 billentyűkombinációval hozhatunk létre.) E menüpont szolgáltatásai:

Edit

A kijelölt töréspont módosítása (lásd *Add breakpoint*).

Delete

A kijelölt töréspont törlése.

View

A kijelölt töréspont programbeli helyének megjelenítése.

Clear all

Az összes töréspont törlése.

b) Call stack Alt-D C, Ctrl-F3

A verem tartalmának megjelenítése. A verem ablakában mindig az éppen futó programszint, majd a magasabb programszintek láthatók – utoljára a programmal.

c) Register Alt-D R

A regiszterek tartalmának megjelenítése.

d) Watch Alt-D W

Változók, kifejezések értékének folyamatos (programfutas közbeni) megjelenítése. A Watch funkcióhoz rendelt lokális menü az Alt-F10 billentyűkombinációval hívható elő.

Add... Alt-D W A, Alt-D W Ins

Új változó felvétele a *Watch* funkcióval vizsgált változó- és kifejezéslistára. A vizsgálandó változó nevét a *Watch expression* sorban kell meg-
(folytatás a következő oldalon)

(folytatás az előző oldalról)

adni. A korábban megadott változónevek és ki-
fejezések visszahívhatók.

Modify... Alt-D W M, Alt-D W Enter
A kiválasztott változó nevének megváltoztatá-
sa.

Remove... Alt-D W R, Alt-D W Del
A kiválasztott változó törlése a *Watch* listájáról.

Clear all Alt-D W C
Az összes változó törlése a *Watch* listájáról.

Enable Alt-D W E
Letiltott (lásd *Disable*) változó újbóli engedélye-
zése.

Disable Alt-D W D
Változó ideiglenes kitiltása az értékvizsgálatból.

e) Output Alt-D O
A DOS-felület ablakban való megjelenítése.

f) User screen Alt-D U, Alt-F5
Átkapcsolás a DOS-felületre. A visszatérés egy bil-
lentyű leütésével lehetséges.

g) Evaluate/modify... Alt-D E, Ctrl-F4
Változók értékének megjelenítése. Az *Expression*
sorban megadott változó értéke a *Result* sorban
jelenik meg. A *New value* sorral, illetve a *Modify*
funkcióval a változó értéke megváltoztatható. Az
Expression sorban néhány alapvető függvény is
használható.

h) Add watch... Alt-D A, Ctrl-F7
Változó kiválasztása és megjelenítése a *Watch*
menüponttal megnyitható ablakban.

i) Add breakpoint...

Alt-D P, Ctrl-F8

Új töréspont megadása. A *Condition* sorban a törésponthez feltétel rendelhető, a *Pass count* sorban meghatározható, hogy a program hányszor futhat át a törésponton. A *File name* sorban a töréspontot tartalmazó állomány neve szerepel. A *Line number* sorban a töréspont sorának száma látható.

7. Tools Alt-T

Segédeszközök.

a) Messages Alt-T M

A segédprogramok üzeneteinek megjelenítése külön ablakban.

b) Go to next Alt-T N, Alt-F8A *Messages* ablak következő sorára lép.**c) Go to previous** Alt-T P, Alt-F7A *Messages* ablak előző sorára lép.**d) Grep** Alt-T G, Shift-F2Szövegkeresés több állományban. A *GREP* program indul el (a paraméterezést lásd az *Options/Tools...* menüpontnál).**e) Turbo Assembler** Alt-T A, Shift-F3A *Turbo Assembler* elindítása (a paraméterezést lásd az *Options/Tools...* menüpontnál).**f) Turbo Debugger** Alt-T D, Shift-F4A *Turbo Debugger* elindítása (a paraméterezést lásd az *Options/Tools...* menüpontnál).**g) Turbo Profiler** Alt-T R, Shift-F5A *Turbo Profiler* elindítása (a paraméterezést lásd az *Options/Tools...* menüpontnál).

8. Options Alt-O

A *Turbo Pascal 7.0* működési paramétereinek beállítása.

a) Compiler... Alt-O C

A teljes programra vonatkozó paraméterek.

Code generation mező:***[] Force far calls***

A hívási mód beállítása. Ha be van kapcsolva, a Turbo Pascal fordításkor csak FAR típusú hívásokat használ; egyébként NEAR típusúakat is.

[] Word align data

Bekapcsolása esetén minden változó és konstans szóhatárra (kettővel osztható, azaz páros címre), egyébként byte-határra kerül.

[] Overlays allowed

Bekapcsolása esetén lehetőség van az eljárások *overlay*-rendszerbe szervezésére.

[] 286 instructions

Bekapcsolása esetén csak 80286-os vagy fejlettebb processzoron működő kódot generál (amely 8086-os és 8088-as gépeken, XT-ken nem működik).

Runtime errors mező:***[] Range checking***

A tömbindexek ellenőrzésének bekapcsolása és kikapcsolása.

[] Stack checking

Bekapcsolása esetén a Turbo Pascal minden szubrutinhíváskor ellenőrzi, hogy van-e elegendő memória a program futtatásához.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

[] I/O checking

Ha be van kapcsolva, akkor I/O-hiba esetén a program megáll, Kikapcsolt állapotában az I/O-hibák az *IoResult* függvénnyel a programban kezelhetők.

[] Overflow checking

Ha be van kapcsolva, a program futása egész típusú aritmetikai műveletek végrehajtásakor túlsordulás esetén 215-ös hibával megáll.

Syntax options mező:

[] Strict var-strings

Ha be van kapcsolva, a paraméterként átadott karakterláncok hosszának meg kell egyeznie a várt hosszal.

[] Complete boolean eval

Bekapcsolása esetén a lefordított program teljesen kiértékeli a logikai kifejezéseket. Bekapcsolása nélkül a kiértékelés véget ér, amint a logikai kifejezés értéke egyértelműen igaz vagy hamis.

[] Extended syntax

Használata esetén az eljárások függvényként is hívhatók.

[] Typed @ operator

Bekapcsolt állapotában a Turbo Pascal ellenőrzi a @ operátor (címpoperátor) által generált mutatók típusát. Különben a @ operátor eredménye típus nélküli mutató lesz.

[] Open parameters

Bekapcsolása esetén az új String típus nyitott paraméterként is szerepelhet (lásd az \$P direktívát!).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

Debugging mező:

Debug information

Bekapcsolása esetén a Turbo Pascal a fordításkor olyan táblázatot készít, amelynek segítségével a program futása, illetve nyomkövetése a menüből, vagy külső nyomkövető programmal (lásd *Turbo Debug*) könnyen elvégezhető.

Local symbols

Bekapcsolása esetén a Turbo Pascal táblázatot készít a program változóiról.

Numeric processing mező:

8087/80287

Bekapcsolt állapotában engedélyezi a számítógépben levő matematikai társprocesszor használatát. Társprocesszor nélkül a kapcsoló hatástalan.

Emulation

Ha nincs a gépben matematikai társprocesszor, a *Turbo Pascal 7.0* emulálja a 80x87 működését.

A *Conditional defines* sorban feltételes fordítási direktíva is bekapcsolható.

b) Memory sizes... Alt-O M

A verem méretének, valamint a változók által használható memória alsó (*Low heap limit*) és felső (*High heap limit*) határának beállítása.

c) Linker... Alt-O L

A fordításkor létrehozandó térképállomány típusának beállítása. Az *Off* állásban nem készül térképállomány. A *Segment*, a *Public* és a *Detailed* (a fel-

(folytatás a következő oldalon)

(folytatás az előző oldalról)

sorolás sorrendjében) egyre több információt ad a program fordításáról. A *Memory* és a *Disk* kapcsolókkal a program szerkesztésének (*link*) helye adható meg (memória, illetve lemez).

d) Debugger... Alt-O B
A nyomkövetéshez használt táblázat kezelésének beállítása.

e) Directories... Alt-O D
A *Turbo Pascal 7.0* működéséhez szükséges (al)könyvtárak megadása. Az *EXE & TPU directory* sorban a lemezrefordítás (illetve az ehhez tartozó más állományok) célkönyvtárát, az *Include directories* sorban a befordítandó, illetve kezelendő külső állományok elérési útját, a *Unit directories* sorban a *Turbo Pascal 7.0* eljárás egységeit (*unitjait*) tartalmazó könyvtár helyét kell megadni. Az *Object directories* sorban annak a könyvtárnak kell szerepelnie, ahol a szerkesztéshez használt *.OBJ* állományok vannak.

f) Tools... Alt-O T
A külső segédprogramok paraméterezése. Az *Edit* kapcsolóval létező menübejegyzések javíthatók, a *New* funkció új bejegyzés beszúrására, míg a *Delete* létező bejegyzés törlésére szolgál.

g) Environment Alt-O E
A fejlesztői környezet tulajdonságainak beállítása. Almenüi: a *Preferences...* a *Turbo Pascal 7.0* szövegszerkesztőjével; az *Editor...* a szövegszerkesztő, a *Mouse...* az egér működésével foglalkozik; a *Start up...* menüpontban a *Turbo Pascal 7.0* indítási állapota határozható meg; a *Colors...* szolgáltatással a képernyő színei módosíthatók.

h) Open... Alt-O O

A *Save...*, illetve a *Save as...* menüponttal állományba mentett környezeti paraméterek betöltése. Kezelése: mint *File•Open* (lásd ott).

i) Save Alt-O S

A környezeti paraméterek elmentése az alapértelmezés szerinti konfigurációs állományba. Az alapértelmezés szerinti állomány a *Save as...* menüpontban állítható be. Az állomány neve a menüpontban látható.

j) Save as... Alt-O A

A környezeti paraméterek elmentése megadott nevű állományba, az alapértelmezés szerinti konfigurációs állomány nevének beállítása. (Az alapértelmezett állománynév kezdetben TURBO.TP.) Kezelése: mint *File•Save as...* (lásd ott).

- 9. Window** Alt-W
Az ablakok kezelése.
- a) Tile** Alt-W T
A képernyőn minden nyitott ablak megjelenik, a képernyő területét téglalapokra osztva.
- b) Cascade** Alt-W A
A képernyőn – egymást szabályosan átfedve – minden nyitott ablak megjelenik.
- c) Close all** Alt-W O
Az összes nyitott ablak lezárása.
- d) Refresh display** Alt-W R
A képernyő tartalmának felfrissítése.
- e) Size/Move** Alt-W S, Ctrl-F5
Az aktív ablak mozgatása, illetve méretének megváltoztatása. Az ablak a nyílbillentyűkkel mozgatható; a nyílbillentyűket Shifttel együtt használva az ablak mérete változtatható meg.
- f) Zoom** Alt-W Z, F5
Az aktív ablak kinagyítása a teljes képernyőre.
- g) Next** Alt-W N, F6
A megnyitás sorrendje szerinti következő ablakra lép.
- h) Previous** Alt-W P, Shift-F6
A megnyitás sorrendje szerinti előző ablakra lép.
- i) Close** Alt-W C, Alt-F3
Az aktív ablak lezárása.
- j) List...** Alt-W L, Alt-0
Az ablakok felsorolása az aktív ablak kiemelésével. A *Delete* funkció a kijelölt ablak lezárására szolgál.

- 10. Help** Alt-H, F1
A *Turbo Pascal 7.0* segítőrendszere.
- a) Contents** Alt-H C
A segítőrendszer téma szerinti tartalomjegyzéke.
- b) Index** Alt-H I, Shift-F1
A *Turbo Pascal 7.0* nyelvi elemeinek tárgymutatója.
- c) Topic search** Alt-H T, Ctrl-F1
A kurzor alatt álló szóval kapcsolatos információk megkeresése és megjelenítése. Ha nincs ilyen szó, akkor az alakilag legközelebb álló bejegyzést mutatja meg a segítőrendszer tárgymutatójában.
- d) Previous topic** Alt-H P, Alt-F1
A segítőrendszeren belüli előző tartózkodási helyre (ablakra) ugrik.
- e) Using help** Alt-H H
Leírás a segítőrendszer használatáról.
- f) Files...** Alt-H F
A segítőrendszer adatbázis-állományának kiválasztása.
- g) Compiler directives** Alt-H D
A fordítási direktívák listája, illetve ismertetése *ábécé* sorrendben.
- h) Procedures and functions** Alt-H O
A *Turbo Pascal 7.0* beépített eljárásainak és függvényeinek lexikonszerű ismertetése.
- i) Reserved words** Alt-H R
A *Turbo Pascal 7.0* fenntartott szavainak ismertetése.
- j) Standard units** Alt-H U
A *Turbo Pascal 7.0* alapvető eljárássegységeinek (*unitjainak*) ismertetése.

- k) Turbo Pascal Language** Alt-H L
A *Turbo Pascal* programnyelv ismertetése.
- l) Error messages** Alt-H E
Információk a hibaüzenetekről.
- m) About...** Alt-H A
Általános információk a *Turbo Pascal 7.0* fejlesztőrendszeréről.

A TPX.EXE MENÜRENDSZERE

A TPX.EXE menürendszere a TURBO.EXE menüjéhez képest néhány többlet-menüpontot tartalmaz. Ezek ismertetése következik.

- 3. Search** Alt-S
 Keresések, cserék és ugrások.
- h) Previous browser** Alt-S F
 A legutóbbi *böngésző-(browse-)*művelet ismételt végrehajtása (*ObjectBrowser*).
- i) Objects** Alt-S F
 A (lefordított) forrásállományban található objektumok listájának megjelenítése. A listából kiválasztott objektumokról *böngésző-(browse-)*ablakok kérhetők.
- j) Units** Alt-S F
 A (lefordított) forrásállományban felhasznált eljárásrészegységek (*unitok*) listájának megjelenítése. A listából kiválasztott objektumokról *böngésző-(browse-)*ablakok kérhetők.
- k) Globals** Alt-S F
 A (lefordított) forrásállományban található globális szimbólumok felsorolása. A listából kiválasztott objektumokról további *böngésző-(browse-)*ablakok is kérhetők.
- l) Symbol...** Alt-S F
*Böngésző-(browse-)*információk a kurzor alatt álló szóról mint szimbólumról.
- 5. Compile** Alt-C
 Pascal program fordítása és szerkesztése (link).
- d) Destination Memory/Disk** Alt-C D
 Ez a menüpont hiányzik a TPX-ből, mert a TPX mindig lemezre fordít.

- k) Turbo Pascal Language** Alt-H L
A *Turbo Pascal* programnyelv ismertetése.
- l) Error messages** Alt-H E
Információk a hibaüzenetekről.
- m) About...** Alt-H A
Általános információk a *Turbo Pascal 7.0* fejlesztőrendszeréről.

A TPX.EXE MENÜRENDSZERE

A TPX.EXE menürendszere a TURBO.EXE menüjéhez képest néhány többlet-menüpontot tartalmaz. Ezek ismertetése következik.

- 3. Search** Alt-S
 Keresések, cserék és ugrások.
- h) Previous browser** Alt-S F
 A legutóbbi *böngésző-(browse-)*művelet ismételt végrehajtása (*ObjectBrowser*).
- i) Objects** Alt-S F
 A (lefordított) forrásállományban található objektumok listájának megjelenítése. A listából kiválasztott objektumokról *böngésző-(browse-)*ablakok kérhetők.
- j) Units** Alt-S F
 A (lefordított) forrásállományban felhasznált eljárásrészegységek (*unitok*) listájának megjelenítése. A listából kiválasztott objektumokról *böngésző-(browse-)*ablakok kérhetők.
- k) Globals** Alt-S F
 A (lefordított) forrásállományban található globális szimbólumok felsorolása. A listából kiválasztott objektumokról további *böngésző-(browse-)*ablakok is kérhetők.
- l) Symbol...** Alt-S F
*Böngésző-(browse-)*információk a kurzor alatt álló szóról mint szimbólumról.
- 5. Compile** Alt-C
 Pascal program fordítása és szerkesztése (link).
- d) Destination Memory/Disk** Alt-C D
 Ez a menüpont hiányzik a TPX-ből, mert a TPX mindig lemezre fordít.

8. Options Alt-O

A *Turbo Pascal 7.0* működési paramétereinek beállítása.

f) Browser... Alt-O W

Új menüpont a *Directories* és a *Tools* menüpont között: a *böngésző-(browse-)*funkciók működésének szabályozása.

Symbols mező

Az *ObjectBrowser* szolgáltatással vizsgálendő objektumok kiválasztása.

 Labels

Címkék.

 Constants

Konstansok.

 Types

Típusok.

 Variables

Változók.

 Procedures

Eljárások, függvények, konstuktorok, destruktorok.

 Inherited

Örökölt szimbólumok.

Sub-browsing mező:

A *böngésző-(browse-)*információk második szintű ablakjainak kezelése.

 New browser

A további információknak új ablakot nyit.

 Replace current

A további információkat az előző ablak tartalmának felülírásával jeleníti meg.

(folytatás a következő oldalon)

(folytatás az előző oldalról)

Prefered pane mező:

Az alapértelmezés szerinti listatípus beállítása.

Scope

Területlista.

Reference

Referencialista.

Display mező:

Az információk kiírásának mélysége.

Qualified symbols

Ha be van kapcsolva, a TPX a szimbólumok származását is megjeleníti.

Sort always

Bekapcsolt állapotában a lista alfabetikus, egyébként a deklarációk szerinti sorrendű.

LOKÁLIS MENÜK

LOKÁLIS EDIT MENÜ

Szövegszerkesztés közben az Alt-F10 billentyűkombinációval aktivizálható.

a) Cut Alt-F10 T, Shift-Del

A kijelölt blokkot kitörli a szerkesztett állományból, és a vágóasztalra másolja. A törölt blokk az *Undo* funkcióval állítható vissza.

b) Copy Alt-F10 C, Ctrl-Ins

Az aktuális ablakból a kijelölt blokkot a vágóasztalra másolja. Ha a vágóasztalon már van adat, akkor a most bemásolt adatokat hozzáfűzi a már ott lévőkhöz.

c) Paste Alt-F10 P, Shift-Ins

A vágóasztal tartalmából kijelölt részt a szerkesztett állomány ablakába másolja, az aktuális kurzorpozíciótól kezdve.

d) Clear Alt-F10 L, Ctrl-Del

A kijelölt blokk törlése.

e) Open file at cursor Alt-F10 F, Ctrl-Enter

Létező állomány gyors megnyitása. Az állomány neve a kurzor alatt álló szó.

f) Browse symbol at cursor Alt-F10 B

*Böngésző-(browse-)*információk a kurzor alatt álló szóról mint szimbólumról. Csak a TPX-ben használható.

g) Topic search Alt-F10 S, Ctrl-F1

A kurzor alatt álló szóval kapcsolatos segítő információkat megkeresése és megjelenítése. Ha nincs ilyen szó a segítőrendszerben, akkor az alakilag legközelebb levő szóra ugrik a segítőrendszer tárgymutatójában.

- h) Toggle breakpoint** Alt-F10 K, Ctrl-F8
Új töréspont kijelölése, illetve a kurzor alatt álló régi megszüntetése.
- i) Go to cursor** Alt-F10 G, F4
A program folyamatos futtatása a kurzor helyéig. A kurzor helyétől a futtatás a *Step over*, a *Trace into* vagy a *Run* menüponttal folytatható.
- j) Evaluate/modify...** Alt-F10 E, Ctrl-F4
Változók értékének megtekintése és módosítása. Az *Expression* sorban kiválasztott változó értéke a *Result* sorban jelenik meg. A *New value* sorral, illetve a *Modify* funkcióval a változónak új érték adható. Az *Expression* sorban a SYSTEM eljárássegység (*unit*) néhány alapvető függvénye is használható.
- k) Add watch...** Alt-F10 A, Ctrl-F7
Változó kiválasztása és megjelenítése a *Watch* menüponttal megnyitható ablakban.
- l) Options...** Alt-F10 O
Az *Options* menü *Environment* almenüjének *Editor...* menüpontja.

LOKÁLIS BROWSE MENÜ

Böngésző-(browse-)ablak használata közben az Alt-F10 billentyűkombinációval hívható elő.

a) Browse Alt-F10 B
Újabb *böngésző*-ablak az aktív *böngésző*-lista egy eleméről.

b) Previous Alt-F10 P, Ctrl-P
Visszatér az előző *böngésző-(browse-)ablak*hoz.

c) Goto source Alt-F10 G, Ctrl-Enter
Az aktív *böngésző-(browse-)ablak*ban a kurzor alatt álló listaelem deklarációját tartalmazó programsorra ugrik. Referencialistánál alkalmazva a jelzett sorszámú sorra ugrik. Ha a jelzett sort tartalmazó állomány nincs megnyitva, akkor ez a funkció azt is megteszi.

d) Track source Alt-F10 T, Space
Megjelöli az aktív *browse*-ablakban a kurzor alatt álló listaelem deklarációját tartalmazó programsort. Referencialistánál alkalmazva a jelzett sorszámú sort jelöli meg. Ha a jelzett sort tartalmazó állomány nincs megnyitva, akkor ez a funkció megnyitja. Ha az *Options* menü *Environment* almenüjének *Preferences* menüpontjában az *Auto track source* kapcsoló aktív, akkor a kijelölés e menüpont használata nélkül automatikusan megtörténik.

e) Options... Alt-F10 O
Az *Options* menü *Browser* menüpontjának szűkített változata.

***Symbols* mező:**

Az *ObjectBrowser* szolgáltatással vizsgálandó szimbólumok kiválasztása.

[] Labels
Címkék.

(folytatás az előző oldalról)

(folytatás az előző oldalról)

Constans

Konstansok.

Types

Típusok.

Variables

Változók.

Procedures

Eljárások, függvények, konstuktorok, destruktorok.

Inherited

Örökölt szimbólumok.

Display mező:

Az információk kiírásának mélysége.

Qualified symbols

Bekapcsolt állapotában a TPX a szimbólumok származását is feltünteti.

Sort always

Bekapcsolt állapotában a lista alfabetikus, egyébként a deklarációk szerinti sorrendű.

LOKÁLIS WATCH MENÜ

A *Watch*-ablak használata közben az Alt-F10 billentyűkombinációval hívható elő.

a) Add... Alt-F10 A, Ins
Változó kiválasztása és megjelenítése a *Watch* menüponttal megnyitható ablakban.

b) Modify... Alt-F10 M, Enter
A kurzor alatt álló kifejezés szerkesztése a *Watch*-ablakban.

c) Remove Alt-F10 R, Del, Ctrl-Y
A kurzor alatt álló kifejezés törlése a *Watch*-ablakból.

d) Clear all Alt-F10 C
A *Watch*-ablakban levő összes kifejezés törlése.

e) Enable Alt-F10 E
A kurzor alatt álló letiltott kifejezés felszabadítása. A kifejezés értéke ezentúl ismét megjelenik a kiértékelés során.

f) Disable Alt-F10 D
A kurzor alatt álló kifejezés letiltása. A kifejezés értéke ezután nem jelenik meg a kiértékelés során.

LOKÁLIS MESSAGES MENÜ

A *Messages* ablak használata közben az Alt-F10 billentyűkombinációval aktivizálható.

a) Clear Alt-F10 C
Törli a *Messages* ablakban levő összes üzenetet.

b) Goto source Alt-F10 G, Enter
Ha a *Messages* ablak aktuális sorában állománynév is szerepel, akkor ez a szolgáltatás betölti az állományt, majd abban a jelzett sorra ugrik.

c) Track source Alt-F10 T, Szóköz
Ha a *Messages* ablak aktuális sorában állománynév is szerepel, akkor ez a szolgáltatás betölti az állományt, és a kiemeli benne a jelzett sort. A lokális *Browse* menünél említett *Auto track source* kapcsoló aktivitása ezt a funkciót is automatizálja.

LOKÁLIS HELP MENÜ

A segítőrendszer használata közben az Alt-F10 billentyűkombinációval hívható elő.

a) Contents Alt-F10 O

A segítőrendszer téma szerinti tartalomjegyzéke.

b) Index Alt-F10 I, Shift-F1

A *Turbo Pascal 7.0* nyelvi elemeinek tárgymutatója.

c) Topic search Alt-F10 T, Ctrl-F1

A kurzor alatt álló szóval kapcsolatos információk megkeresése és megjelenítése. Ha nincs ilyen szó a segítőrendszerben, akkor az alakilag legközelebb álló bejegyzésre ugrik a tárgymutatóban.

d) Previous topic Alt-F10 P, Alt-F1

A segítőrendszeren belüli előző tartózkodási helyre (ablakra) ugrik.

e) Copy Alt-F10 C, Ctrl-Ins

A segítő ablak kijelölt részét a vágóasztalra másolja, ahonnan az a *Paste* funkcióval az aktív ablakba másolható.

f) Copy example Alt-F10 E

A beépített függvények és eljárások példaprogramjait a vágóasztalra másolja, ahonnan azok a *Paste* menüponttal az aktív ablakba másolhatók.

A TURBO PASCAL 7.0 SZÖVEGSZERKESZTŐJE

A TURBO PASCAL 7.0 SZÖVEGSZERKESZTŐJE

Ctrl-A, Ctrl-balra nyíl
Ugrás az előző szóra.

Ctrl-C, PgDn
Egy képernyőnyit lapoz lefelé.

Ctrl-D
Ugrás a közvetkező karakterre.

Ctrl-Del
A kijelölt blokk törlése a szövegből.

Ctrl-E
Ugrás az előző sorra.

Ctrl-F, Ctrl-jobbra nyíl
Ugrás a következő szóra.

Ctrl-G, Del
A kurzor alatt álló karakter törlése.

Ctrl-H, Backspace
A kurzortól balra levő karakter törlése.

Ctrl-I, Tab
Tabulátor beszúrása.

Ctrl-Ins
A kijelölt blokkot a vágóasztalra másolja.

Ctrl-K *n*
Jelző elhelyezése, ahol $n=0-9$.

Ctrl-K B
Blokk-kezdet kijelölése.

Ctrl-K C

A kijelölt blokk átmásolása az aktuális kurzorpozíciótól kezdve.

Ctrl-K H

A kijelölt blokk láthatóvá tétele, illetve elrejtése.

Ctrl-K I

A kijelölt blokk egy pozícióval jobbra mozdul.

Ctrl-K K

Blokkvég kijelölése.

Ctrl-K P

A kijelölt blokk kinyomtatása.

Ctrl-K R

Blokk bemásolása lemezről az aktuális kurzorpozíciótól kezdve.

Ctrl-K S, F2

A szerkesztett állomány lemezre mentése. A *Turbo Pascal 7.0* (.BAK kiterjesztéssel) megőrzi az állomány előző változatát.

Ctrl-K T

Egyetlen szó blokk-ként való kijelölése.

Ctrl-K U

A kijelölt blokk egy pozícióval balra mozdul.

Ctrl-K Y

A kijelölt blokk törlése a szövegből.

Ctrl-K V

A kijelölt blokk törlése eredeti helyéről, és beszúrása az aktuális kurzorpozíciótól kezdve.

Ctrl-K W

A kijelölt blokk lemezre írása.

Ctrl-L

Az előző keresés vagy csere megismétlése.

Ctrl-N

Sorbeszúrás.

Ctrl-O F

Az optimális feltöltés ki-, illetve bekapcsolása.

Ctrl-O I

Az automatikus bekezdésállítás (*autoindent*) be-, illetve kikapcsolása.

Ctrl-O O

Az *Options* menü *Compiler* és *Memory sizes* ablakában levő kapcsolók beírása az állomány elejére (fordítási direktívák formájában).

Ctrl-O R

A Tab billentyű működésének beállítása.

Ctrl-O T

A Tab billentyű használatának beállítása.

Ctrl-O U

A visszatörlési eltolás (*unindent*) be-, illetve kikapcsolása.

Ctrl-Q n

Ugrás az *n*-edik jelzőre (lásd Ctrl-K *n*).

Ctrl-Q [

A kurzor alatt álló (, [, {, <, ' vagy " karakter párjának keresése az állomány vége felé.

Ctrl-Q]

A kurzor alatt álló),], }, >, ' vagy " karakter párjának keresése az állomány eleje felé.

Ctrl-Q A

Keresés és csere.

Ctrl-Q B

Ugrás a kijelölt blokk elejére.

Ctrl-Q C, Ctrl-PgDn

Ugrás az állomány végére.

Ctrl-Q D, End

Ugrás az aktuális sor végére.

Ctrl-Q E, Ctrl-Home

Ugrás az ablak elejére.

Ctrl-Q F

Keresés.

Ctrl-Q K

Ugrás a kijelölt blokk végére.

Ctrl-Q P

Ugrás az előző kurzorpozícióra.

Ctrl-Q R, Ctrl-PgUp

Ugrás az állomány elejére.

Ctrl-Q X, Ctrl-End

Ugrás az ablak végére.

Ctrl-Q Y

Törlés a sor végéig.

Ctrl-Q W

Ugrás az előző hiba helyére.

Ctrl-R, PgUp

Egy képernyőnyit lapoz felfelé.

Ctrl-S

Ugrás az előző karakterre.

Ctrl-T

A kurzortól jobbra levő szó törlése.

Ctrl-X

Ugrás a következő sorra.

Ctrl-Y

Az aktuális sor törlése.

Ctrl-W

Folyamatos sorgörgetés felfelé.

Ctrl-Z

Folyamatos sorgörgetés lefelé.

Shift-Ins

A vágóasztal tartalmának beszúrása az aktuális kurzorpozíciótól kezdve.

Shift-Del

A kijelölt blokk törlése a szövegből. A blokk a vágóasztalra kerül.

ADATTÍPUSOK

ADATTÍPUSOK

1. Egyszerű típusok

BOOLEAN

Logikai típus, két értéket vehet fel: TRUE (igaz), és FALSE (hamis). Tárolása: egy byte-on.

BYTE

Előjel nélküli *egész típus*, értékkészlete: 0-tól 255-ig, tárolása: 8 biten.

CHAR

Egyetlen karakter tárolásához szükséges típus. Tárolása: egy byte-on.

COMP

Valós típus, értékkészlete: $-9.2E18$ -tól $9.2E18$ -ig. Tárolása: 8 byte-on, használatához matematikai társprocesszor vagy annak emulálása szükséges.

DOUBLE

Valós típus, értékkészlete: $5.0E-324$ -tól $1.7E308$ -ig. Tárolása: 8 byte-on, használatához matematikai társprocesszor vagy annak emulálása szükséges.

EXTENDED

Valós típus, értékkészlete: $3.4E-4932$ -tól $1.1E4932$ -ig. Tárolása: 10 byte-on, használatához matematikai társprocesszor vagy annak emulálása szükséges.

Felsorolt típus: (azonosító[,azonosító...])

Az összetett adatok tárolására alkalmas adattípus, amely lehet karakter, egész, illetve logikai típus. A tároláshoz szükséges byte-ok száma az azonosítók számának függvénye (256-ig egy byte, fölötte kettő).

INTEGER

Előjeles *egész típus*, értékkészlete: -32768 -tól 32767 -ig, tárolása: 16 biten.

LONGINT

Előjeles *egész típus*, értékkészlete: -2147483648-tól 2147483647-ig, tárolása: 32 biten.

POINTER

Mutatótípus, értéke: négy byte-os cím. Tárolása: 4 byte-on.

REAL

Valós típus, értékkészlete: 2.9E39-től 1.7E38-ig. Tárolása: 6 byte-on.

SHORTINT

Előjeles *egész típus*, értékkészlete: -128-tól 127-ig, tárolása: 8 biten.

SINGLE

Valós típus, értékkészlete: -1.5E-45-től 3.4E38-ig. Tárolása: 4 byte-on, használatához matematikai társprocesszor vagy annak emulálása szükséges.

STRING [hossz]

Karakterlánc-típus egy, illetve több karakter tárolására. A *hossz* **BYTE** típusú szám, ez határozza meg a karakterlánc hosszát. A karakterlánc memóriaigénye a lánc hosszától függ. Az elfoglalt memóriaterület első (nulladik) byte-ja a karakterlánc hosszát tartalmazza. Ezután következnek a karakterlánc byte-jai.

WORD

Előjel nélküli *egész típus*, értékkészlete: 0-tól 65535-ig, tárolása: 16 biten.

Sorszámozott típusok: egész típusok, karakter típus, felsorolt típus.

Összetett típusok

Állomány típusok: TEXT
FILE [OF *típus*]

Adatállományok kezelésére szolgáló típusok. Három fő típust különböztetünk meg, a szöveges állomány típusát (TEXT), a meghatározott rekord típusú állomány típusát (FILE OF *típus*), ahol a megadott típus a típusos állomány rekordjára vonatkozik. A nem meghatározott rekord-szerkezetű állományokat típus nélküli állományoknak nevezzük (FILE).

A szöveges állomány tárolása 256, a típusos és típus nélküli állomány 128 byte-os egység szerint történik.

A szöveges állomány sorvégein EOL, az állomány végén EOF jel található. Ez az állomány csak szekvenciálisan kezelhető, és csak írható vagy csak olvasható.

A típusos állomány alapegysége az alapkompone ns, melyek típusát a FILE OF *típus* meghatározásban a *típus* adja meg. A kezelés alapegysége az alapkompone ns. Az állomány szekvenciálisan vagy az alapkompone nsok sorszáma szerint kezelhető.

A típus nélküli állomány csak közvetlen írással és olvasással kezelhető, ahol az alapegység a byte.

Halmaz típus: SET OF *típus*

Halmaz definiálása a megadott típus értékkészletéből. A *típus* csak olyan sorszámozott típus lehet, amelynek értékkészlete legfeljebb 256 elemű, így egy halmaznak maximálisan 256 eleme lehet.

Intervallum típus: *konstans1*..*konstans2*

Valamely sorszámozott típus résztartományának tárolása. A tároláshoz szükséges byte-ok száma attól a legszűkebb értékkészletű típustól függ, amelybe belefér a megadott intervallum alsó és felső határa is.

Objektum típus

Olyan rekordtípus, amelynek mezőiben eljárások is definiálhatók.

```
OBJECT
  mezőlista
  metódualista
End
```

Mezőlista

```
mezőnév: adattípus
...
mezőnév: adattípus
```

Metódualista

```
metóduálnév: { function név |
  procedure név | constructor név |
  destructor név }
...
metóduálnév: { function név |
  procedure név | constructor név |
  destructor név }
```

Rekord típus

Összetartozó adatok tárolása. A tároláshoz szükséges memória nagyságát az állandó és a legnagyobb változó rész határozza meg.

```
RECORD
  mezőlista
  [ CASE szétválasztó adat:egyszerű
    típus OF mezőlista ]
End
```

Szétválasztó adat

Csak bizonyos körülmények között definiált adat.

Mezőlista

```
mezőnév: adattípus
...
mezőnév: adattípus
```

Tömb típus: ARRAY [*int* [,*int*...]] OF *típus*

Azonos típusú, összetartozó adatok tárolására alkalmas. A *típus* a tömb elemeinek típusa, bármilyen típus lehet (még összetett típus is). A tárolás sorfolytonosan történik, így a kellő memória méretét az alaptípus mérete és az *int* indextípus határozza meg. Az indextípus csak sorszámozott típus lehet.

FENNTARTOTT SZAVAK (KULCSSZAVAK)

FENNTARTOTT SZAVAK

AND

Logikai_kif1 **AND** *Logikai_kif2*

Logikai ÉS művelet.

ASM

ASM

assembler utasításlista
END

Assembler nyelvű utasításlista kezdete.

ARRAY

TYPE

Típus1=**ARRAY** [1..100] OF INTEGER;
Típus2=**ARRAY** ['A'..'Z'] OF CHAR;
Típus3=**ARRAY** [0..9, 0..9] OF REAL;

Tömbdeklaráció (lásd OF és *Adattípusok* fejezet!).

BEGIN

BEGIN

utasításlista
END

Összetartozó utasításcsoport kezdetét jelzi (lásd End). (Egy BEGIN...END blokk leírásunkban egyetlen utasításnak számít.)

CASE

CASE változó OF

érték1: *utasítás1*
érték2: *utasítás2*

...

ELSE

utasítás

END

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

Feltételes elágazás (lásd ELSE, END és az *Adattípusok* fejezet). A *változó* csak sorszámozható típusú lehet.

CONST

CONST

```
konstansnév[ : típus] = konstansérték;  
...
```

Konstansdeklaráció. A *típussal* ellátott konstans változóként viselkedik, melynek kezdőértéke a *konstansérték*. A lehetséges típusokat lásd az *Adattípusok* c. fejezetben!

CONSTRUCTOR

CONSTRUCTOR eljárás (paraméterek);

Az adott objektum használatához szükséges memóriaterület lefoglalása (lásd **DESTRUCTOR**).

DESTRUCTOR

DESTRUCTOR eljárás

Az adott objektum által elfoglalt memóriaterület felszabadítása (lásd **CONSTRUCTOR**).

DIV

*egész*kif1 **DIV** *egész*kif2

Egész osztás. Egész típusú adatok között végezhető el. (Csak numerikus kifejezésen belül, illetve értékdás jobb oldalán állhat.)

DO

Cikluskezdő szó (lásd FOR, WHILE és WITH).

DOWNTO

FOR *változó:=érték1* **DOWNTO** *érték2* DO
 utasítás

FOR-ral képzett ciklus esetén a lépésköz -1 lesz (lásd FOR és TO).

ELSE

IF *logikai kifejezés* THEN *utasítás1*
 ELSE *utasítás2*

A feltételes utasítás (lásd IF és CASE) azon ága, amely akkor hajtódik végre, ha a feltétel nem teljesül (a *logikai kifejezés* értéke FALSE).

END

BEGIN
 utasításlista
END

Összetartozó utasításcsoport végét jelzi (lásd BEGIN).

EXPORTS

Külső eljárás és függvény használatának engedélyezése Windows alatt használható .DLL állományból.

FILE

FILE [*OF típus*]

Állománydeklaráció meghatározott vagy változó hosszúságú rekordokkal.

FOR

FOR változó := érték1 TO érték2 DO
utasítás

Cikluskezdő utasítás (lásd TO, DOWNTO és DO).

FUNCTION

FUNCTION függvéynév (paraméterek): függvény_típusa ;

Függvényként deklarált utasításcsoport kezdetét jelző kulcsszó.

GOTO

LABEL 1, 2
GOTO 1

...
1: 1.utasítás;
2: 2.utasítás;

A program futása a megadott címke utáni első utasításon folytatódik (lásd LABEL).

IF

IF logikai kifejezés THEN utasítás1
ELSE utasítás2

Feltételes elágazás kezdő kulcsszava (lásd ELSE és THEN).

IMPLEMENTATION

Az adott UNIT-ban található utasítások és függvények kifejtését tartalmazó lista kezdetét jelző kulcsszó (lásd INTERFACE).

IN

elem **IN** *halmaz*

Megvizsgálja, hogy az *elem* eleme-e a *halmaz* halmaz típusú kifejezés értékének. Az operátor logikai (BOOLEAN) értéket ad vissza, csak logikai kifejezésben, IF, WHILE utasításban, illetve értékadás jobb oldalán használható.

INHERITED

(új)

A kulcsszót tartalmazó objektumot létrehozó (szülő) objektum egy, már felülírt metódusának újbóli felhasználása (módosított metódusaktiválás). Az INHERITED kulcsszó csak olyan objektumban használható, amelynek van elődje.

INLINE

PROCEDURE *eljárásnév* (VAR *be*; *ki*: WORD;
adat: WORD);

BEGIN

```

INLINE (
  $C4/$7E/<be/          (* LES    DI, be [ BP] *)
  $8B/$4E/<ki/          (* MOV   CX, ki [ BP] *)
  $8B/$46/<Adat/        (* MOV   AX, Adat[ BP] *)
  $FC/                  (* CLD                      *)
  $F3/$AB);            (* REP   STOSW              *)
END;
```

Gépi kódú rutin beszúrása program vagy eljárássegység (*unit*) kódjába.

INTERFACE

Eljárássegységben (*unitban*) levő utasítások és függvények alaki meghatározása (deklarálása) (lásd a UNIT és az IMPLEMENTATION szavakat!).

LABEL

LABEL 1, 2

GOTO 1

...

1: 1.utasítás;

2: 2.utasítás;

Címke Deklaráció (lásd GOTO).

LIBRARY

A megadott .DLL (Windows) állományban levő eljárások halmazának megállapítása.

MOD

egész_kifejezés1 MOD egész_kifejezés2

Maradékképzés. A művelet eredménye egész típusú. A MOD csak aritmetikai kifejezésben, illetve értékadás jobb oldalán állhat. (Lásd még a DIV szót!)

NIL

cím_típusú_változó := NIL

Beépített mutató típusú konstans, a sehová sem mutató mutatóváltozó értéke. (Szerepelhet logikai kifejezésben, egyenlőségvizsgálatban is.)

NOT

NOT logikai_kifejezés

Logikai NEM művelet. Csak kifejezésben, illetve értékadás jobb oldalán állhat.

OBJECT

OBJECT

```

    Mező;
    Mező;
    ...
    Metódu$;
    Metódu$;

```

```
END;
```

Objektumdeklaráció (lásd az Adattípusok c. fejezetet). A *metódu\$* eljárás, függvény, memóriafoglalás (konstruktor) vagy -felszabadítás (destruktor) lehet.

OF

```

változó = ARRAY[ X..Y] OF típus
változó = SET OF típus
változó = FILE OF típus
CASE változó OF (feltételek)

```

Lásd ARRAY, CASE, FILE és SET.

OR

```
Logikai_kif1 OR Logikai_kif2
```

Logikai vagy művelet, csak kifejezésben vagy értékdás jobb oldalán állhat.

PACKED

Tömörített tárolás előírása változó- vagy típusdeklaráció során. Az adott szám tárolásakor a *Turbo Pascal* nem veszi figyelembe az értékhatárokat.

PUBLIC

(új)

Objektum közös használatú (nyilvános, más objektumok számára is látható) részei előtt áll. (Lásd még a PRIVATE szót!)

PRIVATE

Objektum kizárólagos használatú (csak az objektum elemei számára látható) részeit vezeti be. (Lásd még a PUBLIC szót!)

PROCEDURE

PROCEDURE *eljárásnév (paraméterek);*

Eljárásként deklarált utasításcsoport kezdetét jelző kulcsszó.

PROGRAM

PROGRAM *név ; { programnév }*

Programkezdő kulcsszó (lásd UNIT).

Az *Turbo Pascal* program általános felépítése:

PROGRAM *név ; { programnév }*

USES ... ; { *a használt unitok* }

LABEL ... ; { *címkék* }

CONST ... ; { *konstansok* }

TYPE ... ; { *típusok* }

VAR ... ; { *változók* }

PROCEDURE ... ; { *eljárások* }

FUNCTION ... ; { *függvények* }

BEGIN

program; { Pascal vagy Assembler nyelvű utasításlista }

...

END.

RECORD

RECORD

```

    mező ;
    mező ;
    ...
    mező ;
END;
```

vagy

RECORD

```

    mező;
    ...
    CASE változó: TYPE OF
    érték1: ( mező );
    ...
    értékn: ( mező )
END;
```

Összetartozó adatok tárolására alkalmas típus deklaráció (lásd az *Adattípusok* c. fejezetben).

REPEAT

REPEAT

```

    utasítás;
    utasítás;
    ...
    utasítás
UNTIL feltétel;
```

Hátultesztelő ciklus kezdete (lásd UNTIL).

SET

TYPE

```

Nap = (Sun, Mon, Tue, Wed, Thu, Fri, Sat);
Betű = SET OF CHAR;
Szám = SET OF 0..9;
Napok = SET OF Nap;
```

Halmaztípus neve (lásd az OF szót és az *Adattípusok* c. fejezetet).

SHL

egész_kif1 **SHL** *egész_kif2*

Az első kifejezés értékét balra tolja a második kifejezés értékének megfelelő számú helyiértékkal. A művelet csak numerikus kifejezésben, illetve értékadás jobb oldalán szerepelhet.

SHR

egész_kif1 **SHR** *egész_kif2*

Az első kifejezés értékét jobbra tolja a második kifejezés értékének megfelelő számú helyiértékkal. A művelet csak egész típusú kifejezésben, illetve művelet jobb oldalán szerepelhet.

STRING

VAR

változó: **STRING** [*egész_konstans*]

A karakterlánc-típus neve (lásd az *Adattípusok* c. fejezetet).

THEN

IF *logikai kifejezés* **THEN** *utasítás1*
 ELSE *utasítás2*

A feltételes utasítás (IF) igaz ágának kezdetét jelző kulcsszó (lásd IF és ELSE).

TO

FOR *változó* := *érték1* **TO** *érték2* DO
 utasítás

For-ciklus esetén a lépésköz +1 lesz (lásd FOR és DOWNTO).

TYPE

TYPE

```
típusnév1 = típus;
```

```
...
```

```
típusnévn = típus;
```

Típusdeklaráció-sorozat kezdetét jelző kulcsszó.

UNIT

UNIT név; { A létrehozott UNIT neve }

Eljárás egység (*unit*) kötelező kezdőszava (lásd még a PROGRAM szót).

Az eljárás egységek (*unitok*) általános felépítése:

```
UNIT név;           { UNIT-név }
INTERFACE         { lásd INTERFACE } :
USES ...;         { a használt unitok }
CONST ...;        { konstansok }
TYPE ...;         { típusok }
VAR ...;          { változók }
PROCEDURE ...;   { eljárások }
FUNCTION ...;    { függvények }
IMPLEMENTATION { lásd IMPLEMENTATION } :
USES ...;         { a használt unitok }
LABEL ... ;      { címkék }
CONST ... ;      { konstansok }
TYPE ... ;       { típusok }
VAR ... ;        { változók }
PROCEDURE ...;   { eljárások }
FUNCTION ... ;   { függvények }
BEGIN
    program;       { Pascal vagy Assembler
                    nyelvű program }
    ...
END.
```

UNTIL

```
REPEAT
  utasítás;
  utasítás;
  ...
  utasítás
UNTIL feltétel;
```

A hátul tesztelő ciklus feltételének megadása (lásd még a REPEAT szót).

USES

```
USES UNIT1, UNIT2, UNIT3;
```

A programban vagy eljárásegységben felhasznált eljárás-egységek (*unitok*) felsorolását jelző kulcsszó a program (vagy eljárás-egység) elején.

VAR

```
VAR
  változó, ... változó: típus;
  ...
  változó, ... változó: típus;
```

A programban, eljárás-egységben, eljárásban vagy függvényben felhasznált változók deklarálása.

WHILE

```
WHILE logikai_kifejezés DO utasítás
```

Elöltesztelő ciklus kezdete (lásd még a DO szót).

WITH

```
WITH rekord DO utasítás
```

Műveletek a *rekord* mezőin úgy, hogy a mezők nevei a rekord neve nélkül, önálló változókként használhatók. (lásd még a DO szót).

XOR

kifejezés1 XOR kifejezés2

Logikai kizáró vagy. A művelet csak (aritmetikai vagy logikai) kifejezésben, illetve értékadás jobb oldalán állhat.

FORDÍTÁSI DIREKTÍVÁK

FORDÍTÁSI DIREKTÍVÁK

\$A : (+, -; alapértelmezés: -)

A változók által lefoglalt memóriaterületek bekapcsolt állapotban szóhatáron kezdődnek (két byte-onként), különben byte-határon (byte-onként). Bekapcsolt állapota gyorsítja a program futását.

\$B : (+, -; alapértelmezés: -)

A feltételes utasítást bekapcsolt állapotban teljesen kiértékeli. Kikapcsolva csak addig értékeli ki, amíg a feltétel eredménye nem egyértelmű. Kikapcsolt állapotban a program futása gyorsabb, az esetleges hibák azonban nem derülnek ki.

\$C attribútum attribútum : (alapértelmezés: lent)

A kódszegmens állapotának ellenőrzését végző globális direktíva.

Alapértelmezés:**\$C**

Moveable

Demandload

Discardable

MOVEABLE

A kódszegmens helye szabadon változtatható.

FIXED

A kódszegmens helye nem változtatható.

PRELOAD

A kódszegmens a program futásakor töltődik be.

DEMANDLOAD

A kódszegmens csak szükség esetén töltődik be.

PERMAMENT

A kódszegmens egyszeri betöltés után a memóriában marad.

DISCARDABLE

A kódszegmens felszabadítható, ha nem hosszabb, mint szükséges.

\$D : (+, -; alapértelmezés: -)

Bekapcsolt állapotban fordításkor információs állományt hoz létre, amelyben a program jellemzőit, fordítási üzeneteit tárolja. Ennek segítségével a program könnyebben vizsgálható, nyomkövethető. Kikapcsolt állapotban a program mérete csökken.

\$D *szöveg*

A megadott szöveg bekerül az EXE vagy a DLL állomány fejlécébe.

\$DEFINE *szimbólum*

Feltételes fordításhoz létrehozza a megadott szimbólumot. A szimbólum hossza korlátlan, a fordító azonban csak az első 64 karaktert veszi figyelembe. Ha a szimbólum már létezik, a direktíva hatástalan.

\$E : (+, -; alapértelmezés: -)

Bekapcsolt állapotban a programhoz fűzi a 80x87 aritmetikai társprocesszort helyettesítő rutint. Így, ha a gépben nincs társprocesszor, a program emulálja. Ha a gépben van társprocesszor, akkor a direktíva hatástalan.

\$ELSE

Ha a feltételes fordítást indító direktívában megadott feltétel nem teljesül, a fordítás a \$ELSE utáni programrészrel folytatódik, és az \$ENDIF direktíváig tart.

\$ENDIF

A feltételes fordítás vége.

\$F : (+, -; alapértelmezés: -)

Eljáráshívási mód kiválasztása. Bekapcsolt állapotban csak FAR hívások kerülnek a programba, azaz a címek négy byte-osak. Kikapcsolt állapotban a Turbo Pascal a FAR és a NEAR hívási módot is használja.

\$G : (+, -; alapértelmezés: -)

80286-os kód generálása: a direktíva bekapcsolt állapotában a fordító használja a 80286-os processzor ENTER, LEAVE, PUSH, IMUL, SHL, SHR utasításait is.

\$I : (+, -; alapértelmezés: +)

Ki-, illetve bekapcsolja az beviteli/kiviteli egységek működésének vizsgálatát. Bekapcsolt állapota esetén, ha beviteli/kiviteli hiba van, a program hibaüzenettel leáll. Ha a direktíva ki van kapcsolva, a program nem áll le, és a kapott hibakód az IOResult függvénnyel vizsgálható. A direktíva kikapcsolt állapota esetén az első beviteli/kiviteli hiba után a további beviteli/kiviteli műveletek nem hajtódnak végre.

\$I *meghajtó:\út\állománynév*

A fordító a megadott állományt befordítja a direktíva helyére. Ezzel a módszerrel több (mintegy tizenöt) szinten ágyazhatók állományok egymásba. A befűzendő állomány nevének kiterjesztése alapállapotban .PAS. Ha nincs megadva útvonal, a fordító először az aktuális, majd pedig az *Options* menü *Directories...* menüpontjánál megadott könyvtárakban keres.

\$IFDEF *szimbólum*

Ha a direktívában megadott szimbólum létezik, akkor a fordító az \$ELSE, illetve az \$ENDIF direktívákig fordítja le a forrásállomány sorait.

\$IFNDEF *szimbólum*

Ha a direktívában megadott szimbólum nem létezik, akkor a fordító az \$ELSE, illetve az \$ENDIF direktívákig fordítja le a forrásállomány sorait.

\$IFOPT *direktívaállapot*

Ha a direktíva után megadott direktívaállapot (pl. \$L+) igaz, akkor a fordító az \$ELSE, illetve az \$ENDIF direktívákig fordítja le a forrásállomány sorait.

\$L : (+, -; **alapértelmezés: +**)

Bekapcsolt állapotban információt generál a lokális szimbólumokról. Ez a direktíva csak a \$D direktíva bekapcsolt állapotában mellett használható.

\$L *meghajtó: \út\állománynév*

A fordító a megadott tárgykód-állományt befűzi a direktíva helyére. A befűzendő állomány nevének kiterjesztése alapértelmezés szerint .OBJ. Ha nincs megadva *út*, a fordító először az aktuális, majd pedig az *Options* menü *Directories...* menüpontjánál megadott könyvtárakban keres. A tárgykód-állomány PUBLIC rutinjait a Pascal programban az EXTERNAL kulcsszóval deklarálni kell.

\$M *verem, heapmin, heapmax*

A program által használt memóriaterület méretének meghatározása. A *verem* a verem méretét (1024-65520), a *heapmin* és a *heapmax* a munkamemória alsó (0-655360) és felső (*heapmin*-655360) határát adja meg.

\$N : (+, -; **alapértelmezés: +**)

Bekapcsolt állapotában a lebegőpontos számítások az aritmetikai társprocesszor (80x87) segítségével hajtódnak végre. Kikapcsolt állapotban a Turbo Pascal belső lebegőpontos rutinjai működnek. Ekkor csak a REAL típus használható. (A társprocesszor emulálható is a \$E direktíva segítségével.)

\$O : (+, -; **alapértelmezés: -**)

Engedélyezi az eljárás egység átlapoló (Overlay) szerkezetbe szervezését.

\$O *eljárássegység*

Átlapoló (overlay) szerkezetűvé alakítja a megadott eljárássegységet (*unitot*). (Az eljárássegységet a direktíva használata előtt le kell fordítani az \$O+ direktívával.) Az átlapolt program neve a forrásállomány nevét örökli, kiterjesztése pedig .OVR.

\$P : (+, -; **alapértelmezés:** -) (új)

Nyitott karakterláncok használatának engedélyezése. Az \$P+ használata esetén, ha egy függvény vagy eljárás egyik paramétere karakterlánc, akkor ott a függvény vagy eljárás hívásakor különböző hosszúságú karakterláncok állhatnak. Hasonlóan alkalmazható a **System** eljárássegység *OpenString* típusazonosítója is. (*OpenString* alkalmazása esetén a \$P direktíva alapota lényegtelen.)

\$Q : (+, -; **alapértelmezés:** -) (új)

Bekapcsolt állapotban a program futása az egész típusú aritmetikai műveletek túlcsoorduláskor 215-ös számú hibával megáll. Működése a \$R direktíva használatával jól kiegészíthető. A kód méretét növeli.

\$R : (+, -; **alapértelmezés:** -)

Bekapcsolt állapotban engedélyezi az értékhatárokat ellenőrző kód generálását (indexelés, sorszámozott típusú változók értékadása, objektumellenőrzés). Hiba esetén a program futása hibaüzenettel leáll. Használata a kód méretét növeli, futását lassítja.

\$R *állománynév*

Eljárásokat vagy könyvtárjegyzéket tartalmazó erőforrásállomány megadása. Az *állománynév* kiterjesztése alapértelmezés szerint .RES. Lokális direktíva, csak védett üzemmódban (DOS vagy Windows alatt) használható.

\$S : (+, -; alapértelmezés: +)

A veremtúlsordulás figyelésének bekapcsolása, illetve kikapcsolása. Bekapcsolt állapotában a fordító minden szubrutinhívás mellé kódot generál a veremtúlsordulás ellenőrzéséhez. Hatására hiba (pl. végtelen rekurzió) esetén a program futása hibaüzenettel leáll. Ha a direktíva ki van kapcsolva, veremtúlsordulás esetén a rendszer összeomolhat.

\$T : (+, -; alapértelmezés: -) (új)

Bekapcsolt állapotában ellenőrzi a @ operátor (címooperátor) által generált mutatók típusát. Aktivitásokor adott típusú változóhoz tartozó mutató csak olyan mutatóval kompatibilis, amely ugyanolyan típusú változóra mutat. A direktíva kikapcsolt állapotában a @-művelet eredménye típus nélküli mutató.

\$UNDEF *szimbólum*

Törli a \$DEFINE direktíva által létrehozott feltételes szimbólumot. Ha a megadott szimbólum (már) nem létezik, a direktíva hatástalan.

\$V : (+, -; alapértelmezés: +)

Típusazonosság ellenőrzése karakterláncok változóként való átadásánál.

\$W : (+, -; alapértelmezés: +)

Speciális „előszót” és „utószót” generál távoli eljárások és függvények hívásakor.

\$X : (+, -; alapértelmezés: +)

Bekapcsolt állapotban a Turbo Pascal függvényei (kivéve a System UNIT függvényeit) eljárásként hívhatók. Ilyenkor a visszatérési értékek figyelmen kívül maradnak.

\$Y : (+, -; alapértelmezés: +) (új)

Ha be van kapcsolva, a fordító létrehozza a szimbólumok referencia-táblázatát, azaz a generált kódba *böngésző-(browse-)*információkat ír. (Csak védett módban, a TPX-ben használható.)

ELJÁRÁSEGYSÉGEK

CRT UNIT

KONSTANSOK

(képernyő-üzemmódok és színek)

Üzem módok:

BW40=0

BW80=2

Mono=7

C40=1

CO40=1

CO80=3

Font8x8=256

C80=3

Színek:

Black=0

Blue=1

Green=2

Cyan=3

Red=4

Magenta=5

Braun=6

LightGray=7

DarkGray=8

LightBlue=9

LightGreen=10

LightCyan=11

LightRed=12

LightMagenta=13

Yellow=14

White=15

Blink=128

VÁLTOZÓK

CHECKBREAK: BOOLEAN

Ha értéke igaz (true), a program futása a Ctrl-Break billentyűkombinációval bármikor megszakítható.

CHECKEOF: BOOLEAN

Ha értéke igaz (true), a konzolról való adatbevitel a Ctrl-Z karakterrel befejezhető (azaz READ-del vagy READLN-nel olvasáskor az EOF függvény igaz értéket ad vissza).

CHECKSNOW : BOOLEAN

Engedélyezi (igaz, TRUE) vagy megtiltja (hamis, FALSE) a „havazás” ellenőrzését. Akkor kell használni, ha CGA képernyő esetén közvetlenül írunk a képernyő-memóriába (karakteres üzemmódokban).

DIRECTVIDEO : BOOLEAN

Ha értéke igaz, a WRITE és WRITELN eljárások közvetlenül a képernyő-memóriába írnak, egyébként a BIOS-t használják.

LASTMODE : BYTE

Az legutoljára beállított karakteres üzemmód sorszám.

TEXTATTR : WORD

Az aktuális karakter alatti szín és a karakter színe. A felső byte a karakter alatti szín, az alsó pedig a karakter színének sorszám.

WINDMIN : WORD

A megnyitott ablak, illetve képernyő bal felső sarkának koordinátái.

WINDMAX : WORD

A megnyitott ablak, illetve képernyő jobb alsó sarkának koordinátái.

FÜGGVÉNYEK ÉS ELJÁRÁSOK

ASSIGNCRT : eljárás

A megadott szöveges állományt az aktuális konzolhoz rendeli.

ASSIGNCRT (*F*)

F

A szövegállomány (TEXT típusú) változója .

CLREOL : eljárás

A kurzorpozíciótól a sor végéig törli a sort az háttérszín felhasználásával. A kurzorpozíció nem változik.

CLREOL

CLRSCR: eljárás

Törli az aktuális ablak tartalmát, és a kurzort a bal felső sarokba (1,1) állítja.

CLRSCR

DELAY: eljárás

Felfüggeszti a program futását a megadott időtartamra.

DELAY (MS)

MS

A várakozás időtartama miliszekundumban (WORD típusú).

DELLINE : eljárás

Az aktuális ablakban törli a kurzor sorát, és az utána következő sorokat följebb húzza. A kurzor helye változatlan marad.

DELLINE

GOTOXY : eljárás

A kurzort a koordinátaival megadott pozícióra mozgatja az aktuális ablakon belül.

GOTOXY (X, Y)

X

Az oszlop sorszáma (BYTE típusú).

Y

A sor száma (BYTE típusú).

HIGHVIDEO : eljárás

Az eljárás hívása után kiírt karakterek nagy fényerővel kerülnek a képernyőre.

HIGHVIDEO

INLINE : eljárás

Üres sor beszúrása az aktuális ablakban a kurzor helyétől kezdve.

INLINE

KEYPRESSED : függvény

Ha volt lenyomott – és még fel nem dolgozott – billentyű, akkor igaz (TRUE), ellenkező esetben hamis (FALSE).

KEYPRESSED : BOOLEAN

Lásd a READ eljárást és a READKEY függvényt!

LOWVIDEO : eljárás

Az eljárás hívása után kiírt karakterek alacsony fényerővel kerülnek a képernyőre.

LOWVIDEO

NORMVIDEO : eljárás

Az eljárás hívása után kiírt karakterek az eredeti fényerővel kerülnek a képernyőre.

NORMVIDEO

NOSOUND : eljárás

Kikapcsolja a számítógép belső hangszóróját.

NOSOUND

READKEY : függvény

Beolvas egy karaktert a billentyűzetről.

READKEY : CHAR

SOUND : eljárás

Bekapcsolja a számítógép hangszóróját, és azon megadott frekvenciájú hangot ad.

SOUND (HZ)

HZ

A hang frekvenciája (WORD típusú).

TEXTBACKGROUND : eljárás

A háttérszín megadása karakteres üzemmódokban.

TEXTBACKGROUND (*COLOR*)

COLOR

A háttérszín kódja (BYTE típusú).

TEXTCOLOR : eljárás

A kiírt szöveg színének megadása.

TEXTCOLOR (*COLOR*)

COLOR

Színkód (BYTE típusú).

TEXTMODE : eljárás

Karakteres üzemmód beállítása.

TEXTMODE (*MODE*)

MODE

A képernyő-üzemmód száma (BYTE típusú).

WHEREX: függvény

A kurzor oszlopának sorszáma az aktuális ablakon vagy a képernyőn belül.

WHEREX : BYTE

WHEREY : függvény

A kurzor sorának száma az aktuális ablakon, illetve a képernyőn belül.

WHEREY : BYTE

WINDOW : eljárás

Az aktuális ablak méretének (két sarkának) beállítása.

WINDOW (*X1, Y1, X2, Y2*)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

X1 , Y1

A ablak egyik sarkának koordinátái (WORD típusúak).

X2 , Y2

Az ablak áttellenes sarkának koordinátái (WORD típusúak).

DOS UNIT

TÍPUSOK

A CPU regiszterei:

```
Registers = Record  
  Case Integer Of  
    0: (AX,BX,CX,DX,BP,SI,DI, DS,ES,Flags: Word);  
    1: (AL,AH,BL,BH,CL,CH,DL,DH: Byte);  
End;
```

Dátumrekord-típus:

```
DateTime = Record  
  Year,Month,Day,Hour,Min,Sec: Word;  
End;
```

Állománykereső rekord:

```
TSearch = Record  
  Fill : array[1..21] of Byte;  
  Attr : Byte;  
  Time : Longint;  
  Size : Longint;  
  Name : string[12];  
End;
```

Parancssor:

```
ComStr = String[127];
```

Elérési út:

```
PathStr = String[79];
```

Alkönyvtárnév:

```
DirStr = String[67];
```

Állománynév:

```
NameStr = String[8];
```

Kiterjesztés:

ExtStr = String[4];

Szövegpuffer:

TextBuf : Array[0..127] Of Char;

Megnyitott típusos és típus nélküli állomány:

TFileRec = record

Handle: Word;

Mode: Word;

RecSize: Word;

Private: array[1..26] of Byte;

UserData: array[1..16] of Byte;

Name: array[0..79] of Char;

End;

Megnyitott szöveges állomány:

PTextBuf = ^TTextBuf;

TTextBuf = array[0..127] of Char;

TTextRec = record

Handle: Word;

Mode: Word;

BufSize: Word;

Private: Word;

BufPos: Word;

BufEnd: Word;

BufPtr: PTextBuf;

OpenFunc: Pointer;

InOutFunc: Pointer;

FlushFunc: Pointer;

CloseFunc: Pointer;

UserData: array[1..16] of Byte;

Name: array[0..79] of Char;

Buffer: TTextBuf;

End;

KONSTANSOK

Flag-konstansok:

fCarry=\$0001
fParity=\$0004
fAuxiliary=\$0010
fZero=\$0040
fSign=\$0080
fOverflow=\$0800

Állománymegnyitási módok:

fmClosed=\$D7B0
fmInput=\$D7B1
fmOutput=\$D7B2
fmInOut=\$D7B3

Állományattribútumok:

ReadOnly=\$01
Hidden=\$02
SysFile=\$04
VolumeID=\$08
Directory=\$10
Archive=\$20
AnyFile=\$3F

VÁLTOZÓK

DosError : Integer;

2 File not found

Az állomány nincs meg.

3 Path not found

Az elérési út nincs meg.

5 Access denied

A hozzáférési kérelem visszautasítva.

6 Invalid handle

Érvénytelen állományazonosító.

8 Not enough memory

Nincs elég memória.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

10 Invalid environment

Érvénytelen környezet vagy környezetleíró változó.

11 Invalid format

Érvénytelen formátum.

18 No more files

Nincs több állomány.

FÜGGVÉNYEK ÉS ELJÁRÁSOK

DISKFREE : függvény

A megadott meghajtóban lévő lemez szabad területének nagysága byte-ban.

DISKFREE (*DRIVE*) : LONGINT

DRIVE

A meghajtó sorszáma (A: 1, B: 2, C: 3, stb., BYTE típusúak).

DISKSIZE : függvény

A megadott meghajtóban levő lemez teljes kapacitását adja meg byte-ban.

DISKSIZE (*DRIVE*) : LONGINT

DRIVE

A meghajtó sorszáma (A: 1, B: 2, C: 3, stb., BYTE típusúak).

DOEXITCODE : függvény

Az EXEC eljárással elindított állomány visszatérési kódját adja meg. Az alsó byte a futási kódot (lásd még KEEP és SYSTEM/HALT), a felső byte a futás eredményének jelzőszámát tartalmazza. A byte olvasása után az értékek törlődnek.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

A futási eredmény lehetséges értékei:

- 0 Hibátlan futás.
- 1 Ctrl-C billentyűkombinációval megszakítva.
- 2 Beviteli/kiviteli hiba.
- 3 KEEP eljárással megszakítva.

DOSEXITCODE : WORD

DOSVERSION : függvény

A számítógépen futó operációs rendszer verziószámát adja meg. Az alsó byte a verziószámot, a felső byte az alverziószámot tartalmazza.

DOSVERSION : WORD

ENVCOUNT : függvény

A környezetleíró változók számával tér vissza.

ENVCOUNT : INTEGER

ENVSTR : függvény

A megadott sorszámú környezetleíró változóval tér vissza. Ha a megadott sorszám kisebb, mint 1 vagy nagyobb, mint az ENVCOUNT visszatérési értéke, akkor az eredmény üres karakterlánc lesz. A visszaadott karakterlánc formája: *a környezeti változó neve=a változó értéke.*

ENVSTR (INDEX) : String

INDEX

A környezetleíró változó sorszáma (INTEGER típusú).

EXEC : eljárás

Elindítja a PATH-ban megadott programot a CMDLINE-ban megadott paraméterekkel.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

`EXEC (PATH, CMDLINE)`

PATH

A program neve és elérési útja (STRING típusú).

CMDLINE

Az elindított program által igényelt DOS-parancssorbeli paraméterek (STRING típusú).

FEXPAND : függvény

A megadott állománynévhez hozzáfűzi az állományt tartalmazó meghajtó betűjelét és az elérési utat.

`FEXPAND (PATH) : PathStr`

PATH

Az állomány neve (PathStr típusú).

FINDFIRST : eljárás

A *PATH* paraméterben megadott állomány- vagy állománycsoport-név alapján a megkeresi az első *ATTR* attribútumú bejegyzést a megfelelő tartalomjegyzékben és elhelyezi az *F* változóban.

`FINDFIRST (PATH, ATTR, F)`

PATH

Az elérési út (CHAR-POINTER típusú).

ATTR

A keresett attribútum. Az archív attribútumot mindig keresi.

F

A célváltozó (TSearchRec típusú).

FINDNEXT : eljárás

A FINDFIRST eljárással elindított keresés folytatása az ott megadott paraméterek szerint.

FINDNEXT (*F*)

F

A célváltozó (TSearchRec típusú, amelyet

Type TSearchRec = Record

Fill : array[1..21] of Byte;

Attr : Byte;

Time : Longint;

Size : Longint;

Name : string[12];

End;

alakban képzünk).

FSEARCH : függvény

A *PATH* paraméterben megadott állományt megkeresi az aktuális meghajtó aktuális könyvtárában. Ha ott nem találja, akkor az állományt *DIRLIST*-ben megadott, egymástól pontosvesszővel elválasztott elérési utakban keresi tovább.

FSEARCH (*PATH,DIRLIST*) : PathStr

PATH

A keresett állomány neve (PathStr típusú).

DIRLIST

További elérési utak (STRING típusú).

FSPLIT : eljárás

Állományspecifikáció felbontása elérési útra, állománynévre és kiterjesztésre.

FSPLIT (*PATH,DIR,NAME,EXT*)

PATH

Az állomány teljes neve (specifikációja, PathStr típusú).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

DIR

Ebbe a változóba kerül az állomány elérési útja (DirStr típusú).

NAME

Ebbe a változóba kerül az állomány neve (NameStr típusú).

EXT

Ebbe a változóba az állomány kiterjesztése kerül (ExtStr típusú).

GETCBREAK : eljárás

A Ctrl-Break billentyűkombináció működésének lekérdézése. Ha a visszaadott érték igaz, a program futása bárhol megszakítható a Ctrl-Break billentyűvel, ha hamis, akkor csak beviteli/kiviteli művelet közben.

GETCBREAK (Var *BREAK*: Boolean)

BREAK

A figyelésre beállított változó (BOOLEAN típusú).

GETDATE : eljárás

A paraméterekben megadott változókba helyezi a dátum megfelelő összetevőit.

GETDATE (Var *YEAR*, *MONTH*, *DAY*, *DAYOFWEEK*:
Word)

YEAR

Év (WORD típusú).

MONTH

Hónap (WORD típusú).

DAY

Nap (WORD típusú).

DAYOFWEEK

A hét napja (WORD típusú).

GETENV : függvény

A névvel megadott környezetleíró változó értékét adja vissza. Ha a megadott változó nem létezik, a függvény értéke üres karakterlánc lesz.

GETENV(*ENVVAR*: String): String

ENVVAR

A kívánt környezetleíró változó neve (STRING típusú).

GETFATTR : eljárás

A megadott logikai állományhoz tartozó fizikai állomány attribútumaival tér vissza.

GETFATTR(*F*: File, *Var ATTR*:Word)

F

A vizsgált állomány azonosítója (FILE típusú).

ATTR

Az állomány tulajdonságait tartalmazó változó (WORD típusú).

GETFTIME : eljárás

A megadott logikai állományhoz tartozó fizikai állomány utolsó mentési idejét a *TIME* változóba helyezi.

GETFTIME (*F*:File, *TIME*:Longint)

F

A vizsgált állomány (FILE típusú).

TIME

Az időpont változója (LONGINT típusú).

GETINTVEC : eljárás

A megadott megszakítás kezelőrutinjának kezdőcímét adja vissza.

GETINTVEC (*INTNO*: Byte, *Var VECTOR*:
Pointer)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

INTNO

A megszakítás száma (BYTE típusú).

VECTOR

A kezdőcím változója (POINTER típusú).

GETTIME : eljárás

A paraméterekben megadott változókba helyezi az aktuális idő komponenseit.

GETTIME (Var *HOUR*, *MINUTE*, *SECOND*, *SEC100*
:Word)

HOUR

Az óraváltozó (WORD típusú).

MINUTE

A percváltozó (WORD típusú).

SECOND

A másodperc-változó (WORD típusú).

SEC100

A századmásodperc-változó (WORD típusú).

GETVERIFY : eljárás

A paraméterként megadott változóban (logikai értékkel) visszaadja, hogy a lemezműveletek ellenőrzése (*verify*) be van-e kapcsolva.

GETVERIFY (Var *VERIFY*: Boolean)

VERIFY

A változó értéke igaz (TRUE), ha az ellenőrzés be van kapcsolva, egyébként hamis (FALSE) (BOOLEAN típusú).

INTR : eljárás

Az *INTNO* paraméterrel megadott szoftvermegszakítást kéri a a *REGS* változóban megadott regiszterértékekkel.

INTR (*INTNO*: *Byte*, *Var REGS*: *TRegisters*)

INTNO

A megszakítás sorszáma (BYTE típusú).

REGS

A regiszterek értékei (TREGISTERS típusú, amely RECORD

CASE INTEGER OF

0 : (AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : WORD);

1 : (AL,AH,BL,BH,CL,CH,DL,DH : BYTE);

End;

End

módon hozható létre).

KEEP : eljárás

A meghívott program futása felfüggesztődik, s vezérlés újra a hívó programhoz kerül. Hátránya, hogy a meghívott és a KEEP által felfüggesztett program futásához lefoglalt memória nem szabadul fel.

KEEP (*EXITCODE*: *Word*)

EXITCODE

A kilépés kódja, amely programon belül a *DosExitCode* függvénnyel (lásd ott), DOS-ban pedig az *ERRORLEVEL* változóval kezelhető (WORD típusú).

MSDOS : eljárás

Az 21H számú szoftvermegszakítás kérése.

MSDOS (*Var REGS*: *TRegisters*)

REGS

A megszakítás meghívásakor az itt megadott értékek töltődnek be a regiszterekbe (TREGISTERS típusú).

PACKTIME : eljárás

A DATETIME rekordot tömörített számmá alakítja.

PACKTIME (Var T: DateTime, Var TIME
: Longint)

T

A tömörítendő dátum és idő (DATETIME típusú, amely RECORD

YEAR, MONTH, DAY, HOUR, MIN, SEC : WORD;

End

módon képezhető).

TIME

A tömörített szám változója (LONGINT típusú).

SETCBREAK : eljárás

A *BREAK* paraméter értékének megfelelően be-, illetve kikapcsolja a Ctrl-Break billentyű-kombináció figyelését.

SETCBREAK (*BREAK*: Boolean)

BREAK

A figyelést irányító változó. Ha értéke igaz (TRUE), a Ctrl-Break használható, egyébként nem (BOOLEAN típusú).

SETDATE : eljárás

A rendszerdátum beállítása.

SETDATE (*YEAR, MONTH, DAY*: Word)

YEAR

Évszám (WORD típusú).

MONTH

Hónap (WORD típusú).

DAY

Nap (WORD típusú).

SETFATTR : eljárás

A megadott logikai állományhoz tartozó fizikai állomány attribútumainak beállítása.

SETFATTR (Var *F*: *File*, *ATTR*: *Word*)

F

A állomány (FILE típusú).

ATTR

Az állomány tulajdonságait meghatározó érték, vagy az azt tartalmazó változó (WORD típusú).

SETFTIME : eljárás

A megadott logikai állományhoz tartozó fizikai állomány utolsó mentési időpontjának beállítása.

SETFTIME (Var *F*: *File*, *TIME*: *Longint*)

F

Az állomány (FILE típusú).

TIME

A beállítandó időpont (LONGINT típusú).

SETINTVEC : eljárás

Az *INTNO* változóban megadott megszakításvektor beállítása.

SETINTVEC (*INTNO*: *Byte*, Var *VECTOR*
: *Pointer*)

INTNO

A megszakítás száma (BYTE típusú).

VECTOR

A megszakítás kezdőcíme (POINTER típusú).

SETTIME : eljárás

A rendszeridő beállítása.

SETTIME (*HOURL*, *MINUTE*, *SECOND*, *SEC100*: *Word*)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

HOUR

Óra (WORD típusú).

MINUTE

Perc (WORD típusú).

SECOND

Másodperc (WORD típusú).

SEC100

Századmásodperc (WORD típusú).

SETVERIFY : eljárás

Engedélyezi, illetve megtiltja a lemezműveletek ellenőrzését. A DOS VERIFY változóját állítja.

SETVERIFY (VERIFY: Boolean)

VERIFY

A kapcsoló értéke; ha igaz (TRUE), akkor a figyelés aktív (BOOLEAN típusú).

SWAPVECTORS : eljárás

A System UNIT-ban definiált SavelntXX változók értékét felcseréli a megszakítási vektorok értékével.

SWAPVECTORS

UNPACKTIME : eljárás

Tömörített számot DATETIME rekorddá alakít.

UNPACKTIME (TIME: LongInt, Var DT: DateTime)

TIME

A tömörített szám (LONGINT típusú).

DT

A dátum és az idő változója (DATETIME típusú).

GRAPH UNIT

TÍPUSOK

Körív, ellipszis:

```
ArcCoordsType = record  
  X, Y,  
  Xstart, Ystart,  
  Xend, Yend : integer;  
end;
```

Kitöltési minta:

```
FillPatternType = array [1..8] of Byte;
```

A festőszín és az aktuális szín rekordja:

```
FillSettingsType = record  
  Pattern : Word;  
  Color : Word;  
end;
```

Vonaltípus:

```
LineSettingsType = record  
  LineStyle : Word;  
  Pattern : Word;  
  Thickness : Word;  
end;
```

A grafikus ablak mérete és színe:

```
PaletteType = record  
  Size : Byte;  
  Colors : array[0..MaxColors] of Shortint;  
end; (MaxColors = 16)
```

Képpont:

```
PointType = record  
  X, Y : integer;  
end;
```

Betűtípus:

```

TextSettingsType = record
  Font      : Word;
  Direction : Word;
  CharSize  : Word;
  Horiz     : Word;
  Vert      : Word;
end;

```

Grafikus ablak mérete:

```

ViewPortType = record
  x1, y1, x2, y2 : integer;
  Clip           : Boolean;
end;

```

KONSTANSOK**A BAR3D eljárásnál az ablak árnyékolásának engedélyezése, illetve letiltása:**

TopOn: True: Árnyékolás engedélyezése
TopOff: False: Árnyékolás letiltása

Fedési attribútumok (bit-bit operátorok):

NormalPut	0	MOV
CopyPut	0	MOV
XorPut	1	XOR
OrPut	2	OR
AndPut	3	AND
NotPut	4	NOT

A grafikus ablakon kívüli rajzolás engedélyezése, illetve letiltása:

ClipOn: True: Engedélyezés
ClipOff: False: Letiltás

Színek:

Black=0 (fekete)
Blue=1 (kék)
Green=2 (zöld)
Cyan=3 (kékeszöld)
Red=4 (vörös)
Magenta=5 (liláskék)
Braun=6 (barna)
LightGray=7 (világosszürke)
DarkGray=8 (sötétszürke)
LightBlue=9 (világoskék)
LightGreen=10 (világoszöld)
LightCyan=11 (világoscián)
LightRed=12 (rózsaszín)
LightMagenta=13 (világoslila)
Yellow=14 (sárga)
White=15 (fehér)
Blink=128 (villogás)

A 8514-es (EGA) monitor színei:

EGABlack=0 (fekete)
EGABlue=1 (kék)
EGAGreen=2 (zöld)
EGACyan=3 (kékeszöld)
EGARed=4 (vörös)
EGAMagenta=5 (liláskék)
EGABraun=20 (barna)
EGALightGray=7 (világosszürke)
EGADarkGray=56 (sötétszürke)
EGALightBlue=57 (világoskék)
EGALightGreen=58 (világoszöld)
EGALightCyan=59 (világoscián)
EGALightRed=60 (rózsaszín)
EGALightMagenta=61 (világoslila)
EGAYellow=62 (sárga)
EGAWHITE=63 (fehér)

Kitöltési konstansok:

EmptyFill	0	Üres minta.
SolidFill	1	Telt minta.
LineFill	2	Vonalazott.
LtSlashFill	3	Jobbra döntött.
SlashFill	4	Jobbra döntött vastag.
BkSlashFill	5	Balra döntött vastag.
LtBkSlashFill	6	Balra döntött.
HatchFill	7	Négyzetrácsos.
XHatchFill	8	Keresztrácsos.
InterleaveFill	9	Sejtmintás.
WideDotFill	10	Pontozott.
CloseDotFill	11	Zárt pontozott.
UserFill	12	Felhasználói minta.

Grafikus meghajtók:

CurrentDriver	-128	(Aktuális meghajtó)
Detect	0	(A gépben lévő grafikus kártyát automatikusan felismerő beállítás)
CGA	1	
MCGA	2	
EGA	3	
EGA64	4	
EGAMono	5	
IBM8514	6	
HercMono	7	
ATT400	8	
VGA	9	
PC3270	10	

Grafikus üzemmódok:

CGAC0	0	320 x 200
CGAC1	1	320 x 200
CGAC2	2	320 x 200
CGAC3	3	320 x 200
CGAHi	4	640 x 200
EGAHi	1	640 x 350
EGA64Lo	0	640 x 200
EGA64Hi	1	640 x 350

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

MCGAC0	0	320 x 200
EGALo	0	640 x 200
MCGAC1	1	320 x 200
MCGAC2	2	320 x 200
MCGAC3	3	320 x 200
MCGAMed	4	640 x 200
MCGAHi	5	640 x 480
ATT400C0	0	320 x 200
ATT400C1	1	320 x 200
ATT400C2	2	320 x 200
ATT400C3	3	320 x 200
ATT400Med	4	640 x 200
ATT400Hi	5	640 x 400
EGAMonoHi	3	640 x 350
HercMonoHi	0	720 x 348
IBM8514Lo	0	640 x 480
IBM8514Hi	1	1024 x 768
VGALo	0	640 x 200
VGAMed	1	640 x 350
VGAHi	2	640 x 480
PC3270Hi	0	720 x 350

Vízszintes szövegigazítás:

LeftText	0	Balra igazított.
CenterText	1	Középre igazított.
RightText	2	Jobbra igazított.

Függőleges szövegigazítás:

BottomText	0	Alulra igazítás.
CenterText	1	Középre igazítás.
TopText	2	Föülre igazítás.

Vonal típusai és vastagságai:

SolidLn	0	Normál vonal.
DottedLn	1	Szaggatott vonal.
CenterLn	2	Pontvonal.
DashedLn	3	Hosszított szaggatott vonal.
UserBitLn	4	Felhasználó által definiált.
NormWidth	1	Normál vastagságú vonal.
ThickWidth	3	Vastag vonal.

Betűtípusok, szövegállások és nagyságok:

DefaultFont	0	
TriplexFont	1	
SmallFont	2	
SansSerifFont	3	
GothicFont	4	
HorizDir	0	Vízszintes írás.
VertDir	1	Függőleges írás.
UserCharSize	0	A felhasználó által létrehozott betűtípus mérete.

GRAFIKUS HIBAKÓDOK

grOk	0	A grafikus ablak megnyitásakor nem történt hiba.
grNoInitGraph	1	A szükséges BGI állomány nincs meg.
grNotDetected	2	A megadott grafikus kártya nincs a gépben.
grFileNotFound	3	A meghajtóállomány nincs meg.
grInvalidDriver	4	Érvénytelen meghajtóállomány.
grNoLoadMem	5	A meghajtó betöltéséhez nincs elég memória.
grNoScanMem	6	Rajzolás közben elfogyott a memória.
grNoFloodMem	7	Színezés közben elfogyott a memória.
grFontNotFound	8	A megadott betűkészlet nincs meg.
grNoFontMem	9	A megadott betűkészlet betöltéséhez nincs elég memória.
grInvalidMode	10	A megadott üzemmód nem felel meg a kiválasztott meghajtónak.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

grError	11
Grafikus hiba.	
grIOerror	12
Grafikus I/O hiba.	
grInvalidFont	13
Érvénytelen betűkészletnév.	
grInvalidFontNum	14
Érvénytelen betűkészletszám.	

VÁLTOZÓK

A grafikus memória mérete, illetve az ebből még szabadon lévő byte-ok száma:

GraphGetMemPtr : Pointer

A teljes grafikus memória mérete byte-ban.

GraphFreeMemPtr : Pointer

A szabad grafikus memória mérete byte-ban.

FÜGGVÉNYEK ÉS ELJÁRÁSOK

ARC : eljárás

Kör vagy körív rajzolása az aktuális grafikus üzemmódban.

ARC (*X, Y: Integer, STANGLE, ENDANGLE, RADIUS*
: *Word*)

X

A kör vagy körív középpontjának vízszintes koordinátája (INTEGER típusú).

Y

A kör vagy körív középpontjának függőleges koordinátája (INTEGER típusú).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

STANGLE

A kör vagy körív kezdő szöge fokban (WORD típusú).

ENDANGLE

A kör vagy körív végszöge fokban (WORD típusú).

RADIUS

A kör vagy körív sugara (WORD típusú).

BAR : eljárás

Kitöltött téglalap rajzolása grafikus üzemmódban az aktuális kitöltési színnel.

BAR (X1, Y1, X2, Y2 : Integer)

X1, Y1

A téglalap egyik csúcsának koordinátái (WORD típusúak).

X2, Y2

A téglalap másik (átellenes) csúcsának koordinátái (WORD típusúak).

BAR3D : eljárás

Kitöltött háromdimenziós hatású téglalest rajzolása az aktuális színnel.

BAR3D (x1, y1, x2, y2, DEPTH : Word, TOP
: Boolean)

X1, Y1

A terület egyik csúcsának koordinátái (WORD típusúak).

X2, Y2

A terület másik (átellenes) csúcsának koordinátái (WORD típusúak).

DEPTH

A háromdimenziós hatás mélysége (WORD típusú).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

TOP

Azt határozza meg, hogy a három dimenziós hatás kettő vagy egy oldalon nyilvánuljon-e meg. Ha kettő, akkor IGAZ (TRUE) értéke, egyébként pedig HAMIS (FALSE) (BOOLEAN típusú).

CIRCLE : eljárás

Kör rajzolása.

CIRCLE (*X, Y: Integer, RADIUS: Word*)

X

A kör középpontjának vízszintes koordinátája (INTEGER típusú).

Y

A középpont függőleges koordinátája (INTEGER típusú).

RADIUS

A kör sugara (WORD típusú).

CLEARDEVICE : eljárás

Törli az aktuális grafikus kimeneti eszközt, és alap helyzetbe állítja annak mutatóját.

CLEARDEVICE

CLEARVIEWPORT : eljárás

Törli a látható grafikus területet.

CLEARVIEWPORT

CLOSEGRAPH : eljárás

A grafikus üzemmód kikapcsolása, visszatérés karakteres üzemmódba.

CLOSEGRAPH

DETECTGRAPH : eljárás

A megadott változóba írja a számítógépben levő grafikus kártya típusát és a javasolt grafikus üzemmód számát.

DETECTGRAPH (Var GRAPHDRIVER, GRAPHMODE
:Integer)

GRAPHDRIVER

A grafikus kártya típusa (INTEGER típusú).

GRAPHMODE

A javasolt grafikus üzemmód (INTEGER típusú).

DRAWPOLY : eljárás

Sokszög rajzolása.

DRAWPOLY (NUMPOINTS: Word, Var
POLYPOINTS: Pointtype)

NUMPOINTS

A rajzolandó sokszög csúcsainak száma (WORD típusú).

POLYPOINTS

A pontok koordinátái (POINTTYPE típusú).

ELLIPSE : eljárás

Ellipszis vagy ellipsziszív rajzolása.

ELLIPSE (X,Y: Integer, STANGLE, ENDANGLE, XRADIUS,
YRADIUS: Word)

X

Az ellipszis középpontjának vízszintes koordinátája (INTEGER típusú).

Y

Az ellipszis középpontjának függőleges koordinátája (INTEGER típusú).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

STANGLE

Az ellipszis ívének kezdő szöge fokban (WORD típusú).

ENDANGLE

Az ellipszis ívének befejező szöge fokban (WORD típusú).

XRADIUS

Az ellipszis vízszintes sugara (WORD típusú).

YRADIUS

Az ellipszis függőleges sugara (WORD típusú).

FILLELLIPSE : eljárás

Kitöltött ellipszis rajzolása az aktuális színnel.

FILLELLIPSE (*X, Y:Integer, XRADIUS, YRADIUS: Word*)

X

Az ellipszis középpontjának vízszintes koordinátája (INTEGER típusú).

Y

Az ellipszis középpontjának függőleges koordinátája (INTEGER típusú).

XRADIUS

Az ellipszis vízszintes sugara (WORD típusú).

YRADIUS

Az ellipszis függőleges sugara (WORD típusú).

FILLPOLY : eljárás

Kitöltött sokszög rajzolása.

FILLPOLY (*NUMPOINTS: Word, Var POLYPOINTS: PointType*)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

NUMPOINTS

A rajzolandó sokszög csúcsainak száma (WORD típusú).

POLYPOINTS

A pontok koordinátái (POINTTYPE típusú).

FLOODFILL : eljárás

Képernyőterület kitöltése (beszínezése).

FLOODFILL (*X, Y: Word, BORDER: Byte*)

X

A színezésre szánt terület belső pontjának vízszintes koordinátája (WORD típusú).

Y

A színezésre szánt terület belső pontjának függőleges koordinátája (WORD típusú).

BORDER

A megadott keretszín (BYTE típusú).

GETARCCOORDS: eljárás

Az utolsó kör vagy ellipszis rajzolásakor megadott paramétereket adja vissza.

GETARCCOORDS (*ARCCOORDS: ArcCoordsType*)

ARCCOORDS

Az utolsó kör- vagy ellipszisrajzoló utasítás paraméterei (ArcCoordsType típusú, amely

Record

X, Y,

Xstart, Ystart,

Xend, Yend : integer;

end;

módon képezhető).

GETASPECTRATIO : eljárás

Az aktuális grafikus üzemmód torzítási értékét adja vissza.

GETASPECTRATIO (*XASP*, *YASP*: *Word*)

XASP

A vízszintes torzítás értéke (*WORD* típusú).

YASP

A függőleges torzítás értéke (*WORD* típusú).

GETBKCOLOR : függvény

Az aktuális háttérszínt adja vissza.

GETBKCOLOR : *WORD*

GETCOLOR : függvény

Az aktuális rajzolási színt adja vissza.

GETCOLOR : *WORD*

GETDEFAULTPALETTE : függvény

Visszaadja a grafikus rendszer indításakor érvényes palettabeállítást.

GETDEFAULTPALETTE (Var *PALETTE*:
PaletteType)

PALETTE

A paletta változója (*PALETTE* típusú).

GETDRIVERNAME : függvény

Az aktuális grafikus képernyő típusát adja vissza karakterláncként.

GETDRIVERNAME : *String*

GETFILLPATTERN : eljárás

Az aktuális színezési mintát adja vissza.

GETFILLPATTERN (Var FILLPATTERN:
FillpatternType)

FILLPATTERN

Az adatok tárolására szolgáló változó (FILLPATTERN-
TYPE típusú).

GETFILLSETTINGS : eljárás

Az aktuális kitöltései mintát és szintet adja vissza a
megadott változóban.

GETFILLSETTINGS (Var FILLINFO:
FillsettingsType)

FILLINFO

A kitöltési információk változója (FILLSETTINGSType
típusú).

GETGRAPHMODE : függvény

Az aktuális grafikus mód sorszámával tér vissza.

GETGRAPHMODE : INTEGER

GETIMAGE : eljárás

A megadott grafikus terület változóba mentése
bittérképként.

GETIMAGE (X1,Y1,X2,Y2: Word ,Var BITMAP:
Pointer)

X1,Y1

A téglalap egyik csúcsának koordinátái (WORD típusú-
ak).

X2,Y2

A téglalap átellenes csúcsának koordinátái (WORD típu-
súak).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

BITMAP

A bittérkép memóriacíme (POINTER típusú).

GETLINESETTINGS : eljárás

Az aktuális vonaljellemzők lekérdezése.

GETLINESETTINGS (Var *LINEINFO*:
LineSettingsType)

LINEINFO

A vonaljellemzők változója (LINESETTINGSTYPE típusú).

GETMAXCOLOR : függvény

Visszaadja az aktuális grafikus üzemmódban egyszerűen használható színek számát.

GETMAXCOLOR : Word

GETMAXMODE : függvény

Megadja a grafikus kártyát legjobban kihasználó üzemmód sorszámát.

GETMAXMODE : Integer

GETMAXX : függvény

Az aktuális grafikus üzemmód (még látható) vízszintes koordinátájának legnagyobb értékével tér vissza.

GETMAXX : Integer

GETMAXY : függvény

Az aktuális grafikus üzemmód (még látható) függőleges koordinátájának legnagyobb értékével tér vissza.

GETMAXY : Integer

GETMODENAME : függvény

A számmal megadott grafikus mód nevével tér vissza.

```
GETMODENAME (MODENUMBER: Integer):
              STRING
```

MODENUMBER

A grafikus mód sorszáma (INTEGER típusú).

GETMODERANGE : eljárás

A grafikus meghajtót legkevésbé és leginkább kihasználó üzemmódok sorszámát adja vissza

```
GETMODERANGE (GRAPHDRIVER: Integer, Var
              LOMODE , HIMODE: Integer)
```

GRAPHDRIVER

Az grafikus meghajtó száma (INTEGER típusú).

LOMODE

A grafikus meghajtót legkevésbé kihasználó üzemmód számának változója (INTEGER típusú).

HIMODE

A grafikus meghajtót legjobban kihasználó üzemmód változója (INTEGER típusú).

GETPALETTE : eljárás

Az aktuális paletta paramétereinek lekérdezése.

```
GETPALETTE (Var PALETTE: PaletteType)
```

PALETTE

A palettainformációk tárolására szolgáló változó (PALETTE TYPE típusú).

GETPALETTE SIZE : függvény

Az aktuális palettában levő színek számát adja vissza.

```
GETPALETTE SIZE : Integer
```

GETPIXEL : függvény

A megadott képernyőpont színének lekérdezése.

GETPIXEL (*X,Y: Integer*) : Word

X,Y

A pont koordinátái (INTEGER típusúak).

GETTEXTSETTINGS : eljárás

A szövegjellemzők lekérdezése.

GETTEXTSETTINGS (Var *TEXTINFO: TextSettingsType*)

TEXTINFO

A szövegjellemzők változója (TEXTSETTINGSTYPE típusú, amely

RECORD

FONT : WORD;

DIRECTION : WORD;

CHARSIZE : WORD;

HORIZ : WORD;

VERIZ : WORD;

End;

formában képezhető).

GETVIEWSETTINGS : eljárás

Az aktuális grafikus ablak paramétereit adja vissza a megadott változóban.

GETVIEWSETTINGS (Var *VIEWPORT: ViewPortType*)

VIEWPORT

Az ablakjellemzők változója (VIEWPORTTYPE típusú, amely

RECORD

x1,y1,x2,y2 : Integer; {a grafikus képernyő mérete}

CLIP : Boolean; {az ablakon kívülre is írhatunk}

End;

módon képezhető).

GETX : függvény

Az aktuális képernyőpont vízszintes irányú koordinátájának lekérdezése.

GETX : Integer

GETY : függvény

Az aktuális képernyőpont függőleges irányú koordinátájának lekérdezése.

GETY : Integer

GRAPHDEFAULTS : eljárás

Alaphelyzetbe állítja a grafikus képernyőt, és a kurzort a bal felső sarokba helyezi.

GRAPHDEFAULTS

GRAPHERRORMSG : függvény

A számmal megadott grafikus hiba szövegének lekérdezése.

GRAPHERRORMSG (ERRORCODE: Integer) :
String

ERRORCODE

A hibakód (INTEGER típusú).

GRAPHRESULT : függvény

Visszaadja az utoljára végrehajtott grafikus művelet utáni hibakódot.

GRAPHRESULT : Integer

IMAGESIZE : függvény

A megadott téglalaphoz megfelelő grafikus terület tárolásához szükséges memória méretét adja vissza.

IMAGESIZE (X1, Y1, X2, Y2: Integer) : WORD

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

X1, Y1

A terület egyik csúcsának koordinátái (INTEGER típusúak).

X2, Y2

A terület átellesk csúcsának koordinátái (INTEGER típusúak).

INITGRAPH : eljárás

A grafikus üzemmód beállítása, bekapcsolása.

INITGRAPH (Var GRAPHDRIVER, GRAPHMODE: Integer, PATHTODRIVER: String)

GRAPHDRIVER

A grafikus meghajtó sorszáma (INTEGER típusú).

GRAPMODE

A grafikus üzemmód sorszáma (INTEGER típusú).

GRAPHTODRIVER

A grafikus meghajtó elérési útja (STRING típusú).

INSTALLUSERDRIVER : függvény

Új grafikus meghajtó létrehozása. A függvény visszatérési értéke az új meghajtó száma.

INSTALLUSERDRIVER (NAME: String, AUTODETECTPTR: Pointer) : INTEGER

NAME

Az új grafikus meghajtó neve (STRING típusú).

AUTODETECTPTR

Az új meghajtó kezdőcíme (POINTER típusú).

INSTALLUSERFONT : függvény

Új betűkészlet létrehozása. A függvény visszatérési értéke az új betűkészlet száma.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

INSTALLUSERFONT (FONTFILENAME: String) :
INTEGER

FONTFILENAME

Az új betűkészletet tartalmazó (fizikai) állomány neve (STRING típusú).

LINE : eljárás

Szakasz rajzolása (az aktuális színnel és vonaltípussal).

LINE (X1, Y1, X2, Y2: Word)

X1, Y1

A szakasz egyik végpontjának koordinátái (WORD típusúak).

X2, Y2

A szakasz másik végpontjának koordinátái (WORD típusúak).

LINEREL : eljárás

Szakasz rajzolása az aktuális képernyőponttól relatív koordinátákkal megadott pontig, az aktuális vonaltípussal.

LINEREL (DX, DY: Word)

DX

A vízszintes koordinátapontok lépésszámát adja meg (WORD típusú).

DY

A függőleges koordinátapontok lépésszámát adja meg (WORD típusú).

LINETO : eljárás

Az aktuális képernyőponttól egyenest rajzol a megadott koordinátájú pontig az aktuális színnel.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

`LINETO (X, Y: Word)`

X, Y

A vízszintes és függőleges koordináták (WORD típusúak).

MOVEREL : eljárás

Az aktuális képernyőpont elmozgatása a megadott relatív vízszintes és függőleges koordinátáknak megfelelően.

`MOVEREL (DX, DY: Word)`

DX, DY

Az elmozgatás relatív távolságainak vízszintes és függőleges mértéke (WORD típusúak).

MOVETO : eljárás

Az aktuális képernyőpont elmozgatása a megadott koordinátákra.

`MOVETO (X, Y: Word)`

X, Y

A mozgató célkoordinátái (WORD típusúak).

OUTTEXT : eljárás

A megadott karakterlánc kiírása grafikus képernyőre az aktuális helytől kezdődően, az aktuális szövegtípussal.

`OUTTEXT (TextString: String)`

TextString

A kiírandó szöveg (STRING típusú).

OUTTEXTXY : eljárás

Karakterlánc kiírása a grafikus képernyő meghatározott pontjától (az aktuális szövegtípussal).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

OUTTEXTXY (X, Y: Word, TEXTSTRING: String)

X, Y

A kiírandó karakterlánc kezdő koordinátái (WORD típusúak).

TEXTSTRING

A kiírandó szöveg (STRING típusú).

PIESLICE : eljárás

Körcikk rajzolása az aktuális színnel.

PIESLICE (X, Y: Integer, STANGLE, ENDANGLE, RADIUS: Word)

X, Y

A kör középpontjának koordinátái (INTEGER típusúak).

STANGLE

A körcikk ívének kezdő szöge fokban (WORD típusú).

ENDANGLE

A körcikk ívének befejező szöge fokban (WORD típusú).

RADIUS

A kör sugara (WORD típusú).

PUTIMAGE : eljárás

Az megadott koordinátájú ponttól kezdve megjeleníti a GETIMAGE által elmentett BIT-térképet.

PUTIMAGE (X, Y: Integer, Var BITMAP: Pointer, BITBLT: Word)

X, Y

A megjelenítendő ábra bal felső sarkának koordinátái (INTEGER típusúak).

BITMAP

Az elmentett bittérkép memóriacíme (POINTER típusú).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

BITBLD

A rajzolási mód (WORD típusú).

PUTPIXEL : eljárás

Képernyőpont kirajzolása az aktuális színnel.

PUTPIXEL (X,Y: Integer, PIXEL: Word)

X, Y

A képpont koordinátái (INTEGER típusúak).

PIXEL

A pont színe (WORD típusú).

RECTANGLE : eljárás

Téglalap rajzolása az aktuális vonaltípussal és színnel bármely két átlellenes csúcsból.

RECTANGLE (X1,Y1,X2,Y2: Word)

X1, Y1

A téglalap egyik csúcsának koordinátái (WORD típusúak).

X2, Y2

A téglalap átlellenes csúcsának koordinátái (WORD típusúak).

REGISTERBGIDRIVER : függvény

Regisztrálja a memóriában levő felhasználói vagy gyári grafikus meghajtót, és visszaadja a neki megfelelő sorszámot.

REGISTERBGIDRIVER (DRIVER: Pointer):Integer

DRIVER

A betöltött grafikus meghajtó kezdőcíme (POINTER típusú).

REGISTERBGIFONT : függvény

Regisztrálja a megadott, memóriában levő felhasználói vagy gyári betűkészletet, és visszaadja a neki megfeleltetett sorszámot.

REGISTERBGIFONT (FONT: Pointer): INTEGER

FONT

A betöltött betűtípus kezdőcíme (POINTER típusú).

RESTORECRTMODE : eljárás

Grafikus üzemmódból szöveges üzemmódba lép.

RESTORECRTMODE

SECTOR : eljárás

Kitöltött ellipszisszelet rajzolása az aktuális kitöltési színnel.

SECTOR (X, Y, STANGLE, ENDANGLE, XRADIUS, YRADIUS: Word)

X, Y

Az ellipszis középpontjának koordinátái (WORD típusúak).

STANGLE

Az ív kezdő szöge fokban megadva (WORD típusú).

ENDANGLE

Az ív végének szöge fokban (WORD típusú).

XRADIUS

Az ellipszis vízszintes sugara (WORD típusú).

YRADIUS

Az ellipszis függőleges sugara (WORD típusú).

SETACTIVEPAGE : eljárás

Az aktív képernyőlap beállítása.

SETACTIVEPAGE (PAGE: Word)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

PAGE

A grafikus képernyőlap száma (WORD típusú).

SETALLPALETTE : eljárás

A palettajellemzők beállítása.

SETALLPALETTE (PALETTE: PaletteType)

PALETTE

A paletta információi (PALLETTETYPE típusú, amely

RECORD

Size : Byte; {az ablak mérete}

Colors: Array [0..MaxColors] Of ShortInt; {színek száma}

End;

módon képezhető).

SETASPECTRATIO : eljárás

Az aktív képernyőlap torzítási mértékének beállítása a két torzítási komponens hányadosaként.

SETASPECTRATIO (XASP, YASP: Word) :WORD

XASP

Az x irányú komponens (WORD típusú).

YASP

Az y irányú komponens (WORD típusú).

SETBKCOLOR : eljárás

Az aktuális háttérszín beállítása.

SETBKCOLOR (COLORNUM: Word)

COLORNUM

A háttér színe (WORD típusú).

SETCOLOR : eljárás

Az aktuális kiírási és rajzolósi szín beállítása.

SETCOLOR (COLOR: Word)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

COLOR

A kívánt szín száma (WORD típusú).

SETFILLPATTERN : eljárás

Felhasználó által definiált kitöltési minta létrehozása.

SETFILLPATTERN (PATTERN: FillPatternType,
COLOR: Word)

PATTERN

A kitöltési minta (FILLPATTERNTYPE típusú)

COLOR

A minta színe (WORD típusú).

SETFILLSTYLE : eljárás

Az kitöltési szín és minta beállítása.

SETFILLSTYLE (PATTERN, COLOR: Word)

PATTERN

A kitöltési minta sorszám (WORD típusú).

COLOR

A kitöltés színe (WORD típusú).

SETGRAPHBUFSIZE : eljárás

A színezéshez szükséges memória méretének beállítása (az alapértelmezés: 4kb). Az eljárás használatának csak az INITGRAPH hívása előtt van értelme.

SETGRAPHBUFSIZE (BUFSIZE: Word)

BUFSIZE

A puffer mérete (WORD típusú).

SETGRAPHMODE : eljárás

A grafikus üzemmód beállítása, illetve törlése.

SETGRAPHMODE (MODE: Integer)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

MODE

A grafikus üzemmód száma (INTEGER típusú).

SETLINESTYLE : eljárás

A vonaljellemzők beállítása.

SETLINESTYLE (LINESTYLE: Word, PATTERN: Byte,
THICKNESS: Word)

LINESTYLE

A vonaltípus száma (WORD típusú).

PATTERN

Ha a vonaltípus USERBIN, akkor az itt megadott Byte típusú paraméter bitjei rajzolódnak ki, egyébként figyelmen kívül marad.

THICKNESS

A vonal vastagsága (WORD típusú).

SETPALETTE : eljárás

A palettában levő színt a megadott színre cseréli.

SETPALETTE (COLORNUM: Word, COLOR: ShortInt)

COLORNUM

A paletta színe (WORD típusú).

COLOR

A kívánt szín (SHORTINT típusú).

SETRGBPALETTE : eljárás

Grafikus képernyő (VGA vagy IBM8514) esetén a paletta színének beállítása a három színkomponens alapján. Minden színösszetevőt az érték alsó byte-jának felső 6 bitjéből számol (4 - 256).

SETRGBPALETTE (COLORNUM, REDVALUE,
GREENVALUE, BLUEVALUE: Integer)

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

COLORNUM

A beállítandó szín sorszáma (INTEGER típusú).

REDVALUE

A piros összetevő (INTEGER típusú).

GREENVALUE

A zöld összetevő (INTEGER típusú).

BLUEVALUE

A kék összetevő (INTEGER típusú).

SETTEXTJUSTIFY : eljárás

Szövegigazítási jellemzők beállítása.

SETTEXTJUSTIFY (*HORIZ, VERT: Word*)

HORIZ

Vízszintes kiírási mód (WORD típusú).

VERT

Függőleges kiírási mód (WORD típusú).

SETTEXTSTYLE : eljárás

Beállítja a kiírandó szöveg formáját.

SETTEXTSTYLE (*FONT, DIRECTION, CHARSIZE
:Word*)

FONT

A kiírandó szöveg stílusa, formája (WORD típusú).

DIRECTION

A szövegkiírás módja (pl. vízszintesen vagy függőlegesen) (WORD típusú).

CHARSIZE

A kiírandó szöveg magassága, mérete (WORD típusú).

SETUSERCHARSIZE : eljárás

Megváltoztatja a felhasználó által létrehozott karakterkészlet betűinek jellemzőit.

SETUSERCHARSIZE (MULTX, DIVX, MULTY, DIVY :
Word)

MULTX, DIVX

A karakter vízszintes mértetének beállítása a MULTX / DIVX hányadossal (WORD típusúak).

MULTY, DIVY

A karakter függőleges mértetének beállítása a MULTY / DIVY hányadossal (WORD típusú).

SETVIEWPORT : eljárás

Az aktív grafikus képernyőablak beállítása.

SETVIEWPORT (X1, Y1, X2, Y2 : Word, CLIP
: Boolean)

X1, Y1

Az ablak bal felső sarkának koordinátái (WORD típusúak).

X2, Y2

Az ablak jobb alsó sarkának koordinátái (WORD típusúak).

CLIP

Ha a grafikus ablakon kívül is szeretnénk rajzolni, akkor a CLIP értékének igaznak (TRUE) kell lennie, egyébként hamisnak (FALSE) (BOOLEAN típusú).

SETVISUALPAGE : eljárás

Az látható képernyőlap beállítása.

SETVISUALPAGE (PAGE : Word)

PAGE

A látható képernyőlap sorszáma (WORD típusú).

SETWRITEMODE : eljárás

Az egyenesek rajzolósi módjának beállítása.

SETWRITEMODE (*WRITEMODE: Integer*)

WRITEMODE

Rajzolósi mód (INTEGER típusú).

TEXTHEIGHT : függvény

A megadott karakterlánc grafikus képernyőn elfoglalt magasságát adja meg képernyőpontokban (WORD típusú).

TEXTHEIGHT (*TEXTSTRING: String*): WORD

TEXTSTRING

A vizsgálandó karakterlánc (STRING típusú).

TEXTWIDTH : függvény

Adott karakterlánc grafikus képernyőn elfoglalt szélességét adja meg képernyőpontokban (WORD típusú).

TEXTWIDTH (*TEXTSTRING: String*): Word

TEXTSTRING

A vizsgálandó karakterlánc (STRING típusú).

OVERLAY UNIT

TÍPUSOK

OvrReadFunc = Function(OvrSeg: WORD) :
INTEGER;

KONSTANSOK

ovrOK = 0;

Nem történt hiba.

ovrError = -1;

Overlay-kezelési hiba.

ovrNotFound = -2;

Az overlay-állomány nincs meg.

ovrNoMemory = -3;

Az overlay-puffer számára nincs elég memória.

ovrIOError = -4;

Beviteli/kiviteli hiba az overlay-állomány kezelése során.

ovrNoEmsDriver = -5;

Az EMS meghajtó nincs a memóriában.

ovrNoEmsMemory = -6;

Nincs elég EMS memória.

VÁLTOZÓK

OvrResult :INTEGER;

OvrFileMode :BYTE;

OvrLoadCount :WORD;

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

OvrReadBuf :OvrReadFunc;
OvrTrapCount :WORD;
OvrEMSPages :WORD;

FÜGGVÉNYEK ÉS ELJÁRÁSOK

OVRCLEARBUF : eljárás

Az overlay-puffer törlése. A további overlay-eljárásokat lemezről vagy az EMS-ből kell betölteni. Ha overlay-állományban használjuk, a törlés után az állomány újra betöltődik.

OVRCLEARBUF

OVRGETBUF : függvény

Az aktuális overlay-puffer méretét adja vissza (byte-ban).

OVRGETBUF : LONGINT

OVRGETRETRY : függvény

A feltételes overlay-puffer (próbálkozási terület) méretét adja vissza byte-ban.

OVRGETRETRY : LONGINT

OVRINIT : eljárás

Alaphelyzetbe állítja az overlay-kezelőt, és megnyitja a megadott overlay-egységeket tartalmazó fizikai állományt. Az eljárást minden program elején meg kell hívni, még a munkaterület-(heap)-foglalások előtt.

OVRINIT (*FILENAME*)

FILENAME

Az overlay-állomány neve (STRING típusú).

OVRINITEMS : eljárás

Ha az EMS meghajtó működik, az overlay-állomány az EMS-be töltődik. Így a lemezolvasásokat memóriamásolások fogják helyettesíteni. A program befejeztével a lefoglalt memória automatikusan felszabadul.

OVRINITEMS

OVRSETBUF : eljárás

Beállítja az overlay-puffer méretét. Az eljárás hívásakor a munkaterületnek (heap) üresnek kell lennie.

OVRSETBUF (*SIZE*)

SIZE

A puffer mérete byte-ban (LONGINT típusú).

OVRSETRETRY : eljárás

A feltételes overlay-puffer (próbálkozási terület) méretét állítja be. Tesztelésük során az overlay-egységek a feltételes területre kerülnek. Ilyenkor, ha a hívott eljárás az egységben van, az egység a puffer elejére kerül.

OVRSETRETRY (*SIZE*)

SIZE

A puffer mérete byte-ban (LONGINT típusú). Alapértelmezése 0.

PRINT UNIT

VÁLTOZÓK

Lst : TEXT;

Az eljárássegég automatikusan hozzárendeli az első nyomtatót az Lst állományazonosítóhoz, amelynek segítségével a PRINT eljárássegéget használó programból állományírási művelettel lehet nyomtatni.

SYSTEM UNIT**KONSTANSOK**

MaxInt=32767;
MaxLongInt=2147483647;

VÁLTOZÓK**Szabványos I/O:**

Input : TEXT;
Output : TEXT;

I/O hibakezelés:

InOutRes : INTEGER=0;

Állománykezelés:

FileMode : BYTE=2;

Programbefejezés:

ExitCode : INTEGER=0;
ErrorAddr : POINTER=NIL;
ExitProc : POINTER=NIL;

Matematikai társprocesszor:

Test8087 : BYTE=0;

Véletlenszám:

RandSeed : LONGINT=0;

Memórlacím:

PrefixSeg : WORD=0;
StackLimit : WORD=0;

Heap-kezelés:

HeapOrg : POINTER=NIL;
HeapEnd : POINTER=NIL;
HeapPtr : POINTER=NIL;
FreeList : POINTER=NIL;
HeapError : POINTER=NIL;

Overlay-kezelés:

OvrHeapSize : WORD=0;
OvrDebugPtr : POINTER=NIL;
OvrLoadList : WORD=0;
OvrCodeList : WORD=0;
OvrHeapOrg : WORD=0;
OvrHeapPtr : WORD=0;
OvrHeapEnd : WORD=0;
OvrDosHandle : WORD=0;
OvrEMSHandle : WORD=0;

Elmentett megszakítási vektorok:

SaveInt00 : POINTER;
SaveInt02 : POINTER;
SaveInt1B : POINTER;
SaveInt21 : POINTER;
SaveInt23 : POINTER;
SaveInt24 : POINTER;
SaveInt34 : POINTER;
SaveInt35 : POINTER;
SaveInt36 : POINTER;
SaveInt37 : POINTER;
SaveInt38 : POINTER;
SaveInt39 : POINTER;
SaveInt3A : POINTER;
SaveInt3B : POINTER;
SaveInt3C : POINTER;
SaveInt3D : POINTER;
SaveInt3E : POINTER;
SaveInt3F : POINTER;
SaveInt75 : POINTER;

FÜGGVÉNYEK ÉS ELJÁRÁSOK**ABS : függvény**

Visszaadja a megadott változó vagy valós kifejezés abszolút értékét.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

ABS (X)

X
Tetszőleges numerikus kifejezés (egész vagy valós típusú lehet).

ADDR : függvény

Visszaadja a megadott változó, típusos állandó, eljárás, függvény, illetve objektum memóriacímét. Eljárás típusú változó esetén e függvényt kétszer kell alkalmazni. Az ADDR függvény a @ operátorral helyettesíthető.

ADDR (X) : POINTER

X
Változó, típusos állandó, eljárás, függvény stb. azonosítója.

APPEND : eljárás

Megnyitja írásra az *F* szöveges állományt. Az állományba írt adatok az állomány eredeti tartalmának végéhez fűződnek. Ha az állomány már meg van nyitva, akkor az APPEND először lezárja, majd újra megnyitja. Az *F* szöveges állományt először az ASSIGN eljárással *létező* fizikai állományhoz kell rendelni. Ha *F*-hez ASSIGN-nal hozzárendelt név üres karakterlánc, az írás a szabványos CON perifériára történik.

APPEND (F)

F
Logikai állományazonosító (TEXT típusú).

ARCTAN : függvény

Valós kifejezés arkusz tangensét adja vissza radiánban.

ARCTAN (X) : REAL

X
A valós kifejezés (REAL típusú).

ASSIGN : eljárás

Logikai állományazonosítóhoz fizikai állománynevet rendel. Ezek után a program végéig, illetve újabb ASSIGN eljárásig minden olyan művelet, amelyet a program a logikai állományon elvégez, a *String* fizikai állományra vonatkozik. A *String* teljes útvonalleírást (állományspecifikációt) és periférianévet is tartalmazhat. Ha *String* üres karakterlánc, az *F* a CON perifériához lesz rendelve. Csak lezárt állományra használható.

ASSIGN (*F*, *String*)

F

Logikai állományazonosító (TEXT vagy FILE típusú).

String

Fizikai állománynév (STRING típusú).

ASSIGNED : függvény (új)

Megadja, hogy a paraméterként kapott cím típusú változó mutat-e valahova. Ha igen, akkor értéke igaz (TRUE), ha a cím NIL, hamis (FALSE).

ASSIGNED (*P*) : BOOLEAN

P

Cím típusú változó (POINTER típusú).

BLOCKREAD : eljárás

Típus nélküli állományból adott számú rekord beolvasása, és elhelyezése a memóriában a megadott változó memóriacímétől kezdve. Az alapértelmezés szerinti blokkméret 128 byte. A maximális érték 65535 byte. A tárolás kezdőcíméül megadott változó méretére ügyelni kell, mert ha ez kevés, az utána deklarált változók fölülírodhatnak.

BLOCKREAD (*F*, *BUF*, *COUNT*[, *RESULT*])

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

F

A típus nélküli logikai állomány (FILE típusú).

BUF

Az olvasás célváltozója, bármilyen típusú lehet.

COUNT

A beolvasandó rekordok száma (WORD típusú).

RESULT

A sikeresen beolvasott rekordok száma (WORD típusú).

BLOCKWRITE : eljárás

Adott számú rekord kiírása típus nélküli állományba az adott változó memóriacímétől kezdve. Ha az *F* állomány megnyitásakor nem adtunk meg más értéket, az alapértelmezett blokkméret 128 byte. A maximális érték 65535 byte. Az eljárás csak teljes rekordot ír ki.

`BLOCKWRITE (F, BUF, COUNT, RESULT)`

F

Típus nélküli logikai állomány (FILE típusú).

BUF

Az írás forrásváltozója, bármilyen típusú lehet.

COUNT

A felírandó rekordok száma (WORD típusú).

RESULT

A sikeresen felírt rekordok száma (WORD típusú).

BREAK : eljárás

(új)

Kilép a REPEAT, a WHILE vagy a FOR ciklusból, függetlenül attól, hogy a ciklus végrehajtása hol tart.

BREAK

CHDIR : eljárás

Az aktuális könyvtár beállítása.

CHDIR (S)

S

A beállítandó könyvtár elérési útja (STRING típusú).

CHR : függvény

Az paraméterben megadott ASCII-kódhoz tartozó karaktert adja vissza.

CHR (X) : CHAR

X

A karakterre alakítandó ASCII-kód (BYTE típusú).

CLOSE : eljárás

RESET, REWRITE vagy APPEND eljárással megnyitott állomány lezárása. Az állomány lezárása előtt még lemezre írja a pufferben maradt adatokat.

CLOSE (F)

F

Logikai állomány.

CONCAT : függvény

Kettő vagy több karakterlánc összekapcsolása egyetlen karakterlánccá.

CONCAT (s1[,s2, ..., sn]) : STRING

s1, s2, ..., sn

Az összefűzendő karakterláncok (String típusúak).

CONTINUE : eljárás

(új)

FOR, WHILE vagy REPEAT ciklus következő iterációjának azonnali megkezdése, bárhol tart a program a ciklusmagban.

CONTINUE

COPY : függvény

Részkarakterlánc kivágása adott karakterláncból. Ha a megadott vágási pozíció nagyobb, mint a karakterlánc hossza, az eredmény üres karakterlánc lesz.

COPY (S, INDEX, COUNT) : STRING

S

Az a karakterlánc, amelyből részletet kell kivágni (STRING típusú).

INDEX

A vágás kezdő pozíciója (INTEGER típusú).

COUNT

A kivágandó karakterek száma (INTEGER típusú).

COS : függvény

Koszinuszfüggvény.

COS (X) : REAL

X

Az a valós kifejezés (radiánban), melynek koszinuszát ki kell számolni (REAL típusú).

CSEG : függvény

A CS regiszter tartalmát adja vissza. A CS regiszter tartalma a függvényt hívó eljárás vagy függvény kód-szegmensének szegmenscíme. Így a visszaadott érték eljárássegységenként más és más lehet.

CSEG : WORD

DEC : eljárás

Változó értékének csökkentése a második paraméterben megadott értékkel.

DEC (X [, N])

X

A csökkentendő változó neve (sorszámozott típusú).

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

N

Növekmény (LONGINT típusú).

DELETE : eljárás

Részkarakterlánc törlése karakterláncból adott pozíciótól kezdve. Ha a megadott törlési pozíció nagyobb, mint a teljes karakterlánc hossza, a törlés nem hajtódik végre.

DELETE (*S*, INDEX, COUNT)

S

Az alap karakterláncot tartalmazó változó (STRING típusú), a törlés utáni karakterlánc is ebbe kerül.

INDEX

A törlés kezdő pozíciója (INTEGER típusú).

COUNT

A törlendő karakterek száma (INTEGER típusú).

DISPOSE : eljárás

Felszabadítja a mutatóval megadott munkaterület-részt. Ha a mutató nem a munkaterületbe (heapbe) mutat, futási hibát kapunk. Ha a felszabadítás a munkaterület (heap) tetején történik, akkor a HeapPtr lejjebb csúszik. Egyébként egyszerű szabad blokk keletkezik, melyet a GETMEM, illetve a NEW eljárásokkal használhatunk újra. Amennyiben az adott mutató egy objektum mutatója, paraméterként a destruktorként is megadható. Ebben az esetben előbb végrehajtódik a destruktorként megadott destruktorként, majd felszabadítja a megfelelő munkaterület-részt. Ha az objektumnak van virtuális metódustáblázata, akkor a destruktorként az ebben foglaltak szerint végzi a felszabadítást. Ekkor azonban a méret nem mindig utal az objektum típusára.

DISPOSE (*P* [; *destruktor*]

P

Típusos mutató (POINTER típusú).

destruktor

Az objektum destruktora.

DSEG : függvény

A DS regiszter tartalmával visszatérő függvény. A DS regiszter tartalma azon adatszegmens szegmenscíme, amelyben a program globális konstansai és változói vannak. Így a visszatérési érték minden egységben ugyanaz.

DSEG : WORD

EOF : függvény

Értéke igaz, ha a paraméterben megadott állomány aktuális pozíciója a megadott állomány utolsó komponense mögötti, illetve ha az állomány üres. Típusos és típus nélküli állományokon egyaránt használható.

EOF (F) : BOOLEAN

F

A vizsgált logikai állomány.

EOLN : függvény

Értéke igaz, ha a paraméterben megadott állomány aktuális pozíciója a megadott szöveges állomány sor-, illetve állományvég jelén áll. Állománynév nélkül a szabványos bemenetet vizsgálja.

EOLN [(F)] : BOOLEAN

F

A vizsgált szövegállomány logikai azonosítója (TEXT típusú).

ERASE : eljárás

Törli az adott logikai állományhoz rendelt fizikai állományt. Csak lezárt állományra szabad használni.

ERASE (F)

F

Logikai állomány.

EXIT : eljárás

Az aktuális programszint END sorára ugrik, így hatására a programfutás egy szinttel feljebb lép.

EXIT

EXCLUDE : eljárás

(új)

Adott elem kizárása halmazból.

EXCLUDE (S, I)

S

Az a halmaz, amelyből kizárunk (halmaz típusú).

I

A kizárandó elem. A halmaz elemének kell lennie. (halmaz típusúnak kell lennie).

EXP : függvény

Valós érték exponenciálisát (az e^x számot) adja vissza (ahol x a függvény paramétere).

EXP (X) : REAL

X

A vizsgált érték (REAL típusú).

FAIL : eljárás

Csak konstruktorból hívható eljárás. Hatása részben megegyezik az EXIT eljárással.

Ha a konstruktort megfelelő módon hívjuk meg (pl. a NEW második paramétere), akkor a dinamikus objektum számára maga a konstruktor foglalja le a megfelelő munkaterület-(heap)-részt. Ilyenkor, ha memória-foglalási hiba történik, a program lefagy. (Ezeket a hibákat természetesen a programozó is kezelheti a HeapError mutató állása alapján.) Ha a heap-foglalás továbbra is sikertelen, a FAIL eljárással felszabadítható az objektum.

FAIL

FILEPOS : függvény

A megnyitott állomány állománymutatójával (aktuális komponensének sorszámával) tér vissza. Az állomány szöveges állományon kívül minden más típusú lehet. Az eljárás a READ eljáráshoz hasonlóan eggyel továbblépteti a mutatót.

FILEPOS (*F*) : LONGINT

F

A vizsgált logikai állomány.

FILESIZE : függvény

Az adott állomány méretével (komponenseinek számával) tér vissza. A vizsgált állományt előzőleg meg kell nyitni. Szöveges állományon kívül minden típusra alkalmazható.

FILESIZE (*F*) : LONGINT

F

A vizsgált logikai állomány.

FILLCHAR : eljárás

Adott változó kezdőcímétől kezdve feltölti a memóriát a megadott sorszámozott típusú kifejezés értékével adott számban. Mivel a feltöltés nem ellenőrzött, az eljárás óvatosan kezelendő.

FILLCHAR (*X*, *COUNT*, *VALUE*)

X

Az a változó, melynek kezdőcímétől kezdődik a feltöltés.

COUNT

A feltöltendő terület byte-okban (WORD típusú).

VALUE

A feltöltő érték (0-255-ig).

FLUSH : eljárás

Az átmeneti puffer tartalmát kiírja a megadott, előzőleg megnyitott szöveges állományba.

FLUSH (*F*)

F

A már megnyitott szöveges állomány (TEXT típusú).

FRAC : függvény

Valós kifejezés tizedes részével tér vissza.

FRAC (*X*) : REAL

X

Valós kifejezés (REAL típusú).

FREEMEM : eljárás

Felszabadítja a heapből a mutatóval jelzett adott nagyságú területet. Hibát okoz, ha a mutató nem a heapre mutat.

FREEMEM (*P, SIZE*)

P

A felszabadítás kezdőcíme (POINTER típusú).

SIZE

A felszabadítandó terület mérete (WORD típusú).

GETDIR : eljárás

A megadott meghajtó aktuális könyvtárával tér vissza.

GETDIR (*D, S*)

D

A vizsgált meghajtó sorszáma (A: = 1, B: = 2 stb.) (BYTE típusú).

S

Ebbe a változóba kerül az aktuális könyvtár neve (STRING típusú).

GETMEM : eljárás

Lefoglalja a második paraméterben megadott nagyságú területet a munkaterületen (heap), és kezdőcímét az első paraméterbe írja. Hibát okoz, ha nincs megfelelő nagyságú összefüggő munkaterület. A legnagyobb lefoglalható terület 64 kilobyte.

GETMEM (*P*, *SIZE*)

P

A kezdőcím tárolója (POINTER típusú).

SIZE

A lefoglalandó munkaterület-rész nagysága (WORD típusú).

HALT : eljárás

A program futását megállítja, és a vezérlést visszaadja az operációs rendszernek, illetve a hívó programnak. A paraméter megadása nem kötelező. Ez adja meg a megfelelő ExitCode-ot, illetve ERRORLEVEL értéket.

HALT [(*EXITCODE*)]

EXITCODE

Opcionális kilépési kód (WORD típusú).

HI : függvény

Kétbyte-os kifejezés felső BYTE-ját adja vissza.

HI (*X*) : BYTE

X

A vizsgált kifejezés (INTEGER vagy WORD típusú).

HIGH : függvény

(*új*)

Az adott paraméter típusától függően a típus legjellemzőbb komponensének legnagyobb értékét adja vissza. Így a felsorolt típusnál az értelmezési tartomány legnagyobb elemét, tömb típusnál a tömb utolsó elemének indexét, karakterláncnál a fizikai méretet adja vissza.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

HIGH (X) : az adott típustól függ.

X

Bármilyen típusú változó lehet.

INC : eljárás

A megadott sorszámozott típusú változó értékét növeli a második paraméterben meghatározott értékkel, vagy ennek hiányában eggyel.

INC (X [, N])

X

A növelendő változó (sorszámozott típusú).

N

Növekmény (LONGINT típusú).

INCLUDE : eljárás

(új)

Új elem felvétele a megadott halmazba.

INCLUDE (S, I)

S

A bővítendő halmaz (halmaz típusú).

I

Az új elem (ugyan olyan típusú mint a halmaz).

INSERT : eljárás

Adott karakterláncba a harmadik paramétertől kezdve beszúr egy másik karakterláncot úgy, hogy az eredetileg ott levő karaktereket jobbra tolja.

INSERT (SOURCE, S, INDEX)

SOURCE

A beszúrandó karakterlánc (STRING típusú).

S

Az eredeti karakterlánc (STRING típusú).

INDEX

A beszúrás kezdőpozíciója (INTEGER típusú).

INT : függvény

Valós kifejezés alsó egészrészével tér vissza.

INT (X) : REAL

X

A REAL típusú kifejezés, melynek értékéből alsó egészrészét kell számolni.

IORESULT : függvény

Az utolsó beviteli/kiviteli művelet hibakódját adja vissza (az \$I- direktíva érvényessége mellett) az InOutRes változó alapján. Olvasás után az InOutRes változó nullázódik.

IORESULT : INTEGER

LENGTH : függvény

A paraméterként megadott karakterlánc hosszával tér vissza.

LENGTH (S) : INTEGER

S

A vizsgált karakterlánc (STRING típusú).

LN : függvény

Valós kifejezés logaritmusának kiszámítása.

LN (X) : REAL

X

A vizsgált kifejezés (REAL típusú).

LO : függvény

Kétbyte-os kifejezés alsó byte-ját adja vissza.

LO (X) : BYTE

X

A vizsgált kifejezés (INTEGER vagy WORD típusú).

LOW : függvény (új)

Az adott paraméter típusától függően a típus legjellemzőbb komponensének legkisebb értékét adja vissza. Így a felsorolt típusnál az értelmezési tartomány legkisebb elemét, tömb típusnál a tömb első elemének indexét, karakterlánc, nyitott karakterlánc és tömb esetén pedig egyaránt nullát.

LOW (X) : az adott típustól függ.

X

Bármilyen típusú változó lehet.

MARK : eljárás

Az aktuális munkaterület-(heap-)mutató értékét a paraméterben megadott változóba másolja.

MARK (P)

P

Ebbe a POINTER típusú változóba kerül a munkaterület-(heap-)mutató értéke.

MAXAVAIL : függvény

A legnagyobb összefüggő munkaterület-(heap)-rész méretét adja vissza byte-ban.

MAXAVAIL : LONGINT

MEMAVAIL : függvény

A munkaterületen (heap) levő összes szabad memória nagyságát adja meg byte-ban. A DISPOSE és FREEMEM eljárások használata nélkül MEMAVAIL=MAXAVAIL.

MEMAVAIL : LONGINT

MKDIR : eljárás

Új könyvtár létrehozása.

MKDIR (S)

(Folytatás a következő oldalon)

S
A létrehozandó könyvtár neve, mely teljes elérési utat is tartalmazhat (STRING típusú).

MOVE : eljárás

Adott számú byte másolása változóból változóba.

MOVE (SOURCE, DEST, COUNT)

SOURCE

Forrásváltozó.

DEST

Célváltozó.

COUNT

Az átmásolandó byte-ok száma (WORD típusú).

NEW : eljárás és függvény

A paraméterben megadott típusú dinamikus változó létrehozása a munkaterületen (heap). Ha függvényként hívjuk, akkor a munkaterületen (heap) létrejön az adott típusú dinamikus változó, és a függvény értéke ennek mutatója lesz. Ha a paraméter egy objektum mutatója, a konstruktor is átadható. Ebben az esetben a memóriefoglalást maga a konstruktor végzi, még a konstruktorblokk végrehajtása előtt.

NEW (P [, INIT])

P

A kezdőcímet tároló változó (POINTER típusú).

INIT

Az objektum konstruktora.

ODD : függvény

A paraméterben megadott egész számról eldönti, hogy az páros-e vagy páratlan. Ha páros, a függvény értéke FALSE (hamis), ellenkező esetben TRUE (igaz) lesz.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

ODD (X) : BOOLEAN

X

A vizsgált kifejezés (LONGINT típusú).

OFS : függvény

A paraméter eltolási címét adja vissza.

OFS (X) : WORD

X

Azon függvény vagy eljárás azonosítója, amelynek eltolási címét keressük.

ORD : függvény

A paraméterben megadott sorszámozott típusú értéknek a típus értékészletén belüli sorszámaival tér vissza.

ORD (X) : LONGINT

X

Sorszámozott típusú kifejezés.

PARAMCOUNT : függvény

A program elindításakor a DOS-parancssorban megadott paraméterek számát adja vissza. A paraméterek a PARAMSTR függvénnyel kérdezhetők le.

PARAMCOUNT : WORD

PARAMSTR : függvény

A program elindításakor a DOS-parancssorban megadott paraméterek közül a megadott sorszámút adja vissza.

PARAMSTR (INDEX) : STRING

INDEX

A kívánt paraméter sorszáma (WORD típusú).

PI : függvény

A pi (π) értékével visszatérő függvény. Értéke alapállapotban: 3.1415926536, matematikai társprocesszorral: 3.1415926535897932385.

PI : REAL

POS : függvény

Megadja, hogy az első paraméterként megadott karakterláncot hányadik pozíciótól kezdve tartalmazza a második karakterlánc. Ha nem tartalmazza, a visszaadott érték nulla.

POS (SUBSTR, S) : BYTE

SUBSTR

A keresett karakterlánc (STRING típusú).

S

A keresés karakterlánca (STRING típusú).

PRED : függvény

Sorszámozott típusú értékhez visszaadja a típus értékkészletének azon elemét, amely a megadott értéket megelőzi. Így fontos, hogy a megadott érték sorszámánál eggyel kisebb sorszámhoz tartozzon érték.

PRED (X) : a paramétertől függ

X

Bármilyen sorszámozott típusú változó lehet.

PTR : függvény

A megadott szegmens- és eltolási címet típus nélküli mutatóvá konvertálja.

PTR (SEG, OFS) : POINTER

SEG

Szegmencím (WORD típusú).

OFS

Eltolási cím (WORD típusú).

RANDOM : függvény

Véletlenszám-generátor. Értékhatár nélkül 0 és 1 közötti REAL, értékhatárral 0 és a megadott szám közötti WORD típusú számot képez. Ez a függvény a már definiált RandSeed változó soron következő értékét számítja.

RANDOM [(RANGE)] :REAL vagy EGÉSZ

RANGE

Pozitív egész (EGÉSZ típusú).

RANDOMIZE : eljárás

Az operációs rendszer szerinti véletlen értékkel inicializálja a véletlenszám generátort, beállítja a RandSeed értékét.

RANDOMIZE

READ : eljárás típusos állományokhoz

Komponensek beolvasása típusos állományból rendre a megadott változóba. Fontos, hogy a változók típusa egyezzen az állomány komponensének típusával. Az állománymutató minden olvasáskor automatikusan továbblép.

READ (F, V1[, V2, . . . , Vn])

F

Azon állomány logikai neve, melyből olvasni szeretnénk.

V1 . . . Vn

Azon változók neve, amelyekbe olvasni szeretnénk.

READ : eljárás szöveges állományokhoz

Az eljárás szöveges állomány aktuális pozíciójától olvas be a paraméterek típusától függően értékeket, és azokat a paraméterként megadott változóba helyezi. Ha nincs megadva logikai állománynév, akkor az eljárás a szabványos bemenetről (a billentyűzetről) olvas.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

Ilyenkor a bevitt szöveg a képernyőn is megjelenik, és az első Enterig szerkeszthető. Ebben az esetben azonban előfordulhat, hogy a billentyűzetpufferben maradt karakterek a felhasználó helyett cselekszenek és a program kezelhetetlenné válik. Ilyen esetben a READLN eljárás használatos egyetlen, paraméterként megadott változóval.

```
READ ([ F] V1[ ,V2, ... ,Vn] )
```

F

Logikai állomány (TEXT típusú).

V1 ... Vn

Azon változók neve, melyekbe olvasni szeretnénk. Ezek a változók valós, egész, karakteres és karakterlánc típusúak lehetnek. Karakteres változó esetén egy karaktert; karakterlánc változó esetén egy sort (sorvégjelig), numerikus változó esetén a következő határoló karakterig (Space, Tab, EOL, EOF) olvas.

READLN : eljárás

Ugyanazt teszi, mint a READ eljárás, ám változókba olvasáskor minden olvasás után a következő sor elejére áll a mutató.

```
READLN ([ F] V1 [ ,V2, ... ,Vn] )
```

F

Logikai állomány (TEXT típusú). Ha nincs megadva, a szabványos bemenetről (a billentyűzetről) olvas. Ilyenkor minden olvasás után törli a beviteli puffert.

V1 ... Vn

Azon változók neve, melybe olvasni szeretnénk. Ha nincsenek megadva, nem történik olvasás, csak a mutató áll az olvasás szerinti soron következő sor elejére.

RENAME : eljárás

Állomány átnevezése.

```
RENAME (F,NEWNAME)
```

(Folytatás az következő oldalon)

(Folytatás az előző oldalról)

F

A logikai állománynév (FILE,TEXT,FILE OF típusú).

NEWNAME

A logikai állományhoz tartozó fizikai állomány új neve (STRING típusú).

RESET : eljárás

Létező fizikai állomány megnyitása. Az így megnyitott állományt előzőleg az ASSIGN eljárással logikai állományhoz kell rendelni. Típusos és típus nélküli állományok így megnyitva olvashatók és írhatók, szöveges állományok csak olvashatók. Típus nélküli állományok megnyitásakor a rekordméret is megadható. Az eljárás az állománymutatót lenullázza, tehát az állomány elejére állítja. A nyitott állományt először lezárja, majd újranyitja.

RESET (*F* [,RECSIZE])

F

Az állomány logikai neve (FILE,TEXT,FILE OF típusú).

RECS ZE

A típusos állomány rekordmérete (WORD típusú). Alapbeállítása 128 byte.

REWRITE : eljárás

Fizikai állomány létrehozása és megnyitása elsődlegesen írásra. Az így megnyitott állományt előzőleg az ASSIGN eljárással logikai állományhoz kell rendelni. Típusos és típus nélküli állományok így létrehozva olvashatók és írhatók, szöveges állományok csak írhatók. Típus nélküli állományok készítésekor a rekordméret is megadható. Az eljárás az állománymutatót és az állományméret-mutatót lenullázza. A már létező állományt először törli, majd újra létrehozza.

REWR TE (*F* [,RECSIZE])

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

F

Az állomány logikai neve (FILE, TEXT, FILE OF típusú).

RECS ZE

A típusos állomány rekordmérete (WORD típusú). Alapbeállítás 128 byte.

RMDIR : eljárás

Alkönyvtár törlése. Hibát okoz, ha a könyvtár nem létezik, illetve nem üres.

RMD R (S)

S

A törlendő könyvtár neve (elérési útja) (STRING típusú).

ROUND : függvény

Valós típusú kifejezés egész típusúvá alakítása (kerekítése a legközelebbi egészre).

ROUND (X) : LONG NT

X

A kerekítendő kifejezés (REAL típusú).

RUNERROR : eljárás

Hatására a program futása a megadott futási hibával felfüggesztődik úgy, mintha a hiba tényleg előfordult volna. Ez a hiba a hívó programban a DOSEXIT-CODE függvénnyel, parancsköteggben az ERROR-LEVEL változó használatával kezelhető.

RUNERROR [(ERRORCODE)]

ERRORCODE

A visszaadandó hibakód (BYTE típusú). Ha nincs megadva, a generált hiba kódja 0 (*Runtime error*).

SEEK : eljárás

Típusos vagy típus nélküli állomány megadott sorszámú rekordjára (komponensére) állítja a rekordmutatót. Az állománynak nyitva kell lennie.

SEEK (*F*, *N*)

F

Az állomány logikai neve (FILE, FILE OF, TEXT típusú).

N

A kívánt komponens sorszáma (LONGINT típusú).

SEEKEOF : függvény

Az állománymutatót eggyel tovább lépteti — átugorja a soronkövetkező elválasztó karaktert (TAB, szóköz, sorvégjel), és igaz (TRUE) vagy hamis (FALSE) értéket ad vissza, attól függően, hogy a megadott szöveges állománynak vége van-e vagy sem. Ha nincs megadva állomány, akkor a függvény a szabványos bemenetet (a konzolt, billentyűzetet) vizsgálja.

SEEKEOF [(*F*)] : BOOLEAN

F

Az állomány logikai neve (TEXT típusú).

SEEKEOLN : függvény

Az állománymutatót eggyel tovább lépteti — átugorja a következő elválasztó (TAB vagy szóköz) karaktert, és igaz (TRUE) vagy hamis (FALSE) értéket ad vissza attól függően, hogy a megadott szöveges állomány aktuális sorának vége van-e vagy sem. Ha nincs megadva állomány, akkor a függvény a szabványos bemenetet (a billentyűzetet) vizsgálja.

SEEKEOLN [(*F*)] : BOOLEAN

F

Az állomány logikai neve (TEXT típusú).

SEG : függvény

A megadott objektum szegmenscímét adja vissza.

SEG (X) : WORD

X

Változó vagy eljárás-, illetve függvényazonosító.

SETTEXTBUF : eljárás

A megadott szöveges állományhoz átmeneti puffert kapcsol. Az eljárás a megadott logikai állományra vonatkozó ASSIGN eljárásig hatásos. Billentyűzetről olvasáskor a puffer nagysága adja meg az egyszerre bevihető karakterek számát. Nyitott és zárt állományokra egyaránt használható, ám nyitott állományok esetében a megnyitás után azonnal használni kell.

SETTEXTBUF (F, BUF[, SIZE])

F

Logikai állománynév (TEXT típusú).

BUF

Az átmeneti puffer (CHAR típusú).

SIZE

Az átmeneti puffer mérete (WORD típusú).

SIN : függvény

A megadott paraméter szinuszát adja vissza.

SIN (X) : REAL

X

Kifejezés vagy változó, a szög, amelynek szinuszát ki kell számítani (radiánban; REAL típusú).

SIZEOF : függvény

A paraméterben megadott változó vagy kifejezés memóriafoglalásának kiszámítása (byte-ban).

SIZEOF (P) : INTEGER

(Folytatás a következő oldalon)

P

Tetszőleges típusú kifejezés vagy változó.

SPTR : függvény

Az SP (*stack pointer*) regiszter tartalmát, a veremmutató eltolási címét adja vissza. A verem töltődése közben az SPTR által visszaadott érték csökken. A nulla elérésekor veremtúlcsordulás következik be, melyet a *Stack overflow error* hibaüzenet is jelez.

SPTR : WORD

SQR : függvény

A megadott kifejezés vagy változó négyzetét adja vissza.

SQR (X) : egész vagy valós

X

Kifejezés vagy változó (egész vagy valós).

SQRT : függvény

A megadott kifejezés vagy változó négyzetgyökét adja vissza.

SQRT (X) : egész vagy valós

X

Kifejezés vagy változó (egész vagy valós).

SSEG : függvény

Az SS (*stack segment*) regiszter tartalmát, a verem szegmenscímét adja vissza.

SSEG : WORD

STR : eljárás

Egész vagy valós kifejezés értékének karakterlánccá alakítása. A számokat a karakterláncon belül a megadott pozíción levő tizedesponthoz igazítja, az üres helyeket szóközzel tölti fel.

(Folytatás a következő oldalon)

(Folytatás az előző oldalról)

STR (X [WIDTH [:DECIMALS]] , S)

X

Az átalakítandó egész vagy valós kifejezés.

WIDTH, DECIMALS

Az első paraméter a mezőszélességet, a második a teljes mezőszélességből a tizedes jegyeknek szánt karakterek számát adja meg. A paraméter hiányában a szám egész, illetve lebegőpontos alakban jelenik meg. Ha a tizedesjegyeknek szánt karakterek száma kevés, a függvény a számot a lehetőségekhez mérten kerekíti.

S

Karakterlánc (STRING) típusú változó, amelybe az átalakítással kapott karakterlánc kerül. Ha a célkarakterlánc hossza nem elegendő, a függvény csonkítja az eredményt.

SUCC : függvény

Sorszámozott típusú kifejezés értékéhez visszaadja a megfelelő típus értékészletében soron következő értéket. (Karakter esetén a kódkészlet következő elemét, egész szám esetén az eggyel nagyobb számot stb.) Fontos, hogy az eggyel nagyobb sorszámhoz tartozzon érték.

SUCC (X) : a paramétertől függ

X

Tetszőleges sorszámozott típusú változó.

SWAP : függvény

A megadott kétbyte-os (INTEGER vagy WORD típusú) kifejezés alsó és felső byte-jának fecserélésével kapott értéket adja vissza.

SWAP (X) : a paramétertől függ

X

A kétbyte-os kifejezés (INTEGER vagy WORD típusú).

TRUNC : függvény

Valós kifejezés egész részével tér vissza. Működését tekintve az INT eljárással azonos, a különbség csak a visszatérő érték típusában van.

TRUNC (X) : LONGINT

X

A csonkítandó kifejezés (REAL típusú).

TRUNCATE : eljárás

A megadott típusos vagy típus nélküli állomány csonkítása az aktuális pozíciótól kezdve. Az állománynak nyitva kell lennie.

TRUNCATE (F)

F

Az állomány logikai neve.

TYPEOF : függvény

A paraméterben megadott típushoz vagy változóhoz tartozó virtuális metódustáblázat címét adja meg. Csak virtuális metódust, konstruktort és destruktort tartalmazó objektumra alkalmazható.

TYPEOF (OBJECT) : POINTER

OBJECT

A vizsgált objektum.

UPCASE : függvény

A paraméterben megadott karakter nagybetűs megfelelőjével tér vissza. Ha a megadott karakter nem kisbetű, a függvény hatástalan.

UPCASE (CH) : CHAR

CH

A konvertálandó karakter (CHAR típusú).

VAL : eljárás

Az első paraméterben megadott karakterláncot a második paraméter típusának megfelelően számmá alakítja. Az esetleges hiba helyét a harmadik paraméterben adja meg.

VAL (S, V, CODE)

S

Az átalakítandó karakterlánc (STRING típusú).

V

Az átalakított érték (REAL vagy INTEGER típusú).

CODE

Az átalakításkor keletkezett hiba változója (INTEGER típusú). Hibátlan átalakítás esetén értéke 0, egyébként a hiba pozíciója.

WRITE : eljárás típusos állományokhoz

Típusos állományba ír ki egyenként komponenseket a megadott változók (kifejezések) alapján. Fontos, hogy a változók típus egyezzen az alaptípussal. Az állománymutató minden írás után automatikusan továbblép. Ha az állománymutató az állomány végére mutat, az állomány bővül.

WRITE (F, V1[, V2, ... , Vn])

F

Azon állomány logikai neve, melybe írni szeretnénk.

V1 ... Vn

Azon változók neve, amelyekből írni szeretnénk.

WRITE : eljárás szöveges állományokhoz

Az eljárás szöveges állomány aktuális pozíciójától ír ki a paraméterek típusától függően értékeket. Az állománymutató minden íráskor automatikusan továbblép. Ha az első paraméterként megadható logikai álló-

(Folytatás a következő oldalon)

mánynév hiányzik, akkor az eljárás a szabványos kimenetre, a képernyőre ír. Ilyenkor a kurzor az utolsó kiírt karakter után áll.

```
READ ([ F] V1[ ,V2, ... ,Vn] )
```

F

Logikai állomány (TEXT típusú).

V1 ... Vn

A kiírandó kifejezések, illetve változók neve. Ha a változó vagy kifejezés értéke szám, az STR eljárásnál ismertettekhez hasonlóan ábrázolási szélesség és tizedesjegy szélesség is megadható *V [:WIDTH [:DECIMALS]]* formában.

WRITELN : eljárás

Ugyanazt teszi mint a WRITE eljárás csak a kifejezések után még sorvégjelet is ír. Kifejezések nélkül csak sorvégjeleket tesz.

```
WRITELN ([ F] V1[ ,V2, ... ,Vn] )
```

F

Annak a logikai állománynak a neve, amelybe írni szeretnénk (TEXT típusú). Elhagyása esetén a képernyőre ír.

V1 ... Vn

A kiírandó kifejezések, változónevek, konstansok.

**MÁS VERZIÓKBÓL
ÁTVETT ELJÁRÁSEGYSÉGEK**

A TURBO3 eljárássegység

A TURBO3 eljárássegység a *Turbo Pascal 3.0-ban* deklarált függvények és eljárások gyűjteménye. Az eljárássegység segítségével a régebbi (3.0) Pascal programok újrafordíthatók, csak a program által használt eljárássegységek felsorolásánál kell a USES TURBO3 sort szerepeltetni.

A TURBO3 eljárássegység függvényei és eljárásai:

Append	FillChar
Arc	FillEllipse
Assign	FillPoly
AssignCrt	FindFirst
Bar	FindNext
Bar3D	FloodFill
BlockRead	Flush
BlockWrite	FreeMem
ChDir	FSplit
Circle	GetArcCoords
ClearDevice	GetAspectRatio
ClearViewPort	GetCBreak
Close	GetDate
CloseGraph	GetDir
ClrEol	GetFAttr
ClrScr	GetFillSettings
Dec	GetFTime
Delay	GetImage
Delete	GetIntVec
DelLine	GetLineSettings
DetectGraph	GetMem
Dispose	GetPalette
Ellipse	GetTextSettings
Erase	GetTime
Exec	GetVerify
Exit	GetViewSettings
Fail	GotoXY

Halt	Reset
HighVideo	RestoreCrtMode
Inc	Rewrite
InitGraph	Rmdir
Insert	RunError
InsLine	Sector
Intr	Seek
Keep	SetActivePage
Line	SetAllPalette
LineRel	SetAspectRatio
LineTo	SetBkColor
LowVideo	SetCBreak
Mark	SetColor
Mkdir	SetDate
Move	SetFAttr
MoveTo	SetFillPattern
New	SetFillStyle
NormVideo	SetFTime
NoSound	SetGraphMode
OutText	SetIntVec
OutTextXY	SetLineStyle
OvrClrBuf	SetPalette
Ovrlnit	SetRGBPalette
OvrlnitEMS	SetTextBuf
OvrSetBuf	SetTextJustify
OvrSetRetry	SetTextStyle
PackTime	SetTime
PieSlice	SetUserCharSize
PutImage	SetVerify
PutPixel	SetViewPort
Randomize	SetVisualPage
Read (text)	SetWriteMode
Read (typed)	Str
ReadLn	SwapVectors
Rectangle	TextBackground
Release	TextColor
Rename	TextMode

Truncate
UnPackTime
Val
Window

Write (text)
Write (typed)
WriteLn

A GRAPH3 eljárássegység

A GRAPH3 eljárássegység a *Turbo Pascal 3.0* grafikai függvényeinek és eljárásainak gyűjteménye. Annak, hogy ez az eljárássegység még létezik, az az oka, hogy támogatja a programozók körében jól ismert teknősbéka-grafikát.

A GRAPH3 eljárássegység függvényei és eljárásai:

Arc	Home
Back	NoWrap
Circle	Palette
ClearScreen	Pattern
ColorTable	PenDown
Draw	PenUp
FillPattern	Plot
FillScreen	PutPic
FillShape	SetHeading
Forwd	SetPenColor
GetDotColor	SetPosition
GetPic	ShowTurtle
GraphBackground	TurnLeft
GraphColorMode	TurnRight
GraphMode	TurtleDelay
GraphWindow	TurtleThere
Heading	TurtleWindow
HideTurtle	Wrap
HiRes	XCor
HiResColor	YCor

HIBAKÓDOK

A TURBO PASCAL 7.0 FORDÍTÁSI HIBAKÓDJAI

- 1 Out of memory
Elfogyott a memória.
- 2 Identifier expected
A fordító azonosítót vár.
- 3 Unknown identifier
Az azonosító nem létezik.
- 4 Duplicate identifier
Az azonosító már létezik.
- 5 Syntax error
Szintaktikai (helyesírási) hiba.
- 6 Error in real constant
Hiba a valós (típusú) konstansban.
- 7 Error in integer constant
Hiba az egész (típusú) konstansban.
- 8 String constant exceeds line
A karakter (típusú) konstans túlnyúlik a soron.
- 10 Unexpected end of file
Nem várt forrásállomány-vég.
- 11 Line too long
A sor túl hosszú.
- 12 Type identifier expected
A fordító típusazonosítót vár.
- 13 Too many open files
Túl sok nyitott állomány.
- 14 Invalid file name
Érvénytelen állományspecifikáció.
- 15 File not found
A keresett állomány nem létezik.
- 16 Disk full
Az aktuális meghajtó betelt.

- 17 Invalid compiler directive
Érvénytelen fordítási direktíva.
- 18 Too many files
Túl sok az egymásba ágyazott állomány.
- 19 Undefined type in pointer def
Nemdefiniált típus a Pointer deklarációban.
- 20 Variable identifier expected
A fordító változóazonosítót vár.
- 21 Error in type
Rossz típusdeklaráció.
- 22 Structure too large
Az összetett típus túl nagy.
- 23 Set base type out of range
Az alaptípus rendszáma kisebb, mint 0, vagy nagyobb, mint 255.
- 24 File components may not be files or objects
Az állomány komponense nem lehet állomány vagy objektum.
- 25 Invalid string length
Hibás karakterlánc hossz.
- 26 Type mismatch
Típusazonosítási, egyeztetési hiba; típusinkompatibilitás
- 27 Invalid subrange base type
Az alaptípus nem sorszámozott.
- 28 Lower bound > than upper bound
Az alsó határ nagyobb, mint a felső.
- 29 Ordinal type expected
A fordító sorszámozott típust vár.
- 30 Integer constant expected
A fordító egész konstanst vár.
- 31 Constant expected
A fordító konstanst vár.

- 32 Integer or real constant expected
A fordító egész vagy valós konstanst vár.
- 33 Pointer Type identifier expected
A fordító mutatóazonosítót vár.
- 34 Invalid function result type
Érvénytelen függvénytípus.
- 35 Label identifier expected
A fordító címkét vár.
- 36 BEGIN expected
A fordító a BEGIN kulcsszót várja.
- 37 END expected
A fordító az END kulcsszót várja.
- 38 Integer expression expected
A fordító egész típusú konstanst vár.
- 39 Ordinal expression expected
A fordító egész típusú konstanst vár.
- 40 Boolean expression expected
A fordító BOOLEAN típusú kifejezést vár.
- 41 Operand types do not match
A műveletnek nem lehet ilyen operandusa.
- 42 Error in expression
Hiba a kifejezésben.
- 43 Illegal assignment
Hibás értékadás.
- 44 Field identifier expected
A fordító mezőazonosítót vár.
- 45 Object file too large
A tárgykódállomány nagyobb, mint 64 kilobyte.
- 46 Undefined EXTERN
Hibás EXTERN definíció.
- 47 Invalid object file record
Az tárgykódállományban hibás rekord van.

- 48** Code segment too large
A kódszegmens túl hosszú.
- 49** Data segment too large
Az adatszegmens túl hosszú.
- 50** DO expected
A fordító a DO kulcsszót várja.
- 51** Invalid PUBLIC definition
Érvénytelen PUBLIC definiálás.
- 52** Invalid EXTRN definition
Érvénytelen EXTRN definiálás.
- 53** Too many EXTRN definitions
Érvénytelen EXTRN definiálás.
- 54** OF expected
A fordító az OF kulcsszót várja.
- 55** INTERFACE expected
A fordító az INTERFACE kulcsszót várja.
- 56** Invalid relocatable reference
Érvénytelen cím.
- 57** THEN expected
A fordító a THEN kulcsszót várja.
- 58** TO or DOWNT0 expected
A fordító a TO vagy a DOWNT0 kulcsszót várja.
- 59** Undefined forward
Hiányzó deklaráció.
- 61** Invalid typecast
Hibás típuskonverzió.
- 62** Division by zero
Osztas nullával.
- 63** Invalid file type
Érvénytelen állománytípus.
- 64** Cannot read or write variables of this type
Ez a típus nem szerepelhet a Read és a Write eljárásban.

- 65 Pointer variable expected
A fordító mutató típusú változót vár.
- 66 String variable expected
A fordító karakterlánc típusú változót vár.
- 67 String expression expected
A fordító karakterlánc típusú kifejezést vár.
- 68 Circular unit reference
Kereszthivatkozás az eljárássegységekben (UNIT-okban).
- 69 Unit name mismatch
Eljárásegység név (UNIT) ütközése, hibája.
- 70 Unit version mismatch
Eljárásegység (UNIT) verzióhiba.
- 71 Internal stack overflow
Veremtúlcsordulás.
- 72 Unit file format error
Hibás eljárásegység-formátum.
- 73 Implementation expected
A fordító az IMPLEMENTATION kulcsszót várja.
- 74 Constant and case types don't match
A konstans és a CASE szelektor típusa nem egyezik.
- 75 Record or object variable expected
A fordító rekord vagy objektum típusú azonosítót vár.
- 76 Constant out of range
A konstans kívül esik a megengedett tartományon.
- 77 File variable expected
A fordító állomány típusú változót vár.
- 78 Pointer expression expected
A fordító mutató típusú kifejezést vár.
- 79 Integer or real expression expected
A fordító egész vagy valós típusú kifejezést vár.
- 80 Label not within current block
A címke a blokkon kívül van.

- 81 Label already defined
A címke már létezik.
- 82 Undefined label in preceding stmt part
A címke még nem létezik.
- 83 Invalid @ argument
Hibás @-művelet.
- 84 UNIT expected
A fordító a UNIT kulcsszót várja.
- 85 ";" expected
A fordító a ; jelet várja.
- 86 ":" expected
A fordító a : jelet várja.
- 87 "," expected
A fordító a , jelet várja.
- 88 "(" expected
A fordító a (jelet várja.
- 89 ")" expected
A fordító a) jelet várja.
- 90 "=" expected
A fordító a = jelet várja.
- 91 ":=" expected
A fordító a := jeleket várja.
- 92 "[" or "(." expected
A fordító a [vagy a (. jeleket várja.
- 93 "]" or ".)" expected
A fordító a] vagy a .) jelet várja.
- 94 "." expected
A fordító a . jelet várja.
- 95 ".." expected
A fordító a .. jelet várja.
- 96 Too many variables
Túl sok változó.

- 97** Invalid FOR control variable
A FOR ciklusváltozója érvénytelen.
- 98** Integer variable expected
A fordító egész típusú változót vár.
- 99** Files types are not allowed here
Az állomány típus nem megengedett.
- 100** String length mismatch
A karakterlánc vagy tömb hossza nem megfelelő.
- 101** Invalid ordering of fields
Hibás mezősorrend a rekord típusú állandóban.
- 102** String constant expected
A fordító szövegkonstanst vár.
- 103** Integer or real variable expected
A fordító egész vagy valós típusú változót vár.
- 104** Ordinal variable expected
A fordító sorszámozott típusú változót vár.
- 105** INLINE error
Hiba az INLINE direktíva blokkjában.
- 106** Character expression expected
A fordító karakter típusú kifejezést vár.
- 107** Too many relocation items
Túl nagy a program - túl sok az adatáthelyezés.
- 108** Overflow in arithmetic operation
Túlcsordulás aritmetikai műveletnél.
- 109** No enclosing For, While or Repeat statement
Break és a Continue eljárások a FOR, While vagy a Repeat ciklusok magján kívül.
- 112** CASE constant out of range
A CASE utasítás konstansa kívül esik a megengedett tartományon.
- 113** Error in statement
Hibás utasítás.

- 114** Cannot call an interrupt procedure
Az INTERRUPT típusú eljárást nem lehet közvetlen módon hívni.
- 116** Must be in 80X87 mode to compile
Az ilyen műveleteket a \$N+ direktíva mellett kell fordítani.
- 117** Target address not found
A megadott címen nincs utasítás.
- 118** Include files are not allowed here
Az Include direktíva nem megengedett.
- 119** No inherited methods are accessible here
Metóduson kívül, vagy előd nélkül metódusban használt INHERITED eljárás.
- 121** Invalid qualifier
Érvénytelen minősítés.
- 122** Invalid variable reference
Érvénytelen változóhivatkozás.
- 123** Too many symbols
Túl sok szimbólum.
- 124** Statement part too large
A végrehajtandó rész túl hosszú.
- 126** Files must be var parameters
Az állomány típusú paraméternek változónak kell lennie.
- 127** Too many conditional symbols
Túl sok a feltételes szimbólum.
- 128** Misplaced conditional directive
Pár nélküli feltételes direktíva.
- 129** ENDIF directive missing
Hiányzó ENDIF direktíva.
- 130** Error in initial conditional defines
Az *Options* menü *Compiler* menüpontjának *Conditional defines* sorában hiba van.

- 131** Header does not match previous definition
A szubrutinok fejlécei ütköznek.
- 133** Cannot evaluate this expression
A kifejezés kiértékelhetetlen.
- 134** Expression incorrectly terminated
Helytelen kifejezés.
- 135** Invalid format specifier
Érvénytelen formátum.
- 136** Invalid indirect reference
Érvénytelen indirekt hivatkozás.
- 137** Structured variables are not allowed here
Összetett típusú változó nem használható.
- 138** Cannot evaluate without System unit
Ez a kifejezés a SYSTEM eljárássegység nélkül nem használható.
- 139** Cannot access this symbol
Elérhetetlen szimbólum.
- 140** Invalid floating-point operation
Érvénytelen lebegőpontos művelet.
- 141** Cannot compile overlays to memory
Az *overlay* alegység nem fordítható memóriába.
- 142** Pointer or procedural variable expected
A fordító mutató vagy eljárás típusú változót várt.
- 143** Invalid procedure or function reference
Érvénytelen eljárás- vagy függvényhívás.
- 144** Cannot overlay this unit
Az egység nem szervezhető *overlay* struktúrába.
- 145** Too many nested scopes
Túl sok az egymásba ágyazott terület (maximum 512).
- 146** File access denied
Hozzáférés az állományhoz nem megengedett.
- 147** Object type expected
A fordító objektum típust vár.

- 148 Local object types are not allowed
A lokális objektum típus nem megengedett.
- 149 Virtual expected
A fordító a VIRTUAL kulcsszót várja.
- 150 Method identifier expected
A fordító metódus azonosítót vár.
- 151 Virtual constructors are not allowed
A konstruktor nem lehet virtuális.
- 152 Constructor identifier expected
A fordító konstruktor azonosítót vár.
- 153 Destructor identifier expected
A fordító destruktorkonstruktor azonosítót vár.
- 154 Fail only allowed within constructors
A FAIL csak konstruktorból hívható.
- 155 Invalid combination of opcode and operands
Érvénytelen utasítás- és operanduskombináció.
- 156 Memory reference expected
A fordító memóriahivatkozást vár.
- 157 Cannot add or subtract relocatable symbols
Relokálható szimbólumok nem adhatók össze és nem vonhatók ki egymásból.
- 158 Invalid register combination
Érvénytelen regiszterkombináció.
- 159 286/287 instructions are not enabled
A 286/287 kódok nem használhatók.
- 160 Invalid symbol reference
Érvénytelen hivatkozás szimbólumra.
- 161 Code generation error
Kódgenerálási hiba.
- 162 ASM expected
A fordító az ASM kulcsszót várja.

REÁL KÖNYVAJÁNLAT

Még kapható...

Kis Balázs: DOS 5.0 & 6.0	Ára: 330 Ft
Szabadhegyi Csaba: WORD 5.5	Ára: 275 Ft
EXECOM szótár	Ára: 275 Ft
Borbély Viktor: Norton Utilities 6.0	Ára: 220 Ft
Dedinszky Ferenc: A gépvásárlástól a programozásig	Ára: 275 Ft

Friss könyveink

Borbély Viktor: Norton Editor 1.5, 2.0, Norton Commander 3.0 és 4.0	Ára: 440 Ft
Greznér F.: EXCEL 4.0 for Windows	Ára: 330 Ft
Bakó-Burán: Az AutoCAD parancsai 1. és 12. verzió	Ára: 220 Ft
Pongor György: Szabványos Pascal Programozás, algoritmusok	Ára: 770 Ft

Előkészületben

K. Borbély Katalin: Quattro Pro 4.0	Ára kb: 330 Ft
Borbély Viktor: Norton Desktop 2.2 for Windows	Ára kb: 440 Ft

A könyvek megrendelhetők
a Múzsák Kiadó Kft.-nél
1139, Budapest, Kartács u. 24-26.
Tel.: 1208572 Fax: 1494185
Keresse Gál Györgynét!

Pongor György

Szabványos PASCAL

Programozás és algoritmusok

Könyvünk olyan tankönyv, amely a szabványos PASCAL nyelv leírását is tartalmazza. Elsősorban egyetemek és főiskolák programozás-oktatásához készült, de az érdeklődő középiskolásoknak is ajánljuk.

Ez a könyv több fontos dologban is különbözik más PASCAL-tankönyvektől.

- Nemcsak a Pascal nyelv szabványát tartalmazza, hanem a szabályok részletes magyarázatát is, például azt, hogy az egyes szabályoknak milyen — esetleg rejtett — következményük van. Így a könyv hivatkozási kézikönyvnek is használható, de talán még fontosabb, hogy általa a kezdő programozó is hozzájuthat olyan ismeretekhez, amelyeket mások hosszú gyakorlattal szereztek meg.
- Részletesen foglalkozik a tapasztalok szerint legnehezebben megérthető témákkal, nevezetesen a rekurzióval és más, mutatók segítségével felépíthető dinamikus adatszerkezetekkel. Ezeket más könyvek általában kisebb súllyal tárgyalják.
- Az egyes nyelvi eszközöknél mindig megadja azokat a tipikus hibákat, félreértéseket, amiket a programozók elkövetnek. Ezek összegyűjtését a szerző hosszú oktatói és vizsgáztatói gyakorlata tette lehetővé,
- E második, átdolgozott kiadásban a szerző a legmesszemenőkig figyelembe vette annak a több ezer hallgatónak az észrevételeit, megjegyzéseit, akik a sikeres első kiadásból tanultak.

Bakó András – Burán Zsuzsa

Az AutoCAD parancsai

11. és 12. verzió

A könyv az AutoCAD 11. és 12. verziójának alrendszerében és a szilárdtest-modellező, területmodellező és árnyékoló alrendszereiben használható utasításokat, valamint azok paramétereit ismerteti röviden.

Mindenütt szerepelnek az angol parancsok magyar változatban használható megfelelői is és a magyar-angol parancsmegfeleltetés is.

Jól használható a korábbi AutoCAD-verziókkal is, mivel a rendszer a fejlesztések során csupán bővült, lényege azonban nem változott.

Elsősorban azoknak ajánljuk, akik már ismerik valamennyire az AutoCAD-et. Rajtuk kívül azonban segítséget nyújt a tanfolyamokon vagy egyéb oktatásban résztvevőknek is, mivel nem kell terjedelmes – rendszerint föl nem lelhető – kézikönyveket forgatniuk.

Ára: 440

