

Angster Erzsébet

Kertész László

Turbo Pascal 5.0 – 5.5

'A' .. 'Z'



**ANGSTER ERZSÉBET — KERTÉSZ LÁSZLÓ**

**Turbo Pascal 5.0 – 5.5**  
**'A'..'Z'**

© Angster Erzsébet – Kertész László , 1990

Első kiadás

Minden jog fenntartva. A szerzők együttes írásbeli hozzájárulása nélkül tilos a könyvet vagy annak részeit bármilyen eljárással másolni, sokszorosítani, terjeszteni.

Lektorálta: Kertész György

Felelős kiadó: Angster Erzsébet, Kertész László

Készült a PRINT PACK nyomdájában, 1990-ben.

Felelős vezető: Báder György

Megrendelő száma: **17-1990**

**ISBN 963 400 131 9**

<b>Jelölések</b> .....	<b>1</b>
<b>Keretrendszer</b> .....	<b>3</b>
Állományok .....	3
A keretrendszer indítása .....	4
Az installálás indítása .....	5
Installáló menüfunkciók .....	6
<b>Menürendszer</b> .....	<b>11</b>
File .....	11
Edit .....	13
Run .....	13
Compile .....	14
Options .....	16
Debug .....	21
Break/watch .....	22
<b>Forróbillentyűk</b> .....	<b>25</b>
<b>A szövegszerkesztő parancsai</b> .....	<b>27</b>
Kurzormozgatás .....	27
Beszúrás/törlés .....	28
Blokk-kezelés .....	28
Egyéb funkciók .....	29
Keresés, keresés és csere opciói .....	29
A szövegszerkesztő opciói .....	30

# Tartalomjegyzék

<b>Szintaktika</b> .....	<b>31</b>
Definíciók .....	31
A Pascal program elemei .....	31
Szimbólumok .....	31
Fenntartott szavak .....	31
Programsor .....	32
Megjegyzés .....	32
Direktíva .....	32
Azonosító .....	32
Minősített azonosító .....	32
Címke .....	32
Számkonstans .....	32
Szövegkonstans .....	33
Formális paraméterlista .....	33
Aktuális paraméterlista .....	33
Függvényhívás .....	33
Típuskonvertált változó .....	33
Típuskonvertált érték .....	33
Változóhivatkozás .....	33
Halmazkonstruktor .....	34
Kifejezés .....	34
Konstans kifejezés .....	34
Feltétel .....	35
Utasítás .....	35
Egyszerű utasítások .....	35
Értékadás .....	35

Eljáráshívás .....	35
Metódushívás .....	35
Vezérlésátadás – GoTo .....	35
Gépi kódú betét – InLine .....	35
Struktúrált utasítások .....	36
Összetett utasítás – Begin–End .....	36
Elágazás – If .....	36
Választás – Case .....	36
Hátultesztelő ciklus – Repeat–Until .....	36
Elöltesztelő ciklus – While .....	36
Elöltesztelő léptető ciklus – For .....	37
Minősítő utasítás – With .....	37
A program felépítése .....	38
A metódus felépítése .....	39
Speciális metódusparaméter: SELF .....	39
Constructor, Destructor .....	39
Az egység felépítése .....	40
Deklarációk érvényességi köre .....	41
<b>Adattípusok, műveletek .....</b>	<b>43</b>
Definíciók .....	43
Sorszámozott típusok .....	43
Egyszerű típusok .....	43
Struktúrált (összetett) típusok .....	43
Mutató típusok .....	43
Eljárás típusok .....	43

## Tartalomjegyzék

Turbo Pascal adattípusok .....	43
Logikai típus – Boolean .....	43
Karakter típus – Char .....	43
Egész típusok – ShortInt, Integer, LongInt, Byte, Word .....	44
Felsorolt típus .....	44
Intervallum típus .....	44
Valós típusok – Real, Single, Double, Extended, Comp .....	44
Karakterlánc típus – String .....	45
Tömb típus – Array .....	45
Rekord típus – Record .....	46
Objektum típus – Object .....	46
Halmaz típus – Set Of .....	47
Típusos állomány – File Of .....	47
Szöveges állomány – Text .....	48
Típusnélküli állomány – File .....	48
Típusos mutató .....	48
Típusnélküli mutató – Pointer .....	48
Eljárás típus – Procedure .....	49
Függvény típus – Function .....	49
Speciális tömbök – Mem, MemW, MemL, Port, PortW .....	49
Típusazonosság .....	50
Típus-kompatibilitás .....	50
Értékadás-kompatibilitás .....	50
Műveletek .....	52
Kifejezések kiértékelése .....	56
Értékadás, értékparaméter átadása .....	57

<b>Direktívák</b> .....	<b>59</b>
Kapcsoló- és paraméter direktívák .....	59
\$A+/- (Align data) .....	59
\$B+/- (Boolean evaluation) .....	59
\$D+/- (Debug information) .....	60
\$E+/- (Emulation) .....	60
\$F+/- (Force far calls) .....	60
\$I+/- (I/O checking) .....	60
\$I (Include) .....	61
\$L+/- (Local symbols) .....	61
\$L (link) .....	61
\$M (Memory sizes) .....	61
\$N+/- (Numeric processing) .....	62
\$O+/- (Overlays allowed) .....	62
\$O (overlay) .....	62
\$R+/- (Range checking) .....	62
\$S+/- (Stack checking) .....	63
\$V+/- (Var-string checking) .....	63
Feltételes direktívák .....	63
\$DEFINE .....	63
\$UNDEF .....	64
\$IFDEF .....	64
\$IFOPT .....	64
\$ELSE .....	64
\$ENDIF .....	64
Előre definiált feltételes szimbólumok .....	65



# Tartalomjegyzék

---

Speciális direktívák .....	65
INTERRUPT .....	65
INLINE .....	65
EXTERNAL .....	66
FORWARD .....	66
VIRTUAL .....	66
<b>System egység .....</b>	<b>67</b>
Konstansok .....	67
Változók .....	67
Eljárások, függvények .....	70
<b>Crt egység .....</b>	<b>89</b>
Konstansok .....	89
Változók .....	89
Eljárások, függvények .....	91
<b>Printer egység .....</b>	<b>95</b>
Változó .....	95
<b>Dos egység .....</b>	<b>97</b>
Típusok .....	97
Konstansok .....	99
Változó .....	99
Eljárások, függvények .....	100
<b>Graph egység .....</b>	<b>109</b>
Típusok .....	109
Konstansok .....	111
Változók .....	114
Eljárások, függvények .....	114

<b>Overlay egység</b> .....	<b>131</b>
Típus .....	131
Konstansok .....	131
Változók .....	131
Eljárások, függvények .....	132
<b>Turbo3 egység</b> .....	<b>136</b>
Változók .....	136
Eljárások, függvények .....	136
<b>Graph3 egység</b> .....	<b>137</b>
Konstansok .....	137
Eljárások, függvények .....	137
<b>Fordítási hibakódok</b> .....	<b>139</b>
<b>Futási hibakódok</b> .....	<b>149</b>
DOS hibák .....	149
B/K hibák .....	150
Kritikus hibák .....	151
Fatális, mindig programleállást okozó hibák .....	151
<b>Billentyűzet kódtábla</b> .....	<b>153</b>
<b>ASCII kódtáblázat</b> .....	<b>156</b>
<b>Grafikus üzemmódok jellemzői</b> .....	<b>158</b>
<b>Memóriatérképek</b> .....	<b>159</b>
A program memóriatérképe .....	159
A VMT (Virtual Method Table) felépítése .....	160
A szubrutinverem felépítése .....	160
Valós számábrázolás .....	161
<b>Nyomtató vezérlőkódok</b> .....	<b>162</b>

## Előszó

Könyvünk a Turbo Pascal 5.0 és 5.5 verziók környezeti és programozói összefoglalója. A Borland cég már közel ezer oldalon tárgyalja e rendszereket – nehéz feladatot vállaltunk tehát magunkra: igyekeztünk hordozható méretben, minél több információt adni az Olvasónak. Ez a könyv az IBM PC környezetben többé-kevésbé jártas, Turbo Pascal-ban nem kezdő programozóknak kíván mindennapi segítséget adni.

Itt csak a keretrendszert és az ahhoz tartozó Installáló programot tárgyaljuk, az egyéb – véleményünk szerint ritkábban használatos – kiegészítő szoftverekkel nem foglalkozunk.

Az 5.5-ös verzióra vonatkozó információkat minden esetben jelezzük, a tárgymutatóban az ezeket tartalmazó oldalhivatkozások külön is megtekinthetők.

Bár törekedtünk arra, hogy teljes, átfogó leírást adjunk minél kevesebb hibával a Kedves Olvasó kezébe, előfordulhat, hogy bizonyos kérdéseire nem kap választ. Kérjük, észrevételeit, javaslatait írja meg az alábbi címmel:

**Turbo Pascal 5.0-5.5 'A'..'Z'**

2022 Tahi

Pf. 4

*a szerzők*

**KERETRENDSZER  
MENÜRENDSZER  
FORRÓBILLENTYŰK  
A SZÖVEGSZERKESZTŐ PARANCSAI  
SZINTAKTIKA  
ADATTÍPUSOK, MŰVELETEK  
DIREKTÍVÁK  
SYSTEM EGYSÉG  
CRT EGYSÉG  
PRINTER EGYSÉG  
DOS EGYSÉG  
GRAPH EGYSÉG  
OVERLAY EGYSÉG  
TURBO3 EGYSÉG  
GRAPH3 EGYSÉG  
FORDÍTÁSI HIBAKÓDOK  
FUTÁSI HIBAKÓDOK  
TÁBLÁZATOK**

<b>KIEMELT</b>	a megadott írásmóddal használandó
<i>Dőltbetű</i>	értelemszerűen behelyettesítendő
[ ]	opcionális
...	ismételhető
	a jel egyik vagy másik oldalán szereplő kifejezés használandó
{ }	A jelek közötti sorok egyike használandó
5.5	csak 5.5 verzióban

## Állományok

.PAS	Turbo Pascal forrásprogram
.BAK	forrásprogram biztonsági másolat (backup)
.OBJ	tárgykódot tartalmazó állomány (object)
.EXE	lefordított, futtatható program (executable)
.TPU	lefordított egység (Turbo Pascal Unit)
.MAP	térképállomány (map)
.TPL	egységkönyvtár (Turbo Pascal Library)
.TP	a keretrendszer konfigurációs állománya
.CFG	a TPC.EXE konfigurációs állománya
.PCK	állapotállomány (pick)
.DOC	egységek dokumentációs állománya (document)
.BGI	grafikus meghajtók (Borland Graphics Interface)
.CHR	grafikus karakterkészletek (characters)
TURBO.EXE	Turbo Pascal keretrendszer
TPC.EXE	Turbo Pascal parancssor-fordító (compiler)
TURBO.HLP	segítségállomány (help)
TURBO.TPL	egységkönyvtár
TPUMOVER.EXE	egységmozgató program könyvtárakhoz
TINST.EXE	keretrendszert installáló program (Installation)
TINSTXFR.EXE	automatikus installáló program, mely meglévő 4.0 illetve 5.0 <sup>5.5</sup> verziójú TURBO.EXE alapján dolgozik

## KERETRENDSZER

---

MAKE.EXE	program menedzser
GREP.COM	szövegrészkereső segédprogram
TOUCH.COM	állomány dátum aktualizáló segédprogram
BINOBJ.EXE	bináris / tárgy kód átalakító program
TPCONFIG.EXE	.TP / .CFG , .CFG / .TP átalakító program
README	kiegészítő információk a gyári dokumentációhoz
README.COM	a README állományt megjelenítő program
NONAME.PAS	F/New után a forrásprogram neve
UPGRADE.EXE	Turbo Pascal 3.0 forrásprogramokat 5.0 illetve 5.5 <sup>5.5</sup>
UPGRADE.DTA	verzió alá átiró program és adatállomány

### A keretrendszer indítása

TURBO [*/C*konfigurációs állomány]

[*forrásállomány*] [ { */B*  
*/M* } ] [ */D* ]

- */C*konfigurációs állomány

Hatására a keretrendszer az itt megadott állomány alapján állítja be a menü, a fordító és a szerkesztő opcióit. A */C* után nem állhat szóköz. Elhagyása esetén az aktuális illetve az installáláskor megadott Turbo directory könyvtár TURBO.TP állománya lesz a konfigurációs állomány.

Ha ez nincs, akkor a TURBO.EXE szerinti (Installálással megváltoztatható) alapértelmezések lesznek érvényesek.

- *forrásállomány*

Megadása esetén automatikusan betöltődik a szövegszerkesztőbe.

- */B*

A keretrendszer betöltődése után a **C/Build** menüpont automatikusan aktivizálódik. A fordítás befejeztével a vezérlés visszakerül a DOS-hoz. Az esetleges üzenetek az aktuális kimeneti periférián jelennek meg.

- */M*

Megfelel a */B* opciónak, de a **C/Make** menüpontot aktivizálja.

- */D*

Kettős monitor üzemmód használata. A keretrendszer képe az inaktív, a kimeneti (output) ablak pedig az aktív DOS monitoron jelenik meg. A menürendszer **R/User** screen pontja ilyenkor hatástalan. Ha a gépben nem található két monitorvezérlő kártya, nincs hatása.

! Ne változtassuk meg az aktív DOS monitort, míg ki nem szállunk véglegesen (**F/Quit**) a keretrendszerből!

! Azon programjainkat, melyek az inaktív monitor vezérlőjét direkt portcímzéssel kezelik, kettős monitor üzemmódban ne futtassuk!

! Kettős monitort kezelő programjainkat ne futtassuk a keretrendszerből, ha az is kettős monitor üzemmódban dolgozik!

## Az installálás indítása

**TINST** [*/B*] [*állománynév*]



## KERETRENDSZER

---

Az itt megadott keretrendszert – elhagyása esetén az aktuális katalógusban található TURBO.EXE állományt – installálja. Színes monitorvezérlő kártya és kompozit monitor esetén használjuk a /B opciót!

### Installáló menüfunkciók

A táblázatban kiemelten jelöljük a gyárilag alapértelmezett beállítást. A csillaggal jelölt menüpontok opciói csak az Installáló programmal állíthatók át, a többi a keretrendszerben felülbíráható (leírásukat lásd MENÜRENDSZER).

#### Compile

Destination

Memory / Disk

Primary file

**nincs** / elsődleges állomány

#### Options/Compiler

Range checking

On / **Off**

Stack checking

On / **Off**

I/O checking

On / **Off**

Force far calls

On / **Off**

Overlays allowed

On / **Off**

Align data

**Word** / Byte

Var-string checking

**Strict** / Relaxed

Boolean evaluation

**Short circuit** / Complete

Numeric processing

**Software** / Hardware

Emulation

On / **Off**

Debug information

On / **Off**

Local symbols

On / **Off**

Conditional defines

**üres** / feltételes direktívák

Memory sizes

**(16384, 0, 655360)**

## Options/Linker

Map file (Off / Segments / Publics / Detailed)  
 Link buffer Memory / Disk

## Options/Environment

Config auto save On / Off  
 Edit auto save On / Off  
 Backup files On / Off  
 Zoom windows On / Off  
 Full graphics save On / Off

A grafikus kimeneti képernyők mentéséhez a rendszer 8K memóriát foglal le. Ha csak szöveges képernyővel dolgozunk, ezen opció Off-ra állításával memóriát nyerünk.

Screen size 25 / 43/50

## Options for editor

Szövegszerkesztési opciók, a szövegszerkesztő alapállapotainak beállítása, memória-gazdálkodás:

\* Insert mode On / Off  
 \* Autoindent mode On / Off  
 \* Use tabs On / Off  
 \* Optimal fill On / Off  
 \* Backspace unindents On / Off  
 Tab size 8 / 2..16

Ezen opciók ideiglenesen átállíthatók a keretrendszerben, jelentésük A SZÖVEGSZERKESZTŐ PARANCSAI részben megtalálhatók.

\* Editor buffer size 65534 / 20000..65534

A szöveg számára fenntartott memória maximális mérete.

\* Make use of EMS memory On / Off

On: szövegszerkesztés a kiterjesztett memóriában történik.

## KERETRENDSZER

---

### Options/Directories

Turbo directory

gyári lemezről installált

EXE & TPU directory

aktuális

Include directories

aktuális

Unit directories

gyári lemezről installált

Object directories

aktuális

Pick file name

aktuális/TURBO.PCK

### Options/Parameters

üres

### Debug

Integrated debugging

On / Off

Standalone debugging

On / Off

Display swapping

None / Smart / Always

### \* Editor commands

A szövegszerkesztő vezérlő-karaktereinak beállítása. Az alapértelmezéseket lásd: A SZÖVEGSZERKESZTŐ PARANCSAI.

### \* Mode for display

Default / Color / Black and white /  
Monochrome / LCD

Képernyőüzemmód beállítása. Default: a TURBO.EXE indulásakor detektált üzemmód; Color: színes monitorvezérlő esetén 80 oszlopos színes üzemmód; Black and white: színes monitorvezérlő esetén 80 oszlopos fekete-fehér üzemmód; Monochrome: monokróm monitorvezérlő esetén monokróm üzemmód; LCD: folyékony kristályos képernyőre vonatkozó üzemmód, színes monitorvezérlő esetén fekete-fehér kép.

\* **Set colors**

A keretrendszer képernyőszíneinek beállítása.

Customize colors (színek beállítása külön-külön)

Default color set (gyári alapszínkészlet)

Turquoise color set (türkiz színkészlet)

Version 4.0 color set (4.0 verzió színkészlet)

\* **Resize windows**                      8 sor / 1..16 sor

A kimeneti (output) ablak méretének illetve a figyelő- (watch) ablak maximális méretének beállítása.

**Quit/save**

Kilépés mentéssel vagy anélkül.

### File/Load (F3)

Forrásnyelvi szöveg betöltése. Az ablakban \*.PAS vagy az itt utoljára megadott állományspecifikáció jelenik meg. Bármely, a BACKSPACE és kurzorvezérlőktől eltérő billentyű leütésére a mező kitöltése előlről kezdődik. Ha a névben \* vagy ? szerepel, katalóguslistából választhatunk betöltendő állományt vagy újabb alkatalógust. A listában kurzorvezérlő billentyűkkel és kezdőbetűvel pozícionálhatunk. Ha az állomány nem létezik az illető katalógusban, akkor a megadott névvel új szöveget szerkeszthetünk. Ha nem adunk meg kiterjesztést, az automatikusan .PAS lesz. (Katalógusváltás nem történik, mentéskor az állományspecifikáció a mérvadó.)

### File/Pick (ALT-F3)

Forrásnyelvi szöveg betöltése az állapotlista alapján. Az állapotlista az utoljára szerkesztett maximálisan nyolc forrásállomány specifikációját (és állapotát) tartalmazza veremszerűen. A listában látható utolsó sor megfelel az F/Load pontnak.

### File/New

Új szöveg szerkesztése. A szövegszerkesztőben lévő szöveg törlődik; a forrásállomány automatikusan NONAME.PAS lesz az aktuális lemez aktuális katalógusában. A legközelebbi mentéskor ezt a rendszer átnevezteti.

### File/Save (F2)

A forrásnyelvi szöveg elmentése és a szerkesztés folytatása. Ha az O/E/Backup files pont On állapotú, biztonsági másolat (.BAK állomány) is készül. Ha a név NONAME.PAS, a rendszer átnevezteti.

## MENÜRENDSZER

---

### File/Write to

A szövegszerkesztőben lévő szöveg átnevezése és mentése. Ha nem adunk meg kiterjesztést, az automatikusan .PAS lesz. Ha az ablakba beírt névben \* vagy ? szerepel, katalóguslistából választhatunk felülírandó állományt vagy újabb alkatalógust. A listában kurzorvezérlő billentyűkkel és kezdőbetűvel pozícionálhatunk.

### File/Directory

Katalógus kérése a képernyőre. Az ablakban \*.\* vagy az itt utoljára megadott állományspecifikáció jelenik meg. A névben \* vagy ? is szerepelhet. A katalóguslistából újabb alkatalógust választhatunk, de a funkciót használhatjuk állomány betöltésére is. A listában kurzorvezérlő billentyűkkel és kezdőbetűvel pozícionálhatunk. (Katalógusváltás nem történik.)

### File/Change dir

Az aktuális lemez és/vagy katalógus megváltoztatása. A funkció annyiban tér el az operációs rendszer CD parancsától, hogy lemezegység megadása esetén át is vált az illető egységre. Kilépés után az itt megadott lemez és/vagy katalógus lesz az aktuális.

### File/OS shell

Ideiglenes kilépés az operációs rendszerbe. A Turbo Pascal környezetbe EXIT paranccsal lehet visszatérni.

### File/Quit (ALT-X)

Végleges kilépés a Turbo Pascal keretrendszerből.

**Edit (ALT-E)**

Belépés a szövegszerkesztőbe.

**Run/Run (CTRL-F9)**

Az utoljára fordított program (tovább-)futtatása a következő töréspontig vagy a program végéig. A rendszer a C/Primary file pontban megadott, vagy annak hiányában a szövegszerkesztőben lévő programot automatikusan lefordítja (a C/Make ponthoz hasonlóan), ha ez még nem történt meg, vagy módosítottunk a programon a legutóbbi fordítás óta. A program parancssor-paraméterei az O/Parameters pontban adhatók meg.

**Run/Program reset (CTRL-F2)**

Program inicializálása, előkészítése újabb futtatásra. A futási sorjelző eltűnik, a program által lefoglalt memóriaterületek felszabadulnak. A funkció a nyitott állományokat lezárja. A legközelebbi futtatás a program elején indul, az előzőleg beállított töréspontok megmaradnak.

**Run/Go to cursor (F4)**

Program (tovább-)futtatása a kurzor sorálg, illetve egy esetlegesen közbeeső töréspontig. A funkció csak az O/C/Debug information On (\$D+) és a D/Integrated debugging On állapota mellett használható.

**Run/Trace into (F7)**

Lépésenkénti futtatás: a program a futási sorjelzővel kijelölt sor utasításait végrehajtja, majd a futási sorjelző a következő végrehajtandó sorra lép. A saját eljárásokat és függvényeket is lépésenként hajtja végre. Szükség esetén a megfelelő forrásállományt a rendszer betölti; ha nem találja, rá-

## MENÜRENDSZER

---

kérdez az állomány nevére. A funkció csak az **O/C/Debug information On** (\$D+) és a **D/Integrated debugging On** állapota mellett használható. Futtatható program hiányában a **C/Make** automatikusan aktivizálódik.

### Run/Step over (F8)

Lépésenkénti futtatás: megfelel az **R/Trace into** funkciónak, de az eljárásokat és függvényeket kompletten lefuttatja. A funkció csak az **O/C/Debug information On** (\$D+) és a **D/Integrated debugging On** állapota mellett használható.

### Run/User screen (ALT-F5)

A képernyő átváltása a kimeneti képernyőre, amelyről bármely billentyű leütésével visszatérhetünk. Kettős monitor üzemmódban (ha a keretrendszert /D paraméterrel indítottuk, és a gépben valóban van két monitorvezérlő kártya) a funkciónak nincs hatása.

### Compile/Compile (ALT-F9)

A szövegszerkesztőben lévő program vagy egység fordítása memóriába vagy lemezre a **C/Destination** menüpont függvényében.

### Compile/Make (F9)

Megfelel a **C/Compile** menüpontnak, de a program által használt módosított, az egységkönyvtárban nem található egységeket újrarendelt (a dátum és idő alapján). Ha szerepel állománynév a **C/Primary file** pontban, akkor azt fordítja, nem a bentlévő szöveget!



## Compile/Build

Megfelel a **C/Make** menüpontnak, de minden, a program által használt egységet (amennyiben nincs az egységkönyvtárban, és a hozzátartozó forrásállomány megtalálható) feltétel nélkül újrafordít.

## Compile/Destination

Meghatározza, hogy a fordítás a memórlába (Memory) vagy lemezre (Disk) történjen-e. A lemezre írt kód – programok esetében .EXE, egységek esetén .TPU kiterjesztéssel – az **O/D/EXE** & TPU directory pontban beállított (ennek hiányában az aktuális) katalógusba kerül. A létrejövő állomány neve a **C/Primary file** pontban beállított, illetve a szövegszerkesztőben lévő forrásállomány neve lesz. **C/Make** vagy **C/Build** funkcióval történő fordítás során – a menüpont beállításától függetlenül – az újrafordított egységek .TPU állományait a rendszer felülírja.

## Compile/Find error

Az operációs rendszer alól futtatott, futási hibával leállt program hibás utasításának megkeresése. A program leállásakor kírt futási hibacímét (szegmens:eltolás) kell itt megadni, miután betöltöttük a programot tartalmazó forrásállományt a szövegszerkesztőbe, vagy nevét megadtuk a **C/Primary file** pontban. A rendszer a programot újrafordítja, majd a kurzort a címnek megfelelő (hibás utasítást tartalmazó) sorra viszi, amennyiben a címhez utasítás tartozik. A futási hibacím előtt kírt hibakód alapján a leállás oka visszakereshető (lásd **FUTÁSI HIBAKÓDOK**). A funkció csak az **O/C/Debug information On (\$D+)** és a **D/Integrated debugging On** állapota mellett használható.

## MENÜRENDSZER

---

### Compile/Primary file

Elsődleges forrásállomány megadása. Az itt megadott programot vagy egységet fordítja a **C/Make** és **C/Build** funkció, egyébként a szövegszerkesztőben lévőt.

### Compile/Get info

A forrásállományról és az utóljára fordított egységről vagy futtatható programról ad információkat, valamint megjeleníti a szabad memóriaméretet.

### Options/Compiler/Range checking

Indexek és intervallum típusok esetében értékhatár-ellenőrző kód generálását engedélyezi (On) vagy tiltja (Off). Lásd **\$R+/-** direktíva.

### Options/Compiler/Stack checking

A verem ellenőrzését engedélyezi (On) vagy tiltja (Off) szubrutinok hívásakor. Lásd **\$S+/-** direktíva.

### Options/Compiler/I/O checking

B/K műveletek ellenőrzését engedélyezi (On) vagy tiltja (Off). Lásd **\$I+/-** direktíva.

### Options/Compiler/Force far calls

Szubrutinok hívási módját (távoli – On / automatikus – Off) vezérli. Lásd **\$F+/-** direktíva.

### Options/Compiler/Overlays allowed

Egységek esetében a későbbi lapozási struktúrába szervezést engedélyezi (On) vagy tiltja (Off). Lásd **\$O+/-** direktíva.

**Options/Compiler/Align data**

Az adatok szó- (Word) vagy bájtárra (Byte) való elhelyezését vezérli. Lásd \$A+/- direktíva.

**Options/Compiler/Var-string checking**

Karakterlánc típusú paraméterek esetén a típusellenőrzést vezérli (Strict – szoros, Relaxed – laza). Lásd \$V+/- direktíva.

**Options/Compiler/Boolean evaluation**

A feltételek kiértékelési módját határozza meg: teljes (Complete) vagy gyors (Short circuit). Lásd \$B+/- direktíva.

**Options/Compiler/Numeric processing**

Hardver (Hardware) vagy szoftver (Software) valós számítási módot választ ki. Lásd \$N+/- direktíva.

**Options/Compiler/Emulation**

Hardver (80x87) valós számítási mód esetén az emulátort engedélyezi (On) vagy tiltja (Off). Lásd \$E+/- direktíva.

**Options/Compiler/Debug information**

Hibakereséshez/nyomkövetéshez szükséges információk tábla generálását engedélyezi (On) vagy tiltja (Off). Lásd \$D+/- direktíva.

**Options/Compiler/Local symbols**

Hibakereséshez/nyomkövetéshez szükséges lokális információk generálását engedélyezi (On) vagy tiltja (Off). Lásd \$L+/- direktíva.

## MENÜRENDSZER

---

### Options/Compiler/Conditional defines

Feltételes szimbólumok definiálása. Lásd \$DEFINE direktíva és előre definiált szimbólumok.

### Options/Compiler/Memory sizes

A program memóriagényét meghatározó jellemzők beállítása: Stack size – veremméret (1024..65520); Low heap limit – heapméret alsó határának minimuma (0..655360); High heap limit – heapméret felső határának maximuma (heapmin..655360). Lásd \$M direktíva.

### Options/Linker/Map file

A térképállomány opciói választhatók ki: Off – nem készül térképállomány; Segments – szegmensek, Publics – szegmensek és globális azonosítók, Detailed – szegmensek, globális azonosítók és forrássorok adatai kerülnek be a térképállományba.

### Options/Linker/Link buffer

A program összeszerkesztése lemezen (Disk) vagy memóriában (Memory). Utóbbi esetben a szerkesztés gyorsabb, de több memóriát igényel.

### Options/Environment/Config auto save

On állapotban az opciókat – ha bármelyik módosult – a rendszer automatikusan elmenti kilépéskor és futtatás előtt a konfigurációs állományba. Ennek nevét az O/Save options pontban, a Turbo Pascal keretrendszer indításakor a /Cállománynév paraméterrel, vagy installáláskor adhatjuk meg. Alapértelmezése TURBO.TP.

## Options/Environment/Edit auto save

On állapotban a szövegszerkesztőben lévő szöveget – ha az módosult – automatikusan elmenti kilépéskor és futtatáskor.

## Options/Environment/Backup files

On állapotban mentéskor az eredeti szöveget tartalmazó állomány kiterjesztését átírja .BAK-ra, majd .PAS állományba ment. A módosítás előtti szöveg így a .BAK állományban megmarad.

## Options/Environment/Tab size

Tabulációs lépésköz beállítása. Ha a szövegszerkesztő nem hardver, hanem automatikus tabulálás üzemmódban dolgozik (Tab felirat nem látható), nincs hatása. Értéke 2..16 lehet.

## Options/Environment/Zoom windows (F5)

Off állapotban egyszerre látható a szerkesztő- és a kimeneti vagy figyelőablak a képernyőn, míg On állapotban kiterjeszti az aktív ablakot a teljes képernyőre.

## Options/Environment/Screen size

A keretrendszer képernyősoralnak számát lehet beállítani EGA és VGA meghajtók esetében 25, 43 illetve 50 sorra. Más monitorvezérlőkkel csak 25 soros megjelenítés választható.

## Options/Directories/Turbo directory

A TURBO.HLP és TURBO.TP állományok másodlagos katalógusa: ha az aktuális katalógusban nem található, itt keresi még ezeket a keretrendszer.

## MENÜRENDSZER

---

### Options/Directories/EXE & TPU directory

Az itt megadott katalógusban helyez el a rendszer a lefordított programokat (.EXE állományok) és egységeket (.TPU állományok), egyébként az aktuális katalógusban.

### Options/Directories/Include directories

A \$I direktíva után megadott befoglalandó forrásállományok másodlagos katalógusai (mint a DOS PATH parancsa): ha az aktuális katalógusban nem található, ezekben keresheti még a rendszer.

### Options/Directories/Unit directories

A lefordított egységek (.TPU állományok) másodlagos katalógusai.

### Options/Directories/Object directories

Az \$L direktíva után megadott tárgykódú (.OBJ) állományok másodlagos katalógusai.

### Options/Directories/Pick file name

Az itt megadott illetve az O/D/Current pick file sorban látható állapotállományban rögzíti kilépéskor a rendszer az állapotlistát.

### Options/Directories/Current pick file

Információs sor: az aktuális állapotállomány nevét tartalmazza. Ez az O/D/Pick file name ponttal vagy installáláskor adható meg. Alapértelmezése TURBO.PCK.

### Options/Parameters

A keretrendszerből futtatott program parancssor-paraméterrel.

---

### Options/Save options

A beállított opciók elmentése a felkínált vagy az itt megadott konfigurációs állományba.

### Options/Retrieve options

Opciók betöltése az itt megadott konfigurációs állományból.

### Debug/Evaluate (CTRL-F4)

Kifejezés vagy változó kiértékelése, változó értékének megváltoztatása hibakeresés/nyomkövetés közben: **Evaluate** – kiértékelendő kifejezés megadása (a kifejezésben használható függvényeket lásd a **B/Add watch** funkcionál), **Result** – kifejezés értékének kijelzése, **New value** – változó esetén új érték megadása.

### Debug/Call stack (CTRL-F3)

Nyomkövetés közben a szubrutinverem megjelenítése az esetleges paraméterekkel együtt. Az ablak első sorában látható az aktuális eljárás, függvény vagy metódus<sup>5.5</sup>. Egyidejűleg maximálisan kilenc sor látható. A szövegszerkesztőbe betölthető a kurzorvezérlő billentyűkkel kiválasztott szubrutint tartalmazó forrásnyelvi állomány az ENTER leütésével, de a program az aktuális megszakítási ponttól folytatódik majd.

### Debug/Find procedure

Nyomkövetés közben az itt megadott szubrutint tartalmazó forrásállományt betölti a rendszer a szövegszerkesztőbe, majd a kurzort a szubrutin elejére állítja. A program az aktuális megszakítási ponttól folytatódik majd.

## MENÜRENDSZER

---

### Debug/Integrated debugging

A beépített hibakereső/nyomkövető engedélyezése vagy tiltása. Ha Off állapotban van, nem lehet a programot lépésenként futtatni, a változóit tartalmazó kifejezéseket kiértékelni, illetve a futási hiba helyét megkeresni.

### Debug/Stand-alone debugging

On állapotában kiegészítő információkat is elhelyez az .EXE állományban, ha az O/C/Debug information kapcsoló is On állapotban van. Így a Turbo Debugger külső hibakeresővel is követhető lesz a program. A C/Destination pont Memory állapota mellett nincs hatása.

### Debug/Display swapping

A képernyőváltás módját állítja be: **None** – nincs képernyőváltás kírás előtt (ekkor programunk a Turbo Pascal rendszer képernyőjére ír!), **Smart** – képernyőváltás csak a kírások idejére, **Always** – képernyőváltás minden utasítás-végrehajtáskor. Ha a keretrendszert kettős monitor üzemmódban használjuk (/D parancssor-paraméterrel indítottuk és van a gépben két monitorvezérlő), nincs hatása.

### Debug/Refresh display

A keretrendszer képernyőjének visszaállítása, frissítése (például ha a D/Display swapping menüpont **None** állapota mellett arra ráértünk).

### Break/watch/Add watch (CTRL-F7)

Új értékkövető (figyelendő kifejezés) beszúrása az aktuális elé, melyet megkülönböztetetten jelez a rendszer. A kifejezésben használható függvények és előre deklarált tömbök:



Abs	DSeg	Lo	MemL	Ord	Seg	Succ
Addr	HI	MaxAvall	MemW	Pred	SizeOf	Swap
Chr	IOResult	Mem	Odd	Ptr	SPtr	Trunc
CSeg	Length	MemAvall	Ofs	Round	SSeg	TypeOf <sup>5.5</sup>

Más függvények csak önállóan szerepelhetnek – ekkor azok memórlabe-  
ll címe jelenik meg. A TypeOf függvény csak VMT-vel rendelkező objek-  
tumokra használható<sup>5.5</sup>.

A figyelendő kifejezés után vesszővel elválasztva opcionálisan az ismétlé-  
si tényező, majd közvetlenül egymást követően az alábbi formátumjelzők  
használhatók:

\$,H, X	hexadecimális megjelenítés
C	#xx helyett a vezérlőkarakter képe
D	decimális megjelenítés
F <i>n</i>	<i>n</i> számjegyen történő valós megjelenítés; <i>n</i> =2..18
M	memórlabe'll bájtok megjelenítése érték helyett
P	Ptr(\$ssss:\$oooo) helyett ssss:oooo klírása
R	rekordok esetében a mezőazonosítók klírása is
S	M opcióval: karakterláncokban a vezérlőkaraktereket #xx formá- ban írja kl.

Ismétlési tényező megadása esetén a hivatkozás típusa szerinti *n* számú,  
egymást követő érték jelenik meg. Objektumoknak csak az adatmezői lát-  
hatók a figyelőablakban<sup>5.5</sup>. A program WITH utasítással hatástalanok a fi-  
gyelendő kifejezések.

A funkció kiváltható az INS billentyűvel is, ha a figyelőablak aktív.

## MENÜRENDSZER

---

### **Break/watch/Delete watch**

Az aktuális értékkövető törlése a figyelőablakból. Kiváltható a DEL billentyűvel is, ha a figyelőablak aktív.

### **Break/watch/Edit watch**

Az aktuális értékkövető szerkesztése, módosítása (lásd **B/Add watch** funkció). Kiváltható a RETURN billentyűvel is, ha a figyelőablak aktív.

### **Break/watch/Remove all watches**

Az összes értékkövető törlése a figyelőablakból.

### **Break/watch/Toggle breakpoint (CTRL-F8)**

A kurzor sorára töréspont elhelyezése, illetve ha az aktív, megszüntetése. A töréspontokat megkülönböztetett színű sorok jelzik.

### **Break/watch/Clear all breakpoints**

Az összes töréspont megszüntetése.

### **Break/watch/View next breakpoint**

A kurzort a következő töréspontot tartalmazó sorra viszi. Végrehajtás nem történik, a futási sorjelző nem változik.

<b>F1</b>	segítség kérése a képernyőre (Help)
<b>F2</b>	szöveg mentése (F/Save)
<b>F3</b>	szöveg betöltése (F/Load)
<b>F4</b>	futtatás a kurzor sorálg (R/Go to cursor)
<b>F6</b>	aktív ablak kicsi ↔ nagy (O/E/Zoom windows)
<b>F6</b>	szerkesztő ↔ figyelő/kimeneti ablak (Switch)
<b>F7</b>	lépésenkénti futtatás (R/Trace into)
<b>F8</b>	lépésenkénti futtatás, szubrutin egyben (R/Step over)
<b>F9</b>	dátum szerinti újrafordítás (C/Make)
<b>F10</b>	menürendszer → aktív ablak → főmenü
<b>ALT-F1</b>	előző segítség ablak kérése (Last help)
<b>ALT-F3</b>	betöltés állapotlístából (F/Pick)
<b>ALT-F6</b>	kimeneti képernyő (R/User screen)
<b>ALT-F6</b>	aktuális ↔ előző forrásállomány illetve figyelő ↔ kimeneti ablak (Swap)
<b>ALT-F9</b>	fordítás (C/Compile)
<b>CTRL-F1</b>	segítség a kurzor feletti szóhoz
<b>CTRL-F2</b>	program inicializálása (R/Program reset)
<b>CTRL-F3</b>	szubrutinverem megjelenítése (D/Call stack)
<b>CTRL-F4</b>	kifejezés kiértékelése (D/Evaluate)
<b>CTRL-F7</b>	új értékkövető beszúrása (B/Add watch)
<b>CTRL-F8</b>	töréspont be/ki (B/Toggle breakpoint)
<b>CTRL-F9</b>	a program (tovább-)futtatása (R/Run)
<b>SHIFT-F10</b>	a fordító verzióablakának megjelenítése

## FORRÓBILLENTYŰK

---

ALT-F	File
ALT-E	Edit
ALT-R	Run
ALT-C	Compile
ALT-O	Options
ALT-D	Debug
ALT-B	Break/watch
ALT-X	Exit

### CTRL-BREAK

Program megszakítása. Amennyiben az esetlegesen használt Crt egység CheckBreak változója True értékű, a program mindig megszakítható (ez az alaphelyzet), míg ha False, a Ctrl-Break leütése hatástalan. A keretrendszerből indított programok – a változó értékétől függetlenül – mindig megszakíthatók.

### ESC

Kilépés az aktuális menüből. Amennyiben egy almenü aktív, az előző menüsintre lép a rendszer, míg a főmenüből az aktív ablakba jutunk.

### KURZORVEZÉRLŐ BILLENTYŰK

Ha a kimeneti ablak aktív: az ablak tartalmának pásztázása; ha a figyelőablak aktív: aktuális értékkövető kiválasztása, aktuális sor balra illetve jobbra léptetése.

## Kurzormozgatás

CTRL-S, ←	egy karakterrel balra
CTRL-D, →	egy karakterrel jobbra
CTRL-A, CTRL- ←	egy szóval balra
CTRL-F, CTRL- →	egy szóval jobbra
CTRL-E, ↑	egy sorral fel
CTRL-X, ↓	egy sorral le
CTRL-W	pergetés fel
CTRL-Z	pergetés le
CTRL-R, PgUp	egy lappal fel
CTRL-C, PgDn	egy lappal le
CTRL-Q S, Home	sor elejére
CTRL-Q D, End	sor végére
CTRL-Q E, CTRL-Home	képernyő elejére
CTRL-Q X, CTRL-End	képernyő végére
CTRL-Q R, CTRL-PgUp	szöveg elejére
CTRL-Q C, CTRL-PgDn	szöveg végére
CTRL-Q B	blokk elejére
CTRL-Q K	blokk végére
CTRL-Q P	előző kurzor pozícióra
CTRL-Q W	utolsó hibahelyre
CTRL-Q 0   1   2   3	megadott jelzőre

# A SZÖVEGSZERKESZTŐ PARANCSAI

---

## Beszúrás/törlés

CTRL-V, Ins	beszúró (Insert mode) ↔ felülíró mód
CTRL-N	sor beszúrása
CTRL-Y	sor törlése (véglegesen)
CTRL-Q Y	sor végéig törlés
CTRL-H, BackSpace	kurzor előtti karakter törlése
CTRL-G, Del	kurzor feletti karakter törlése
CTRL-T	szó törlése

## Blokk-kezelés

CTRL-K B	blokk elejének kiválasztása
CTRL-K K	blokk végének kiválasztása
CTRL-K T	szó kiválasztása blokk-ként
CTRL-K P	blokk/teljes szöveg nyomtatása
CTRL-K C	blokk másolása
CTRL-K Y	blokk törlése (véglegesen)
CTRL-K H	blokk látható/rejtett
CTRL-K V	blokk átmozgatása
CTRL-K R	blokk betöltése lemezről
CTRL-K W	blokk mentése lemezre
CTRL-K I	blokk egy karakterrel jobbra
CTRL-K U	blokk egy karakterrel balra

# A SZÖVEGSZERKESZTŐ PARANCSAI

## Egyéb funkciók

**CTRL-U**

művelet megszakítása

**CTRL-O I, CTRL-Q I**

eltolás soremelésnél be/ki (Indent)

**CTRL-O U**

eltolás visszatörlésnél be/ki (unIndent)

**CTRL-P**

vezérlőkarakter prefix

**CTRL-K D, CTRL-K Q**

szövegszerkesztő → főmenü (= F10)

**CTRL-K S**

mentés (= F2)

**CTRL-Q F**

keresés

**CTRL-Q A**

keresés és csere

**CTRL-L**

előző keresés (és csere) ismétlése

**Keresés, keresés és csere opciói:**

**B** visszafelé

**G** a teljes szövegben

**U** kisbetű = nagybetű

**W** csak önálló szavakra

**L** a kiválasztott blokkban

**N** csere visszakérdezés nélkül

*n* *n*-szer, *n* pozitív egész

**CTRL-K 0 | 1 | 2 | 3**

jelző elhelyezése

**CTRL-O O**

az O/Compiler menü szerinti aktuális direktívák beszurása a szöveg elé

## A SZÖVEGSZERKESZTŐ PARANCSAI

CTRL-O F	optimális feltöltés be/ki
CTRL-Q [	[ párjának megkeresése
CTRL-Q ]	] párjának megkeresése
CTRL-Q L	a módosított sor visszaállítása
CTRL-I, TAB	tabulálás
CTRL-O T, CTRL-Q T	tabulációs mód váltása

### A szövegszerkesztő opciói (a szerkesztő ablak felső sora)

- **Insert** – beszúró üzemmód (Insert mode)
- **Indent** – automatikus eltolás (Autoindent mode)  
Soremelés után a kurzor nem az első, hanem az előző sor első nem szóköz karakterének oszlopára áll.
- **Tab** – tabuláció (Use tabs)  
Ha aktív (hardver tabulálás), tabuláláskor az **O/E/Tab size** menüpontban megadott méret szerinti TAB karaktert szúr be a szövegbe, egyébként (szoftver tabulálás) a kurzor az előző nem üres sor következő szavának oszlopára áll – szóközők beszúrásával.
- **Fill** – optimális feltöltés (Optimal fill)  
Csak hardver tabulálás esetén van hatása: automatikusan eltolt sorok elejét optimálisan cseréli le TAB és szóköz karakterekre.
- **Unindent** – automatikus eltolás visszatörlésnél (Backspace unindents)  
Ha a kurzor a sor első nem szóköz karakterén áll, akkor a BACKSPACE nem egy szóközt töröl vissza, hanem a kurzor felett álló legközelebbi olyan sor első nem szóköz karakterének oszlopáig, melyben a kurzortól balra van nem szóköz karakter.
- \* karakter a szerkesztő állapotsorában az állománynév előtt  
Megjelenik, ha a szöveg módosult az előző mentés vagy betöltés óta.



## Definíciók

## • A Pascal program elemei

Szimbólumok, fenntartott szavak, azonosítók, címkék, számkonstan-  
sok, szövegkonstansok és elválasztó jelek (egy vagy több szóköz illet-  
ve megjegyzés, soremelés). Elválasztójel nem lehet elem része – kivé-  
tel a szövegkonstans.

## • Szimbólumok

A..Z,a..z,0..9

0..9,A..F,a..f

+ - \* / = [ ] . , ( ) ; ; ^ @ { } \$ #

&lt; = &gt; &lt; &gt; := .. ( \* ) ( . )

nagybetűk, kisbetűk, számjegyek  
hexadecimális számjegyek – \$ után  
egykarakteres szimbólumok  
kétkarakteres szimbólumok

## • Fenntartott Szavak

ABSOLUTE

AND

ARRAY

BEGIN

CASE

CONST

CONSTRUCTOR<sup>5.5</sup>DESTRUCTOR<sup>5.5</sup>

DIV

DO

DOWNT0

ELSE

END

EXTERNAL

FILE

FOR

FORWARD

FUNCTION

GOTO

IF

IMPLEMENTATION

IN

INLINE

INTERFACE

INTERRUPT

LABEL

MOD

NIL

NOT

OBJECT<sup>5.5</sup>

OF

OR

PACKED

PROCEDURE

PROGRAM

RECORD

REPEAT

SET

SHL

SHR

STRING

THEN

TO

TYPE

UNIT

UNTIL

USES

VAR

VIRTUAL<sup>5.5</sup>

WHILE

WITH

XOR

- **Programsor**

A program elemelt sorokba írjuk; maximálisan 126 karakter hosszú lehet egy sor.

- **Megjegyzés**

Korlátlan hosszúságú szöveg, bárhány programsoron keresztül { } vagy (\* \*) között.

- **Direktíva** – speciális megjegyzés

Fordítónak szóló utasítás {\$ vagy (\*\$ után; valamint speciális direktívák.

- **Azonosító**

Programot, egységet, eljárást, függvényt, metódust<sup>5.5</sup>, konstanst, típust, változót, rekordmezőt és objektummezőt<sup>5.5</sup> azonosít. 63 szignifikáns karakter: kisbetűk, nagybetűk, számjegyek, aláhúzás karakter; számjeggyel nem kezdődhet; a kis- és nagybetűket nem különbözteti meg a fordító.

- **Minősített azonosító**

*programazonosító.azonosító | egységazonosító.azonosító*

A program és egység ugyanolyan azonosítóinak megkülönböztetése.

- **Címke**

Számjegysorozat 0 és 9999 között vagy *azonosító*.

- **Számkonstans**

Decimális egész: [+ | -] számjegyek (-2147483648..2147483647);

hexadecimális egész: [+ | -] \$ max. 8 hexadecimális számjegy;

valós ( $m \cdot 10^e$ ): [+ | -] mantissza / E | e / [exponens]] ,

ahol mantissza: [+ | -] számjegyek / .számjegyek/;

exponens: [+ | -] számjegyek

- **Szövegkonstans (karakterlánc)**

Aposztrófok közé írt ASCII karakterek – kivéve az aposztróf –, és ket-tős aposztróf – ez utóbbl egyet reprezentál. *\*karakter* (kontrol karakter, ahol *karakter* kódja 0..31 lehet) és *#ASCII kód* (ASCII kóddal meg-adott karakter) beágyazhatók a láncba – az aposztrófok közé írt szö-veg(ek) elé, mögé és közé.

- **Formális paraméterlista**

$$\left\{ \begin{array}{l} \text{Var azonosító [ , azonosító ... ]} \\ [ \text{Var} ] \text{ azonosító [ , azonosító ... ] : típusazonosító} \end{array} \right\} [ ; \dots ]$$

- **Aktuális paraméterlista**

*Kifejezés | változó [ , kifejezés | változó ... ]*

- **Függvényhívás**

$$\left\{ \begin{array}{l} \text{Függvényazonosító} \\ \text{Függvényváltozó} \\ \text{Objektum-azonosító.függvénymetódus}^{5.5} \end{array} \right\} [(\text{aktuális paraméterlista})]$$

- **Típuskonvertált változó**

*Típusazonosító(változóhivatkozás)*

A *változóhivatkozás* és *típusazonosító* által meghatározott memória-  
méreteknek meg kell egyezniük.

- **Típuskonvertált érték**

*Típusazonosító(kifejezés)*

*Típusazonosító* és *kifejezés* típusa *sorszámozott* vagy *mutató* lehet.

- **Változóhivatkozás**

Változó (a típusos konstans is változó!), struktúrált vagy karakterlánc típusú változó egy komponense, illetve mutató típusú változó vagy

függvényérték által mutatott dinamikus változó:

változó-azonosító típusos konstans azonosítója típuskonvertált változó függvényhívás <sup>^</sup>	}	[ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>index</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>.mezőnév</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <sup>^</sup> </td> <td style="padding: 5px;">}</td> </tr> </table> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="padding: 5px;">...</td> <td style="padding: 5px;">}</td> </tr> </table> </td> <td style="padding: 5px;">]</td> </tr> </table>	<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>index</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>.mezőnév</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <sup>^</sup> </td> <td style="padding: 5px;">}</td> </tr> </table> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="padding: 5px;">...</td> <td style="padding: 5px;">}</td> </tr> </table>	<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>index</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>.mezőnév</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <sup>^</sup> </td> <td style="padding: 5px;">}</td> </tr> </table>	<i>index</i>	}	<i>.mezőnév</i>	}	<sup>^</sup>	}	}	...	}	]
<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>index</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>.mezőnév</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <sup>^</sup> </td> <td style="padding: 5px;">}</td> </tr> </table> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="padding: 5px;">...</td> <td style="padding: 5px;">}</td> </tr> </table>	<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>index</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>.mezőnév</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <sup>^</sup> </td> <td style="padding: 5px;">}</td> </tr> </table>	<i>index</i>	}	<i>.mezőnév</i>	}	<sup>^</sup>	}	}	...	}	]			
<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>index</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>.mezőnév</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <sup>^</sup> </td> <td style="padding: 5px;">}</td> </tr> </table>	<i>index</i>	}	<i>.mezőnév</i>	}	<sup>^</sup>	}	}							
<i>index</i>	}													
<i>.mezőnév</i>	}													
<sup>^</sup>	}													
...	}													

ahol *index*: [kifejezés[,kifejezés...]]

- **Halmazkonstruktor**

[kifejezés[.kifejezés][,kifejezés[.kifejezés]...]]

ahol *kifejezés* típusa sorszámozott, sorszáma a 0..255 Intervallumban lehet.

- **Kifejezés**

Operandusok műveleti jelekkel (operátorokkal) összekapcsolt sorozata. Az operandusoknak a művelettől függően kompatibilis típusúaknak kell lenniük. A kifejezés típusát az operandusok és a művelet(ek) együttesen határozzák meg. (Lásd ADATTÍPUSOK, MŰVELETEK fejezetet!)

Lehetséges operandusok:

[ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>[@] változóhivatkozás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@eljárás-azonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@függvényazonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>konstans</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>(kifejezés)</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>függvényhívás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>halmazkonstruktor</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>típuskonvertált érték</i> </td> <td style="padding: 5px;">}</td> </tr> </table> </td> <td style="padding: 5px;">}</td> </tr> </table> </td> </tr> </table>	<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>[@] változóhivatkozás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@eljárás-azonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@függvényazonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>konstans</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>(kifejezés)</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>függvényhívás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>halmazkonstruktor</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>típuskonvertált érték</i> </td> <td style="padding: 5px;">}</td> </tr> </table> </td> <td style="padding: 5px;">}</td> </tr> </table>	<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>[@] változóhivatkozás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@eljárás-azonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@függvényazonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>konstans</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>(kifejezés)</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>függvényhívás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>halmazkonstruktor</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>típuskonvertált érték</i> </td> <td style="padding: 5px;">}</td> </tr> </table>	<i>[@] változóhivatkozás</i>	}	<i>@eljárás-azonosító</i>	}	<i>@függvényazonosító</i>	}	<i>konstans</i>	}	<i>(kifejezés)</i>	}	<i>függvényhívás</i>	}	<i>halmazkonstruktor</i>	}	<i>típuskonvertált érték</i>	}	}
<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>[@] változóhivatkozás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@eljárás-azonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@függvényazonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>konstans</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>(kifejezés)</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>függvényhívás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>halmazkonstruktor</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>típuskonvertált érték</i> </td> <td style="padding: 5px;">}</td> </tr> </table> </td> <td style="padding: 5px;">}</td> </tr> </table>	<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>[@] változóhivatkozás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@eljárás-azonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@függvényazonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>konstans</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>(kifejezés)</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>függvényhívás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>halmazkonstruktor</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>típuskonvertált érték</i> </td> <td style="padding: 5px;">}</td> </tr> </table>	<i>[@] változóhivatkozás</i>	}	<i>@eljárás-azonosító</i>	}	<i>@függvényazonosító</i>	}	<i>konstans</i>	}	<i>(kifejezés)</i>	}	<i>függvényhívás</i>	}	<i>halmazkonstruktor</i>	}	<i>típuskonvertált érték</i>	}	}	
<table border="0"> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>[@] változóhivatkozás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@eljárás-azonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>@függvényazonosító</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>konstans</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>(kifejezés)</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>függvényhívás</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>halmazkonstruktor</i> </td> <td style="padding: 5px;">}</td> </tr> <tr> <td style="border-left: 1px solid black; padding: 5px;"> <i>típuskonvertált érték</i> </td> <td style="padding: 5px;">}</td> </tr> </table>	<i>[@] változóhivatkozás</i>	}	<i>@eljárás-azonosító</i>	}	<i>@függvényazonosító</i>	}	<i>konstans</i>	}	<i>(kifejezés)</i>	}	<i>függvényhívás</i>	}	<i>halmazkonstruktor</i>	}	<i>típuskonvertált érték</i>	}	}		
<i>[@] változóhivatkozás</i>	}																		
<i>@eljárás-azonosító</i>	}																		
<i>@függvényazonosító</i>	}																		
<i>konstans</i>	}																		
<i>(kifejezés)</i>	}																		
<i>függvényhívás</i>	}																		
<i>halmazkonstruktor</i>	}																		
<i>típuskonvertált érték</i>	}																		

- **Konstans kifejezés**

Olyan kifejezés, melyet a fordító ki tud értékelni. Konstans kifejezésben az alábbi függvények szerepelhetnek:

Abs	Length	Ord	Round	Swap
Chr	Lo	Pred	SizeOf	Trunc
Hi	Odd	Ptr	Succ	

I Konstans kifejezésben tilos változóra (és típusos konstansra) hivatkozni, valamint nem használható a @ operátor.

- **Feltétel**

Logikai (Boolean típusú) kifejezés.

- **Utasítás**

[*címke* :] egyszerű utasítás | struktúrált utasítás

## Egyszerű utasítások

- Értékadás : *változó*hivatkozás | függvényazonosító := kifejezés
- Eljáráshívás : *eljárás*-azonosító | *eljárás*változó  
[ ( aktuális paraméterlista ) ]
- Metódushívás<sup>5.5</sup>: *objekum*-azonosító.*metódus*-azonosító  
[ ( aktuális paraméterlista ) ]  
ahol *metódus*-azonosító konstruktort, destruktort  
vagy eljárást azonosít
- Vezérlésátadás : GOTO *címke*
- Gépi kódú betét : **INLINE**(*elem* [ / *elem* ... ]) (mint **INLINE** direktívánál)

## Struktúrált utasítások

- Összetett utasítás:

**BEGIN**

*[ utasítás ; utasítás ... ]]*

**END**

- Elágazás:

**IF** *feltétel* **THEN** *utasítás*

*[ ELSE utasítás ]*

- Választás:

**CASE** *szelektor* **OF**

*állandó [ .. állandó ] [ , állandó [ .. állandó ] ... ] : utasítás ;*

*[ állandó [ .. állandó ] [ , állandó [ .. állandó ] ... ] : utasítás ; ... ]*

*[ ELSE utasítás ]*

**END**

ahol *szelektor* sorszámozott kifejezés; *állandó* konstans kifejezés, mely sorszámának az Integer (Integer vagy Word<sup>5.5</sup>) intervallumába kell esnie. *Szelektor* és *állandó* kompatibilis típusúak.

- Hátultesztelő ciklus:

**REPEAT**

*[ utasítás [ ; utasítás ... ]]*

**UNTIL** *kilépési feltétel*

- Elöltesztelő ciklus:

**WHILE** *belépési feltétel* **DO**

*utasítás*

- Elöltesztelő léptető ciklus:

**FOR ciklusváltozó** := kezdőérték TO | DOWNTO végérték DO  
utasítás

ahol *ciklusváltozó* lokális sorszámozott típusú változóhivatkozás, *kezdőérték* és *végérték* azzal értékadás-kompatibilis kifejezések. TO esetén *ciklusváltozó* – sorszám szerint – eggyel növekszik, DOWNTO esetén eggyel csökken.

- Minősítő utasítás:

**WITH változó-azonosító** [ , változó-azonosító ... ] DO  
utasítás

ahol *változó-azonosító* rekord vagy objektum<sup>5.5</sup> azonosítója lehet. A felsorolt változók mezőire való hivatkozásnál *utasításban* elegendő csak a mező-azonosítókat illetve metódus-azonosítókat<sup>5.5</sup> megadni.

! Ne használjuk dinamikus változóra, ha a mutatót az utasításon belül megváltoztatjuk!

## A program felépítése

programfej:

```
[ PROGRAM azonosító [ ( periféria [, periféria ... ] ) ] ; ]
[ USES azonosító [, azonosító ... ] ; ]
```

programblokk-deklarációs rész (a deklarációk száma és sorrendje kötetlen):

```
[ LABEL címke [, címke ... ] ; ]
[ TYPE azonosító = típusleírás | típusazonosító ; ... ]
[ CONST azonosító
  [:típusleírás | típusazonosító] =
  {
    konstans kifejezés
    tömb konstans
    rekord konstans
    halmaz konstans
    NIL
  } ; ... ]
[ VAR azonosító :típusleírás | típusazonosító ;
  [ ABSOLUTE szegmens :eltolás | változó-azonosító ; ] ... ]
[ PROCEDURE azonosító [ ( formális paraméterlista ) ] ;
  [ INLINE ] | [ FORWARD ; ] | [ EXTERNAL ; ] | [ INTERRUPT ; ]
  eljárásblokk * ]
[ FUNCTION azonosító [ ( formális paraméterlista ) ]
  :típusazonosító ;
  [ INLINE ] | [ FORWARD ; ] | [ EXTERNAL ; ]
  függvényblokk * ]
```

\* Eljárás és függvényblokk : deklarációs és végrehajtandó rész mint programnál – de záró END után pontosvessző.



programblokk – végrehajtandó rész:

**BEGIN**

*[utasítás [ ; utasítás ... ]]*

**END.**

## A metódus felépítése<sup>5.5</sup>

{	{	<b>CONSTRUCTOR</b> <b>DESTRUCTOR</b> <b>PROCEDURE</b>	} <i>objektum-azonosító.azonosító</i> <i>[ ( formális paraméterlista ) ] ;</i>	}
	{	<b>FUNCTION</b>	} <i>objektum-azonosító.azonosító</i> <i>[ ( formális paraméterlista ) ] :</i> <i>típusazonosító ;</i>	

*[ VIRTUAL ; ]* (konstruktor esetén nem adható meg!)  
*metódusblokk* (mint eljárás- illetve függvényblokk)

### Speciális metódusparaméter: SELF<sup>5.5</sup>

Belső paraméter, mely a metódus hívásakor jön létre automatikusan. 32 bites mutató, mely a VMT-re mutat. Ezzel biztosított, hogy a metódus az őt tartalmazó objektum virtuális metódusait korrekten hívja.

### CONSTRUCTOR, DESTRUCTOR<sup>5.5</sup>

A konstruktor a virtuális metódus(oka)t tartalmazó objektumot inicializálja (felépíti a VMT-t), míg a destruktork felszabadítja az objektumhoz rendelt heap-beli területet – amennyiben az objektum dinamikus.

## Az egység felépítése

egységfej:

**UNIT** *azonosító* ;

egységblokk – deklarációs rész – illesztő rész:

**INTERFACE**

[ **USES** *azonosító* [, *azonosító ...*] ; ]

[ *Globális deklarációk – az egység és az őt használó program illetve más egység számára elérhető azonosítók.* ]

- *Címkék, változók, típusok, állandók.*
- *Eljárás- és függvényfejek (a kifejtő részből kiemelve).*

egységblokk – deklarációs rész – kifejtő rész:

**IMPLEMENTATION**

[ **USES** *azonosító* [, *azonosító ...*] ; ]

[ *Lokális deklarációk* ]

- *Címkék, változók, típusok, állandók, eljárások, függvények.*
- *Illesztő részben deklarált eljárások és függvények kifejtése.*

egységblokk – inicializáló rész (végrehajtás a program első utasítása előtt):

[ **BEGIN**

*utasítás* [ ; *utasítás ...* ] ]

**END.**

## Deklarációk érvényességi köre

A program-, egység-, eljárás-, függvény és metódusblokk<sup>5.5</sup> deklarációs részében deklarált azonosítók és címkék a blokk lokális azonosítói illetve címkéi. Egy blokk deklarációs részében egy azonosító vagy címke csak egyszer szerepelhet, és az a deklarációtól a blokk végéig van érvényben, beleértve a belső – a blokk által tartalmazott – blokkokat is, a következő kivételektől eltekintve:

- nem érvényes abban a belső blokkban, ahol ugyanilyen azonosító illetve címke deklarációja szerepel ;
- mutató típus alaptípusa lehet olyan típus, melyet még nem deklaráltunk, de a deklarációnak még ugyanazon típusdeklaráló részben szerepelnie kell ;
- rekordok különböző szintjein lehetnek ugyanolyan mezőnevek.

Egységeket használó programban illetve egységben érvényben vannak a használt egységek illetző részeinek deklarációi. Ha ugyanaz az azonosító több egységben is szerepel, akkor a minősítés nélküli azonosítóra való hivatkozás mindig a később felsorolt egységben levőt jelenti. Minősített azonosítóval a program és bármely egység bármely azonosítójára hivatkozhatunk.

## Definíciók

• **Sorszámozott típusok**

Minden lehetséges értékhez hozzárendelhető egy sorszám (ORDinal number, rendszám), az értékek sorszámuk szerinti rendezettek. Sorszámozott típusok: az egészek, a Boolean, a Char, a felsorolt és az Intervallum típus. A sorszám: 0,1,2 ... – kivéve az egész típusokat, ahol a sorszám maga a szám.

• **Egyszerű típusok**

Sorszámozott típusok, valós típusok.

• **Struktúrált (összetett) típusok**

ARRAY, RECORD, OBJECT<sup>6,5</sup>, SET és állomány típusok.

• **Mutató típusok**

POINTER, típusos mutató.

• **Eljárás típusok**

PROCEDURE, FUNCTION típus.

! **PACKED** az Array, Record, Set, String előtt használható, de hatástalan.

## Turbo Pascal adattípusok

• **Logikai típus – BOOLEAN**

Értékel: False, True (speciális felsorolt típus)

Konstans: False, True

Tárolás: 1 bájt, False = 0, True = 1

• **Karakter típus – CHAR**

Értékel: bármely kiterjesztett ASCII karakter (kód 0..255 lehet)

Konstans: egy karakteres szövegkonstans – pl. 't', #13, ^G, '9'

Tárolás: 1 bájt, a karakter kódja (rendszáma)

# ADATTÍPUSOK, MŰVELETEK

## • Egész típusok

Típus:	Értékei:	Tárolás:
SHORTINT	-128 .. 127	előjeles, 8 bit
INTEGER	-32768 .. 32767	előjeles, 16 bit
LONGINT	-2147483648 .. 2147483647	előjeles, 32 bit
BYTE	0 .. 255	előjel nélküli, 8 bit
WORD	0 .. 65535	előjel nélküli, 16 bit

Konstans:      decimális és hexadecimális egészek – pl.: 13, \$45, 3768

## • Felsorolt típus – (konstansazonosító[,konstansazonosító...])

Az azonosítók egyediek.  $\text{Ord}(\text{konstansazonosító}) = 0, 1, \dots$

Értékei:      a felsorolásban szereplő azonosítók

Konstans:    a felsorolásban szereplő azonosító

Tárolás:      az azonosító rendszáma alapján; 1 bájt, ha az azonosítók száma kisebb, mint 256, egyébként 2 bájt

## • Intervallum típus – $\text{konstans1} .. \text{konstans2}$ (konstans kifejezések)

Egy már létező sorszámozott típus (eredeti típus) egy intervalluma.  $\text{konstans1} \leq \text{konstans2}$ ;  $\text{konstans1}$  zárójellel nem kezdődhet!

Értékei:       $\geq \text{konstans1}$ ,  $\leq \text{konstans2}$

Konstans:    mint az eredeti típus konstansa

Tárolás:      mint az eredeti típus

## • Valós típusok

Lebegőpontos ábrázolás  $m \cdot 2^e$  ( $m$ =mantissza,  $e$ =exponens). Két számítási mód: szoftver (\$N-, csak REAL) / 80x87 (\$N+, SINGLE, DOUBLE, EXTENDED és COMP is). A 80x87 emulálható (\$E+).



## ADATTÍPUSOK, MŰVELETEK

! *Indextípus* csak sorszámozott lehet – kivéve LONGINT

### • Rekord típus – RECORD

[mezőleírás ;]

[ CASE [szelektormező:] sorszámozott típus OF  
értékleírás : (mezőleírás)

[ ; értékleírás : (mezőleírás) ... ]

END

ahol mezőleírás : *azonosító*[ ,*azonosító*...] : *típus*

[ ; *azonosító*[ ,*azonosító*...] : *típus*...]

értékleírás : *konstans kifejezés* [,*konstans kifejezés* ...]

Ha nem deklaráljuk *szelektormezőt*, akkor az értékleírásnak csak szintaktikai szerepe van. Egy rekordnak csak egy változó része lehet, a fix rész mögött.

Konstans: csak CONST-ban:

(*mezőazonosító* : *konstans kifejezés*

[ ; *mezőazonosító* : *konstans kifejezés* ...])

Tárolás: mezők deklarálási sorrendjében. Egy rekord változó részének mezőlistái egyazon memóriacímen kezdődnek; méret: [fix rész +] [szelektormező +] [változó rész leg-hosszabb ága]

Hivatkozás egy mezőre: *rekordazonosító* ,*mezőazonosító*  
*rekordazonosító* elhagyható – lásd WITH utasítás

### • Objektum típus – OBJECT [ ( *objektum típusazonosító* ) ]<sup>5.5</sup> [*adatmezők leírása* (mint RECORD-nál)] *metódusfej* [ *metódusfej* ... ]

END

Hivatkozás egy mezőre: *objektumváltozó.mezőazonosító*  
 metódusra: *objektumváltozó.metódusazonosító*

- | A metódust külön deklarálni kell, az objektumban megadott metódusfejből az objektum típusazonosítója és a pont elhagyandó.
- | Objektum csak a program vagy egység legfelső szintjén deklarálható, a TYPE részben.

## • Halmaz típus – SET OF *alaptípus*

*Alaptípus* csak sorszámozott lehet. Az elemek sorszámainak 0 és 255 közé kell esniük. Egy halmaznak maximálisan 256 eleme lehet.

Érték: *alaptípus* típusú értékek (elemek) összessége; értékadás például halmazkonstruktorral

Konstans: [*elem*[..*elem*]][*elem*[..*elem*] ... ]], üres halmaz: [ ] ahol *elem alaptípus* típusú konstans kifejezés (CONST-ban is) – pl.: ['a'..'z','0'..'9',Chr(32)]

Tárolás: 1 elem = 1 bit (0 – nem eleme, 1 – eleme a halmaznak)  
 Ha *alaptípus* Min..Max, és E egy elem rendszáma, akkor:  
 méret:  $\text{Max div } 8 - \text{Min div } 8 + 1$  bájt  
 az elem az  $(E \text{ div } 8 - \text{Min div } 8)$ . bájt  $(E \text{ mod } 8)$ . bitje

- | Amennyiben egy értékintervallum felső határa kisebb, mint az alsó, az Intervallum nem lesz a halmazban.

## • Típusos állomány – FILE OF *alaptípus*

Komponensei egyforma méretűek és típusúak. A komponensek sorszáma (pozíciója) = 0,1,... Elérés szekvenciálisan vagy a komponens sorszáma szerint direkt. Az állomány írható és olvasható is.

- | Komponens nem lehet állomány és objektum<sup>5.5</sup> típusú.



## ADATTÍPUSOK, MŰVELETEK

---

- **Szöveges állomány – TEXT**

Soros szervezés, a sorok változó hosszúságúak. A sorokat sorvégjel (CR/LF), az állományt állományvégjel (^Z) zárja. Elérés csak szekvenciálisan; az állományt vagy csak olvasni, vagy csak írni lehet.

- **Típus nélküli állomány – FILE**

Alacsony szintű, puffer nélküli B/K; komponenshossz az állomány nyitáskor megadható (alapértéke 128). Elérés szekvenciálisan vagy a komponens sorszama szerint direkt. A komponensek sorszama (pozíciója) = 0,1,... Az állomány írható és olvasható is.

Tárolás:            típusos és típus nélküli állomány: 128 bájt (FileRec), szöveges állomány: 256 bájt (TextRec). Lásd DOS egység.

- **Típusos mutató – *\*alaptípus azonosító***

4 bájtos memóriacímét tartalmaz: *alaptípus* típusú változóra mutat a heap-ben. *Alaptípus* típusú változót a program futása közben hozhatjuk létre.

Értékadás:        New, @ , Ptr, GetMem, :=

Konstans:        NIL (nem mutat sehová)

Tárolás:         dupla szó (szegmens:eltolás) , NIL = Ptr(0:0)

Hivatkozás      a mutatott dinamikus változóra: *mutató-azonosító*<sup>A</sup>

- **Típus nélküli mutató – POINTER**

4 bájtos memóriacímét tartalmaz, arra mutat.

Értékadás:        GetMem, @ , Ptr, :=

Konstans:        NIL (nem mutat sehová)

Tárolás:         dupla szó (szegmens:eltolás) , NIL = Ptr(0:0)

Hivatkozás      a mutatott memóriacímre: *mutató-azonosító*<sup>A</sup>

- ! *mutató-azonosító*<sup>^</sup> –nak nincs típusa, így csak a típust nem igénylő műveleteknél, paraméter-átadásnál használható.
- **Eljárás** – **PROCEDURE** [ (*formális paraméterlista*) ]
- **Függvény** – **FUNCTION** [ (*formális paraméterlista*) ]  
: *típusazonosító*
- ! *Típusazonosító* csak sorszámozott, valós, STRING vagy POINTER lehet.
- Érték: eljárás illetve függvény azonosítója
- Tárolás: dupla szó (szegmens:eltolás) – a szubrutin címe
- Hivatkozás eljárásra, függvényre:  
*azonosító[(aktuális paraméterlista)]*
- ! A paraméterlista azonosítóinak csak szintaktikai szerepük van.
- ! Azon eljárások és függvények, melyek azonosítóját ilyen típusú változó kapja, \$F+ mellett fordítandók; nem lehetnek szabványosak, beágyazottak, valamint nem deklarálatók INLINE és INTERRUPT direktívával.

## Speciális tömbök

Direkt memória- és portelérést szolgáló előre deklarált tömbök:

<b>MEM</b> [ <i>szegmens:eltolás</i> ]	minden eleme BYTE típusú
<b>MEMW</b> [ <i>szegmens:eltolás</i> ]	minden eleme WORD típusú
<b>MEML</b> [ <i>szegmens:eltolás</i> ]	minden eleme LONGINT típusú
<b>PORT</b> [ <i>portcím</i> ]	minden eleme BYTE típusú
<b>PORTW</b> [ <i>portcím</i> ]	minden eleme WORD típusú

# ADATTÍPUSOK, MŰVELETEK

---

## Típusazonosság:

Két típus akkor, és csak akkor azonos, ha:

- azonosítójuk ugyanaz
- az egyik típust a másik azonosítójával deklaráltuk

Változó paraméter esetén az aktuális és formális paraméter típusának azonosnak kell lennie.

## Típus-kompatibilitás:

Két típus kompatibilis, ha az alábbiak egyike igaz:

- a két típus azonos vagy leírása ugyanaz
- mindkettő valós vagy mindkettő egész
- az egyik a másik, vagy mindkettő egy harmadik intervalluma
- mindkettő halmaz, kompatibilis alaptípusokkal
- mindkettő karaktertömb, egyenlő számú komponensekkel
- az egyik karakterlánc, a másik karakter, karakterlánc vagy karaktertömb típusú
- az egyik típusnélküli, a másik bármilyen típusú mutató
- mindkettő eljárás vagy függvény típusú, azonos paraméterszámmal, paramétertípusokkal és függvények esetén azonos eredménytípussal

A típus-kompatibilitás kifejezések kiértékelésénél szükséges.

## Értékadás-kompatibilitás:

T2 értékadási szempontból kompatibilis T1-el (vagyis megengedett a *T1 típusú változó := T2 típusú kifejezés*), ha az alábbiak egyike igaz:

- T1 és T2 azonos típusok, de egyik sem állomány, illetve azt tartalmazó strukturált típus
- T1 és T2 kompatibilis sorszámozott vagy valós típusok, és T2 érték-

készlete beleesik T1 értékkészletébe

- T1 valós, T2 egész típus
- T1 karakterlánc, T2 karakter vagy karakterlánc típus
- T1 karakterlánc, T2 karaktertömb
- T1 és T2 kompatibilis karaktertömbök
- T1 és T2 kompatibilis halmaz típusok, és T2 minden eleme beleesik T1 lehetséges elemek tartományába
- T1 és T2 kompatibilis mutató, eljárás vagy függvény típusok
- T1 eljárás vagy függvény típus, T2 eljárás vagy függvény azonos paraméterszámmal, paramétertípusokkal és függvények esetén azonos eredménytípussal

Értékadásnál a jobb oldali kifejezésnek értékadás-kompatibilisnek kell lennie a változó-hivatkozással illetve függvény-azonosítóval; értékparaméter átadásánál az aktuális paraméter kifejezésnek értékadás-kompatibilisnek kell lennie a formális paraméterrel.

# ADATTÍPUSOK, MŰVELETEK

**Műveletek** (a jobboldali oszlopban lévő betűk jelentését lásd később)

<i>Jel</i>	<i>Művelet leírása</i>	<i>Operandusok</i>	<i>Eredmény</i>
NOT	logikai NEM bítenkénti NEM	Boolean egész	Boolean egész
+, -	előjel	egész valós	egész valós
@	cím	változó   szubrutin	Pointer
*	szorzás	egész bármelyikük valós	egész valós
/	metszet valós osztás	halmaz egész bármelyikük valós	halmaz valós valós
DIV	egész osztás	egész	egész
MOD	maradékképzés	egész	egész
AND	logikai ÉS bítenkénti ÉS	Boolean egész	Boolean egész
x SHL <i>n</i>	eltolás balra <i>n</i> bittel	egész, $n \geq 0$	x típusa
x SHR <i>n</i>	eltolás jobbra <i>n</i> bittel	egész, $n \geq 0$	x típusa
+	összeadás	egész bármelyikük valós	egész valós
	összefűzés	String[ <i>n</i> ] Char	String String
	unió	halmaz	halmaz

## ADATTÍPUSOK, MŰVELETEK

<i>Jel</i>	<i>Művelet leírása</i>	<i>Operandusok</i>	<i>Eredmény</i>
-	klonás	egész bármelyikük valós	egész valós
OR	különbség logikai VAGY	halmaz Boolean	halmaz Boolean
XOR	bítenkénti VAGY logikai KIZÁRÓ VAGY	egész Boolean	egész Boolean
	bítenkénti KIZÁRÓ VAGY	egész	egész
<, <=, >, >=, =, <>	összehasonlítás, tartalmazás-vizsgálat	kompatibilis típusok	Boolean
x IN y	elemvizsgálat	y halmaz, x típusa = y alaptípusa	Boolean

### Magyarázatok a MŰVELETEK táblázathoz:

#### A – aritmetikai művelet

• ha az operandus egész számkonstans, akkor a következő típusú lesz:

-2147483648 .. -32769	LongInt
-32768 .. -1	Integer
0 .. 255	Byte
256 .. 32767	Integer
32768 .. 65535	Word
65536 .. 2147483647	LongInt

## ADATTÍPUSOK, MŰVELETEK

- ha az operandus valós számkonstans, akkor az \$N- esetén Real, \$N+ esetén Extended típusú lesz
- előjel megadása a típust nem változtatja meg
- ha a \*, MOD, DIV, + vagy - műveletek mindkét operandusa egész, akkor az eredmény típusa a következő táblázat szerinti lesz:

	ShortInt	Integer	LongInt	Byte	Word
ShortInt	Integer	Integer	LongInt	Integer	LongInt
Integer	Integer	Integer	LongInt	Integer	LongInt
LongInt	LongInt	LongInt	LongInt	LongInt	LongInt
Byte	Integer	Integer	LongInt	Integer	Word
Word	LongInt	LongInt	LongInt	Word	Word

Intervallum típus esetén az eredmény az eredeti típus szerint alakul.

- ha a \*, + vagy - műveletek valamelyik operandusa valós, illetve / művelet esetén az eredmény \$N- mellett Real, \$N+ mellett Extended típusú lesz

### B – bitenkénti művelet

- a NOT művelet a típust nem változtatja meg
- AND, OR és XOR esetén lásd az előző táblázatot

### C – @ (cím) művelet

- a művelet eredménye Pointer (szegmens:eltolás) típusú lesz

### H – halmaz művelet

- halmaz műveletek (+, -, \*) bal és jobb oldalán kompatibilis halmazok-

nek kell szerepelniük

- ha az eredményhalmaz – illetve halmazkonstans – legkisebb eleme  $e1$ , legnagyobb eleme  $e2$ , akkor annak típusa SET OF  $e1..e2$  lesz

## K – karakterlánc művelet

- a + (összefűző) művelet mindkét oldalán karakter vagy karakterlánc szerepelhet
- az eredmény bármely karakterlánc típusal kompatibilis – de a Char típusal nem
- ha az eredmény 255 karakternél hosszabb lenne, csonkul

## L – logikai művelet

- a klértékelési módokat lásd a \$B+/- direktíva leírásánál

## R – reláció (összehasonlító művelet)

- egyszerű és karakterlánc típusok esetén:
  - a <, <=, >, >=, =, <> műveletek használhatók;
  - az operandusoknak kompatibiliseknek kell lenniük, de lehet az egyik egész, a másik valós;
  - a karakterláncok összehasonlítása az ASCII kódok alapján történik, a Char típus mint String[1] viselkedik ekkor
- mutató típus esetén:
  - csak az = és a <> műveletek használhatók;
  - két mutató akkor egyenlő, ha szegmens- és eltolás részük egyenlők;
  - a New és GetMem eljárások mindig normalizálnak, azaz a mutató eltolás része 0 és \$F között lesz



## ADATTÍPUSOK, MŰVELETEK

- ! két mutató, mely ugyanazon címre mutat, nem feltétlenül egyenlő
- halmaz típusok esetén:  
a  $\leq$ ,  $\geq$ ,  $=$ ,  $\lt$ ,  $\gt$ ,  $\text{IN}$  műveletek használhatók

### Kifejezések kiértékelése

- először a zárójelben szereplő kifejezések kerülnek kiértékelésre
- a kifejezés fordításkor kiértékelhető részelt a fordító értékeli ki, és ezen részeredményeket mint konstansokat helyezi el a programban – az így létrejött részeredmények típusát azok értékeli határozzák meg
- minden részeredmény, valamint az egész kifejezés meghatározott típussal rendelkezik, mely az adott művelettől és az operandusok típusától függ
- a műveletek kiértékelési sorrendjét azok prioritása szabja meg:

1. NOT , + , - , @
2. \* , / , DIV , MOD , AND , SHL , SHR
3. + , - , OR , XOR
4. < , > ,  $\leq$  ,  $\geq$  ,  $\lt$  ,  $\gt$  , = , IN

Az 1. csoportba tartozó műveletek egy operandussal rendelkeznek – ezek prioritása a legnagyobb –, míg a többi műveletnek két operandusa van. Azonos prioritás esetén a balról jobbra szabály lép érvénybe, melyet a fordító felülbírálhat optimális kód készítése érdekében.

- ! Az egyébként azonos prioritású műveletek különbözőképpen való csoportosítása a kifejezés típusát befolyásolhatja; az esetleges értékcsonkulásra a programozónak kell figyelnie – célszerű a konstans kifejezéseket előbbre venni.

## Értékadás, értékparaméter átadása

A kifejezésnek értékadás-kompatibilisnek kell lennie a változó-hivatkozással illetve függvényazonosítóval. Az esetleges túlcsoportolások és túlindexelések, amennyiben a fordító ki tudja szűrni, szintaktikai hibát eredményeznek. Ezek futás közbeni ellenőrzését a \$R+ direktívával tehetjük meg. Az alapértelmezés \$R-, tehát a fogadómezőbe kerülő érték csokulhat, illetve akaratlanul is felülírhatjuk a memória különböző területeit.

A direktívák leírásánál használt jelölések jelentése:

- **G** – globális direktíva: hatása az egész programra kiterjed. A deklarációs rész előtt kell állnia.
- **L** – lokális direktíva: a program bármely pontján elhelyezhető, be- és kikapcsolható.
- A gyári installálásra vonatkozó alapértelmezések **kiemelten** láthatók.

## Kapcsoló- és paraméter direktívák

### **\$A+/-** (O/C/Align data)

**G**

Bekapcsolt állapotban (\$A+) minden egy bájtnál nagyobb méretű változó és típusos konstans szóhatárra (kettővel osztható memórlacfmre), kikapcsolt állapotban (\$A-) bájthatárra kerül. Rekordok, objektumok<sup>5.5</sup> és tömbök elemel mindig bájtfolytonosan helyezkednek el a memóriában. 80x88-as processzor esetén a direktíva hatástalan, 80x86-nál a direktíva aktív állapota gyorsítja a futást.

### **\$B-/+** (O/C/Boolean evaluation)

**L**

Bekapcsolt állapotban (\$B+) a feltételeket teljes egészében kiértékeli a program, függetlenül attól, hogy az egész feltétel szempontjából szükséges-e vagy sem. Kikapcsolt állapotban (\$B-) csak addig értékeli az egyes részfeltételeket, míg egyértelműen el nem dönthető, hogy a teljes feltétel igaz vagy hamis lesz. Inaktív állapotában a futás gyorsabb, de a nem kiértékelt feltételrész esetleges hibái nem derülnek ki, függvényel nem kerülnek végrehajtásra.

## DIREKTÍVÁK

---

### **\$D+/-** (O/C/Debug Information)

**G**

Bekapcsolt állapotban (\$D+) Információs tábla készül, így lehetővé válik az illető egység vagy program keretrendszerből történő hibakeresése/nyomkövetése, valamint a futási hiba helyének meghatározása.

### **\$E+/-** (O/C/Emulation)

**G**

Bekapcsolt állapotban (\$E+) a fordító a programhoz szerkeszti a 80x87-es aritmetikai segédprocesszort emuláló rutincsomagot. Futáskor, ha a 80x87 jelen van, a számításokat az végzi, egyébként a rutincsomag. Egységben a kapcsolónak nincs hatása. A kapcsoló hatástalan \$N- mellett.

### **\$F-/+** (O/C/Force far calls)

**L**

Bekapcsolt állapotban (\$F+) FAR hívási módot használ a fordító: a címek duplaszavasak, így az aktuális kódszegmensen kívülre is lehet hivatkozni. Kikapcsolt állapotban (\$F-) a szükséges - FAR vagy NEAR - hívási módot használja. Overlay egységeket használó programok esetében ajánlott minden egység és a program elejére \$F+ direktívát elhelyezni. Eljárás típusú változókat használó programok \$F+ mellett fordítandók.

### **\$I+/-** (O/C/I/O checking)

**L**

Bekapcsolt állapotban (\$I+) hibás B/K műveletek esetén a program futása hibaüzenet kíséretében leáll. Kikapcsolt állapotban (\$I-) a program B/K hiba miatt nem áll le. Ilyenkor a műveletek hibakódja az IOResult függvénnyel minden esetben lekérdezendő, különben a hibát követő további B/K műveleteket a program nem hajtja végre. A függvény a System egység InOutRes változójának értékét adja vissza, és egyben nullázza is azt.

**\$I állománynév**

L

A fordító a megadott forrásállomány tartalmát befoglalja a direktíva helyére. Ha *állománynév* nem rendelkezik kiterjesztéssel, a fordító automatikusan .PAS-t feltételez; ha nem rendelkezik lemezegységnevvvel és/vagy útvonalleírással, akkor a fordító előbb az aktuális, majd a menürendszer O/D/Include directories pontjában meghatározott katalógusokban keresi. Ezen állományok maximálisan nyolc (5.5 verzióban 15) szint mélységre ágyazhatók egymásba. Begin és End között nem szerepelhet.

**\$L+/- (O/C/Local symbols)**

G

Bekapcsolt állapotban (\$L+) engedélyezi a lokális szimbólumokra vonatkozó információk generálását, és hibakeresés/nyomkövetés során azok kiértékelését. A kapcsoló hatástalan \$D- mellett.

**\$L állománynév**

L

Hozzászerkeszti az egységhez vagy programhoz a megadott tárgykódú (object) állomány tartalmát. Ha *állománynév* nem rendelkezik kiterjesztéssel, a fordító automatikusan .OBJ-t feltételez; ha nincs lemezegységnevv és/vagy útvonalleírás, akkor a fordító előbb az aktuális, majd a menürendszer O/D/Object directories pontjában meghatározott katalógusokban keresi. A tárgykódú állomány minden PUBLIC szubrutinjának fejét EXTERNAL direktívával a Pascal forrásállományban deklarálni kell. A direktíva a deklarációs részben bárhol állhat.

**\$M veremméret, heapmin, heapmax (O/C/Memory sizes)** G

A direktíva a program memóriafoglalását határozza meg. A verem méretét (1024..65520) *veremméret*, a heapméret alsó határának minimumát

## DIREKTÍVÁK

(0..655360) *heapmin*, felső határának maximumát (*heapmin*.. 655360) *heapmax* állítja be. Egységekben hatástalan.

**\$N-/+** (O/C/Numeric processing)

**G**

Bekapcsolt állapotban (\$N+) a valós számításokat a 80x87-es aritmetikai segédprocesszorral végezteti a program, míg kikapcsolt állapotban (\$N-) könyvtári rutinokat használ, és a valós típusok közül csak a Real alkalmazható. Ha a direktíva aktív állapota mellett nincs a gépben 80x87-es segédprocesszor, akkor azt emulálnunk kell (\$E+).

**\$O-/+** (O/C/Overlays allowed)

**G**

Bekapcsolt állapotban (\$O+) az egység későbbi overlay-struktúrába szervezését engedélyezi.

**\$O** *egységnév*

**G**

Overlay-struktúrába szervezi a program által használt, itt megadott egységet, melyet előzőleg \$O+ mellett kell fordítani. A direktívának a programfejtés után kell állnia. Az overlay-struktúrába szervezett egysége(ke)t a fordító a programot tartalmazó forrásállomány nevével megegyező, .OVR kiterjesztésű fizikai állományban helyezi el. Egységekben nincs hatása.

**\$R-/+** (O/C/Range checking)

**L**

Bekapcsolt állapotban (\$R+) engedélyezi értékhatár-ellenőrző kód generálását. Az ellenőrzés tömbök és karakterláncok indexelésére, sorszámozott típusú változók értékadásaira, valamint virtuális metódus hívásakor az objektum előkészítettségének vizsgálatára<sup>5.5</sup> terjed ki. Ha \$R+ mellett hiba lép fel, a program futása hibaüzenet kíséretében leáll. Mivel a program sebességét csökkenti és a kód méretét növeli, csak a program fejleszté-

sekor, tesztelésekor célszerű bekapcsolni.

**\$\$+/-** (O/C/Stack checking)

**L**

Bekapcsolt állapotban (\$\$+) kódot generál a verem túlcserélésének ellenőrzésére. Ha egy szubrutin hívásakor nincs elegendő hely a veremben a visszatérési cím, a paraméterek és a lokális változók számára, \$\$+ esetén a program futása hibaüzenet kíséretében leáll – egyébként a rendszer összeomolhat.

**\$V+/-** (O/C/Var-string checking)

**L**

Bekapcsolt állapotban (\$V+) a karakterlánc típusú formális és aktuális változó (Var) paraméter típusának egyeznie kell. Kikapcsolt állapotban (\$V-) bármilyen karakterlánc típusú aktuális paraméter átadható – természetesen ekkor a programozónak kell gondoskodnia az esetleges szemantikai hibák kivédéséről.

## Feltételes direktívák

**\$DEFINE xxx** (O/C/Conditional defines)

**L**

Feltételes fordításhoz definiálja xxx feltételes szimbólumot. Ha xxx már definiált, a direktívának nincs hatása. A szimbólumok hossza nem korlátozott, de a fordító csak az első 63 karakterüket veszi figyelembe. Kis- és nagybetűket, számjegyeket és \_ karaktert tartalmazhatnak; számjeggyel nem kezdődhetnek.

## DIREKTÍVÁK

---

### **\$UNDEF** *xxx*

**L**

Az előzőleg definiált *xxx* feltételes szimbólumot megszünteti. Ha *xxx* nem definiált - vagy már megszüntetett - nincs hatása a direktívának.

### **\$IFDEF** *xxx*

**L**

Ha *xxx* feltételes szimbólum definiált, a fordító a következő forrásorokat (\$ELSE-ig illetve \$ENDIF-ig) lefordítja.

### **\$IFNDEF** *xxx*

**L**

Ha *xxx* feltételes szimbólum nem definiált - vagy a feltételes szimbólum már megszüntetett -, a fordító a következő forrásorokat (\$ELSE-ig illetve \$ENDIF-ig) lefordítja.

### **\$IFOPT** *d+* | *d-*

**L**

Ha *d* kapcsoló típusú direktíva a megadott állapotban van (+ vagy -), a fordító a következő forrásorokat (\$ELSE-ig illetve \$ENDIF-ig) lefordítja. (Pl.: {\$IFOPT S+})

### **\$ELSE**

\$IFDEF | \$IFNDEF | \$IFOPT és \$ENDIF között alkalmazható. Ha a feltételes fordítást indító direktíva nem teljesül, a \$ELSE és \$ENDIF közötti forrásorokat fordítja le a fordító.

### **\$ENDIF**

Határoolja a feltételes fordításba eső forrásorokat.



## Előre definiált feltételes szimbólumok:

VER50	a fordító verziószáma 5.0 verzióra
VER55 <sup>5.5</sup>	a fordító verziószáma 5.5 verzióra
MSDOS	az operációs rendszer neve (MS-DOS esetén)
CPU86	a processzor családneve (80x86, 80x88 esetén)
CPU87	definiált, ha a gépben fordításkor van 80x87-es segédprocesszor

## Speciális direktívák

## INTERRUPT ;

Megszakítási eljárásokhoz használandó direktíva. Megszakítási eljárást – speciális be- és kilépési utasítássorozata miatt – programból közvetlenül meghívni tilos. Bármely megszakítási vektor (kellő körülményekkel) az eljárás címére átállítható. Az eljárás blokkja a szokásos; formális paraméterlistája a következő: (a lista tagjal tetszőleges ponttól kezdve elhagyhatók) :

(Flags, CS, IP, AX, BX, CX, DX, SI, DI, DS, ES, BP : Word) ;

INLINE ( *elem* [ / *elem* ... ] ) ;

ahol elem:

[ { < } ] { konstans változó } [ + | - *konstans* ... ]

## DIREKTÍVÁK

---

Gépi kódban megírt szubrutin beágyazása a programba. Az utasításokban a program változóknak és konstansoknak azonosítóit is szerepelhetnek. A < operátor az utána álló kifejezés alsó bájttját adja, míg > két bájtot jelent (ha a kifejezés egy bájtt, a felső 00h lesz). A direktíva helyettesíti a szubrutin blokkját. (Az InLine utasítás szerkezete ehhez hasonló.)

### EXTERNAL ;

A szubrutin blokkja egy tárgykódot tartalmazó (.OBJ) állományban található, melyet \$L direktívával a programhoz vagy egységhez kell szerkeszteni bárhol a deklarációs részben. A direktívával deklarált szubrutinhoz a tárgykódú állományban ugyanolyan nevű, PUBLIC típusú rutinnak kell tartoznia.

! A tárgykódú (object) állományban nem állhat olyan PUBLIC címke, melyhez a programban nincs EXTERNAL deklaráció.

### FORWARD ;

A direktívával ellátott szubrutin blokkját nem kell feltétlenül a fej után leírni, megadható a deklarációs rész bármely későbbi pontján is. Ott a blokk előtt csak az esetleges paraméterlista nélkül fejet kell megismételni. Ezzel lehetővé válik, hogy szubrutinok kölcsönösen hívják egymást.

### VIRTUAL ;<sup>5.5</sup>

A direktívával ellátott metódus címe a program futása során lesz ismert. Használni kell, ha azonos nevű metódusok szerepelnek egymásból származó objektumokban. Konstruktor ezt a direktívát nem kaphatja.

## Konstansok

**MaxInt** = 32767 ;

## Változók

Overlay-kezelés:

**OVRCodeList** : Word = 0 ;  
**OVRHeapSize** : Word = 0 ;  
**OVRDebugPtr** : Pointer = Nil ;  
**OVRHeapOrg** : Word = 0 ;  
**OVRHeapPtr** : Word = 0 ;  
**OVRHeapEnd** : Word = 0 ;  
**OVRLoadList** : Word = 0 ;  
**OVRDOSHandle** : Word = 0 ;  
**OVREMSHandle** : Word = 0 ;

Heap-kezelés:

**HeapOrg** : Pointer = Nil ;  
**HeapPtr** : Pointer = Nil ;  
**FreePtr** : Pointer = Nil ;  
**FreeMin** : Word = 0 ;  
**HeapError** : Pointer = Nil ;

Szabványos B/K:

**Input** : Text ;  
**Output** : Text ;

**MaxLongInt** = 2147483647 ;

Elmentett megszakítási vektorok:

**SaveInt00** : Pointer ;  
**SaveInt02** : Pointer ;  
**SaveInt1B** : 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 ;

Hibakezelés:

**ErrorAddr** : Pointer = Nil ;

## SYSTEM EGYSÉG

PSP adatok:

**PrefixSeg** : Word = 0 ;

**StackLimit** : Word = 0 ;

Véletlenszám generálás:

**RandSeed** : Longint = 0 ;

Állománykezelés:

**FileMode** : Byte = 2 ;

80x87 kezelés:

**Test8087** : Byte = 0 ;

B/K hibakezelés:

**InOutRes** : Integer = 0 ;

Kilépés, programvég:

**ExitProc** : Pointer = Nil ;

**ExitCode** : Word = 0 ;

- **PrefixSeg**

A DOS által létrehozott PSP szegmencímét tartalmazza.

- **StackLimit**

SP lehetséges legkisebb értékét – a verem alsó határának eltolási címét – tartalmazza. Értéke a program indulásakor 0, de ha a 80x87-et emulálni kell (\$N+, \$E+ és nincs 80x87-es a gépben), akkor a lebegőpontos számítások munkaterületel miatt 224 lesz. A verem túlcserélését (\$S+ mellett) a program e változó alapján ismeri fel.

- **RandSeed**

A véletlenszám generátor a változó értéke alapján számítja a következő „véletlen” számot. A Randomize eljárás Randseed-nek ad indulási értéket a belső óra alapján. RandSeed értékének beállításával ugyanazt a véletlen mintát többször is előállíthatjuk.

- **FileMode**

Típusos és típus nélküli állományok megnyitási módját meghatározó változó. Lehetséges értékek: 0 – csak olvasható; 1 – csak írható; 2 – ír-

ható/olvasható. A Reset eljárás hívása előtt átállítható, a Rewrite figyelmen kívül hagyja: értéke 2 lesz.

- **Test8087**

\$N+ mellett a 80x87-es aritmetikai segédprocesszor hiányát, illetve ha van, típusát jelzi kódolva: 0 – nincs; 1 – 8087; 2 – 80287; 3 – 80387.

- **InOutRes**

A beépített B/K rutinok a műveletek eredményét *e* változóban helyezik el – az IOResult függvény a változó értékét adja vissza, majd nullázza azt. Ha a változó értéke nem 0, a legközelebbi B/K művelet \$I- esetén nem hajródik végre (még a képernyőre írás sem!), \$I+ mellett a program futási hibával leáll.

- **ErrorAddr, ExitCode**

A program végén tartalmuk a következő: Nil és 0 (illetve a Halt paramétere) – normál befejeződés; hiba címe és kódja futási hiba esetén.

- **ExitProc**

A program – a befejeződés okától független – kilépés előtt végrehajtandó eljárásának címét tartalmazza. Ha értéke Nil, nincs végrehajtandó eljárás a DOS-hoz való visszatérés előtt. A címzett szubrutint a rendszer mindaddig végrehajtja, míg a változó értéke Nil nem lesz. Ezért saját szubrutinban célszerű az eredeti címet elmenteni, majd annak végén visszaállítani. A program eredeti kilépési eljárása lezárja a még nyitott állományokat, visszaállítja a SaveIntXX változóba elmentett megszakítási vektorokat, majd ha ErrorAddr értéke nem Nil, megjeleníti a hiba ExitCode szerinti kódját és a hiba címét.

! Ha a saját kilépési eljárásban hiba lép fel, a program – a változó értékétől függetlenül – befejeződik.

## SYSTEM EGYSÉG

---

Az egység inicializáló része az Input és Output változókat a CON perifériához (billentyűzet és képernyő) rendeli. Az Input állományt olvasásra, az Output állományt írásra megnyitja.

! Amennyiben a program használja a Crt egységet, Input és Output a Turbo Pascal-ban definiált CRT perifériához rendelődik, és a Crt egység inicializáló része nyitja meg azokat.

## Eljárások, függvények

**Abs** ( $x$ ) :  $x$  típusa

Visszaadja  $x$  egész vagy valós kifejezés abszolútértékét.

**Addr** ( $x$ ) : Pointer

Visszaadja  $x$  eljárás, függvény, változó vagy típusos állandó memóriacímét. Helyettesíthető @ $x$ -el.

**Append** (Var F:Text)

Megnyitja írásra az  $F$  szöveges állományt, és az állományvég jelre állítja az aktuális pozíciót.  $F$  állományváltozót előzőleg az Assign eljárással egy létező fizikai állományhoz kell rendelni.

**ArcTan** (X:Real) : Real

$X$  valós kifejezés arkusztangensét adja vissza radiánban.

**Assign** (Var F; Name:String)

$F$  bármilyen állomány típusú változóhoz hozzárendeli  $Name$  fizikai álló-

mányt. *Name* tartalmazhat lemezegységnevet és útvonalleírást is, hossza maximálisan 79 karakter lehet. *Name* azonosíthat DOS perifériát is: CON, LPT1 (=PRN), LPT2, LPT3, COM1 (=AUX), COM2, NUL. Ha *Name* üres karakterlánc, az eljárás *F*-hez a CON perifériát rendeli. Nyitott állományra nem szabad használni.

### **BlockRead** (Var F:File; Var Buf; Count:Word [ ; Var Result:Word ] )

*F* típus nélküli állományból beolvasson *Count* rekordot, és elhelyezze *Buf* változó memóriacímetől kezdődően. *Result* tartalmazza visszatéréskor, hogy hány rekordot olvasott be az eljárás. Ha *Result* hiányzik, és nem tudta a megadott számú rekordot beolvasni, B/K hiba lép fel. A rekordméret 128 bájt, ha az *F* állomány megnyitásakor azt külön nem adtuk meg. *Count* \* rekordméret maximálisan 65535 bájt lehet.

### **BlockWrite** (Var F:File; Var Buf; Count:Word [ ; Var Result:Word ] )

*F* típus nélküli állományba írjon *Count* rekordot *Buf* változó kezdőcímetől. *Result* tartalmazza visszatéréskor, hogy hány rekordot írt ki az eljárás. Ha *Result* hiányzik, és nem tudta a megadott számú rekordot kírni, B/K hiba lép fel. A rekordméret 128 bájt, ha az *F* állomány megnyitásakor azt külön nem adtuk meg. *Count* \* rekordméret nem lehet 65535 bájtnál nagyobb.

### **ChDir** (S:String)

Az aktuális katalógust az *S* paraméterben megadottra változtatja. *S* tartalmazhat lemezegységnevet is. Az eljárás annyiban tér el az operációs rendszer CD parancsától, hogy lemezegység megadása esetén át is vált az illető egységre. A DOS az itt megadott katalógust – és lemezegységet – tekinteli majd aktuálisnak a program lefutása után.

## SYSTEM EGYSÉG

---

### **Chr** (X:Byte) : Char

Az *X* Byte típusú kifejezésnek, mint sorszámnak (ASCII kódnak) megfelelő kiterjesztett ASCII karaktert adja vissza.

### **Close** (Var F)

Lezárja az előzőleg megnyitott *F* állományt és felszabadítja a hozzá tartozó kezelőszámot (DOS handle).

### **Concat** (S1 / ,S2 ...] :String) : String

A paraméterként átadott karakterláncokat összefűzi. Ha a függvény által visszaadott karakterlánc hossza 255-nél nagyobb lenne, akkor a 255. karakter utániakat levágja. A függvény helyettesíthető a + művelettel.

### **Copy** (S:String; Index,Count:Integer) : String

*S* karakterlánc *Index* pozíciójától számított *Count* karakterét mint részláncot adja vissza. Ha *Index* nagyobb, mint az *S* karakterlánc aktuális hossza, a függvény értéke üres karakterlánc lesz. Ha *Count* nagyobb, mint ahány karakter az *Index*-edik pozíció után van, akkor csak a karakterlánc hátralévő karaktereit adja vissza.

### **Cos** (X:Real) : Real

*X* valós kifejezés (radián) koszinuszát adja vissza.

### **C\$eg** : Word

Az aktuális kódszegmens címét (CS regiszter tartalmát) adja vissza.

### **Dec** (Var X / ; N:LongInt / )

*X* sorszámozott típusú változó értékét csökkenti eggyel, illetve *N*-nel. Az



eljárás megfelel az  $X := \text{Pred}(X)$  egyszeri illetve  $N$ -szeri végrehajtásának. Az eljáráshoz a fordító optimalizált (gyors) kódot generál.

**Delete** (Var S:String; Index,Count:Integer)

$S$  karakterlánc típusú változóból törli az  $Index$ -edik karaktertől számított  $Count$  karaktert. Ha  $Index$  nagyobb, mint az  $S$  karakterlánc aktuális hossza, nem töröl. Ha  $Count$  nagyobb, mint ahány karakter az  $Index$ -edik pozíció után van, akkor csak a hátralévő karaktereket vágja le.

**Dispose** (Var P:Pointer [ ; destruktor<sup>5.5</sup> ] )

Felszabadítja a  $P$  típusos mutatóval mutatott heap-beli területet. A felszabadítás után  $P$  definílatlan értéket kap, így minden további hivatkozás  $P$ -ra hibát eredményezhet. Futási hibát okoz, ha hívásakor  $P$  nem a heap-be mutat. Amennyiben  $P$  egy objektum mutatója, másodk paraméterként az objektum destruktorát is átadható, melyet az eljárás végrehajt a felszabadítás előtt<sup>5.5</sup>.

**DSeg** : Word

Az adatszövegmens címét (DS regiszter tartalmát) adja vissza.

**Eof** [ (Var F:Text) ] : Boolean (szöveges állományokra)

A függvény értéke True, ha az aktuális pozíció az állomány logikai végén (^Z karakteren) áll vagy üres az állomány, egyébként False. Ha  $F$  hiányzik, a függvény az input állományt vizsgálja. Amennyiben  $\text{Eof}(F)$  True,  $\text{Eoln}(F)$  is True lesz.

**Eof** (Var F) : Boolean (típusos és típus nélküli állományokra)

A függvény értéke True, ha az aktuális pozíció az  $F$  állomány utolsó kom-

## SYSTEM EGYSÉG

---

ponense mögé került (például az utolsó komponens beolvasása után) vagy üres az állomány, egyébként False.

**Eoln** / (Var F:Text) / : Boolean

A függvény értéke True, ha az aktuális pozíció a szöveges *F* állomány sorvégjelén (CR/LF karakterpár) vagy állományvégjelén (^Z karakter) áll, egyébként False. Ha *F* hiányzik, a függvény az Input állományt vizsgálja.

**Erase** (Var F)

Törli az *F* állományváltozóhoz rendelt fizikai állományt. Megfelel a DOS DEL illetve ERASE parancsának. Nyitott állományra nem szabad használni.

**Exit**

Az eljárás hatására a vezérlés egy programszinttel feljebb kerül: kilép az aktuális szubrutinból illetve a főprogramból. Másképpen: GoTo utasítás az Exit-et tartalmazó blokk záró End; illetve End. sorára.

**Exp** (X:Real) : Real

*X* valós kifejezés exponenciálisát adja vissza ( $e^X$ ).

**Fail** <sup>5.5</sup>

Felszabadítja az éppen lefoglalt dinamikus objektumhoz tartozó területet a heap-ben. Csak konstruktorból hívható.

**FilePos** (Var F) : LongInt

*F* állomány azon komponensének sorszámát adja vissza, melyre az állománymutató mutat. Ha az állománymutató a legelső komponensre mutat, a függvény értéke 0 lesz. Szöveges állományokra nem alkalmazható, az

állománynak nyitva kell lennie. Amennyiben  $\text{FilePos}(F)=\text{FileSize}(F)$ , akkor  $\text{Eof}(F)=\text{True}$ .

**FileSize** (Var F) : LongInt

$F$  állomány méretét – azaz komponenseinek számát – adja vissza. Ha az állomány üres, akkor a függvény értéke 0 lesz. Szöveges állományokra nem alkalmazható, és az állománynak nyitva kell lennie.

**FillChar** (Var X; Count:Word; Ch:sorszámozott)

Feltölti  $X$  változó kezdőcímétől indulva a memória  $\text{Count}$  bájt hosszúságú területét  $\text{Ch}$  kifejezés értékével.  $\text{Ord}(\text{Ch})$  a 0..255 intervallumba eshet. Memória-ellenőrzés nincs, ezért **óvatosan használandó!**

**Flush** (Var F:Text)

A Rewrite vagy Append eljárással írásra megnyitott  $F$  szöveges állományba kírja az átmeneti puffer tartalmát. Olvasásra megnyitott állományok esetében nincs hatása.

**Frac** (X:Real) : Real

$X$  valós kifejezés tizedes részét adja vissza:  $\text{Frac}(X) = X - \text{Int}(X)$ .

**FreeMem** (Var P:Pointer ; Size:Word)

Felszabadítja a heap-ből a  $P$  mutatóhoz rendelt  $\text{Size}$  méretű területet. Amennyiben  $\text{Size}$  nem egyezik a  $P$  mutatóval mutatott terület méretével, a heap megsérülhet. A felszabadítás után  $P$  defíniálatlan értéket kap, így minden további hivatkozás  $P^*$ -ra hibát eredményezhet. Futási hibát okoz, ha hívásakor  $P$  nem a heap-be mutat.

## SYSTEM EGYSÉG

---

### **GetDir** (D:Byte; Var S:String)

A *D* paraméter által meghatározott lemezegységre vonatkozó aktuális katalógus teljes útvonalát (*d:útvonal*) adja vissza *S*-ben. *D*=0 az aktuális, 1 az A.; 2 a B.; stb. lemezegységet jelentl.

### **GetMem** (Var P:Pointer; Size:Word)

Lefoglal a heap-ben egy *Size* méretű területet, és kezdőcímét elhelyezi *P*-ben. A *P* aktuális típusától függő memóriaterületre vagy dinamikus változóra *P*<sup>^</sup>-al lehet hivatkozni. Futási hiba lép fel, ha nincs a heap-ben *Size* méretű, egybefüggő szabad terület. Maximálisan 65521 (64K - \$F) bájt foglalható le egyszerre az eljárással.

### **Halt** [ (ExitCode:Word) ]

Leállítja a program futását és a vezérlést visszaadja az operációs rendszernek. *ExitCode* opcionális paraméterrel kiszállási kódot lehet megadni, mely a Dos egység használata mellett a DosExitCode függvénnyel vagy batch állományban ERRORLEVEL-lel lekérdezhető. *ExitCode* bekerül az ExitCode változóba is. A paraméter nélküli Halt megfelel a Halt(0)-nak.

### **Hi** (X) : Byte

*X* Integer vagy Word típusú kifejezés felső (high) bájtját adja vissza.

### **Inc** (Var X [ ; n:LongInt ] )

*X* sorszámozott típusú változó értékét növeli eggyel, illetve *N*-nel. Az eljárás megfelel az *X* := Succ(*X*) egyszeri illetve *N*-szeri végrehajtásának. Az eljáráshoz a fordító optimalizált (gyors) kódot generál.

**Insert** (Source:String; Var S:String; Index:Integer)

*Source* karakterláncot beszúrja *S* karakterlánc változóba az *Index*-edik karakterpozíciótól. Ha a beszúrás utáni karakterlánc 255 karakternél hosszabb lenne, a 255. utáni karaktereket levágja. Ha *Index* nagyobb, mint *S* aktuális hossza, akkor *S* végéhez fűzi hozzá *Source* karaktereit.

**Int** (X:Real) : Real

*X* valós kifejezés egész részét adja vissza:  $\text{Int}(X) = X - \text{Frac}(X)$ .

**IOResult** : Word

Az utolsó B/K művelet hibakódját adja vissza, amennyiben a periféria-műveletek ellenőrzése inaktív (\$I-). A függvény a System egység InOutRes változójának értékét veszi fel – ami egy DOS B/K hibakód – majd nullázza azt. Ha a változó értéke nem 0, a B/K műveletek nem kerülnek végrehajtásra.

**Length** (S:String) : Integer

*S* karakterlánc aktuális hosszát adja vissza.

**Ln** (X:Real) : Real

*X* valós kifejezés természetes alapú logaritmusát adja vissza.

**Lo** (X) : Byte

*X* Integer vagy Word típusú kifejezés alsó (low) bájtját adja vissza.

**Mark** (Var P:Pointer)

Az eljárás a heap-mutató (HeapPtr) aktuális értékét átmásolja *P*-be. A heap-mutató később a Release eljárással visszaállítható.

## SYSTEM EGYSÉG

---

**MaxAvail** : LongInt

Visszaadja a heap legnagyobb egybefüggő területének méretét bájtokban.

**MemAvail** : LongInt

Visszaadja a heap összes szabad területének méretét bájtokban.

**MkDir** (S:String)

Létrehozza az *S* paraméterben megadott katalógust. *S* tartalmazhat lemez-egységnevet is. Az eljárás megfelel a DOS MD parancsának.

**Move** (Var Source, Dest; Count:Word)

*Source* változó kezdőcímétől *Count* bájtot átmásol *Dest* változóba, annak első bájtjától kezdődően. Az eljárás *Source* és *Dest* esetleges átfedését automatikusan lekezeli. Nem végez típus- illetve memória-ellenőrzést, ezért **óvatosan használandó!**

**New** (Var P:Pointer [; konstruktor<sup>5.5</sup> ] )

Létrehoz a heap-ben egy dinamikus változót – mely típusát *P* határozza meg –, és címét elhelyezi *P*-ben. *P* csak típusos mutató lehet. A dinamikus változóra *P*-al lehet hivatkozni. Futási hiba lép fel, ha a heap-ben nincs elegendő egybefüggő terület. Amennyiben *P* egy objektum mutatója, második paraméterként az objektum konstruktora is átadható, melyet az eljárás végrehajt a lefoglalás után<sup>5.5</sup>.

**Odd** (X:LongInt) : Boolean

A függvény értéke True, ha *X* egész típusú kifejezés értéke páratlan, egyébként False.

**Ofs** (X) : Word

A függvény *X* kezdőcímének eltolási (offset) értékét adja vissza. *X* egy változó vagy szubrutin azonosítója lehet.

**Ord** (X) : LongInt

*X* sorszámozott típusú kifejezés sorszámát adja vissza.

**ParamCount** : Word

Visszaadja, hogy a program hány parancssor-paraméterrel indult.

**ParamStr** (Index:Word) : String

Visszaadja az *Index*-edlik parancssor-paramétert, melyet a program indításakor adtak meg. Ha *Index* nulla vagy nagyobb, mint a **ParamCount** függvény értéke, akkor üres karakterláncot ad vissza, de 3.0 és későbbi DOS verziók alatt **ParamStr**(0) értéke a futó programot tartalmazó katalógus teljes útvonalleírása és a program neve.

**Pi** : Real

Pi értékét adja vissza: 3.1415926535897932385. Az érték pontossága hardver (\$N+) vagy szoftver (\$N-) számítási módtól függ.

**Pos** (SubStr,S:String) : Byte

Visszaadja, hogy *SubStr* karakterlánc *S* hányadik karakterpozíciójától kezdődően található meg először. Ha *S* nem tartalmazza *SubStr*-t, akkor a függvény értéke 0 lesz.

**Pred** (X) : X típusa

*X* sorszámozott kifejezés sorszám szerinti előző értékét adja vissza.

## SYSTEM EGYSÉG

---

**Ptr** (Seg,Ofs:Word) : Pointer

Mutató típusúvá konvertálja a *Seg* és *Ofs* paraméterekben megadott szegmens:eltolás címet.

**Random** [ (Range:Word) ] : Real vagy Word

Ha *Range* hiányzik, Real típusú,  $0 \leq X < 1$ , egyébként Word típusú,  $0 \leq X < Range$  véletlenszámot generál. Ha *Range* értéke 0, akkor a függvény is 0 értékű lesz. A függvény RandSeed értékét figyelembe veszi.

**Randomize**

Az operációs rendszer szerinti idő alapján véletlen értékkel inicializálja a véletlenszám generátort.

**Read** ( [ Var F:Text; ] V1 [ ,V2 ... ] ) (szöveges állományok)

Az eljárás beolvassa *F* szöveges állományból az aktuális pozíciótól kezdődő értéke(ke)t a paraméter(ek) típusától függően, majd elhelyezi *V1* [*V2 ...*] változó(k)ban. A változók egész, valós, karakteres vagy karakterlánc típusúak lehetnek. Az állománymutató az utolsó beolvasott karakter mögé áll. Numerikus változók esetében a bevezető TAB, szóköz és sorvégjel (CR/LF) karaktereket átugorja. Karakterláncok esetén sorvégjelig olvas – de ha a változó rövidebb, a szöveg csonkul. Ha *F* hiányzik, akkor az Input állományból olvas. Csak nyitott állományra alkalmazható. Billentyűzetről történő beolvasás során a szöveg szerkeszthető az alábbi billentyűkkel:



Backspace, ←, Ctrl-S	egy karakter visszatörlése
Esc, Ctrl-A	a bevétel visszatörlése, Crt egység nélkül \ jellel jelzi (mint a DOS-nál)
Ctrl-D	előző bevétel következő karaktere (Crt egységgel)
Ctrl-F	az előző bevétel lehívása
Ctrl-Z	bevétel vége, ^Z kerül a pufferbe (Crt egységgel csak ha CheckEOF = True)
Enter	bevétel vége, CR/LF kerül a pufferbe

### Read (F; V1 [, V2 ... ] ) (típusos állományok)

*F* típusos állományból beolvas egy(-egy) komponenst *V1* [*V2* ...] változó(k)ba. Az állománymutató minden olvasás után automatikusan a következő komponensre lép. A változó(k) típusának és *F* alaptípusának egyezniük kell. Csak nyitott állományra alkalmazható. Ha az állománymutató az utolsó komponens után áll, az olvasási kísérlet B/K hibát eredményez.

### ReadLn [ ( [ Var F:Text;] V1 [, V2 ...] ) ]

Csak szöveges állományokra használható. Megfelel a Read eljárásnak, de a változó(k)ba történő beolvasás után a következő sor elejére állítja az állomány mutatóját, illetve a kurzort. Ha *F* hiányzik, az Input állományból olvas – beolvasás után a puffer törlődik.

### Release (Var P:Pointer)

*P* mutató értékét átmásolja a heap-mutatóba (HeapPtr-be).

### Rename (Var F; NewName:String)

Átnevezi *NewName* névre az előzőleg *F*-hez rendelt fizikai állományt. Nyi-

## SYSTEM EGYSÉG

---

tott állományra nem szabad használni! Megfelel a DOS REN parancsának.

### **Reset** (Var F / :File ; RecSize:Word/ )

Megnyitja az előzőleg  $F$ -hez rendelt fizikai állományt, és az állománymutatót az állomány elejére állítja ( $\text{FilePos}(F)=0$ ). Ha  $F$  szöveges típusú, akkor csak olvasni lehet az állományból, míg típusos és típus nélküli állományok írhatók és olvashatók is lesznek az eljárás hívása után. Típus nélküli állományok esetében megadható a rekordméret is *RecSize* paraméterben, egyébként az 128 bájt lesz. Ha  $F$  állományváltozóhoz üres karakterláncot rendeltünk, akkor az Input állományt nyitja meg – erre nincs szükség a program indulásakor az automatikus megnyitás miatt. Nyitott állományokat automatikusan lezár újrainytás előtt.

### **Rewrite** (Var F / :File; Recsize:Word/ )

Ha az  $F$  állományváltozóhoz rendelt fizikai állomány még nem létezik, létrehozza azt, és szöveges állomány esetében írásra, típusos és típus nélküli állomány esetében írásra és olvasásra megnyitja. Ha a fizikai állomány már létezik, akkor előbb törli annak tartalmát. Az állománymutató az állomány elejére áll ( $\text{FilePos}(F)=0$ ), és az állomány üres lesz ( $\text{FileSize}(F)=0$ ). Ha  $F$  állományváltozóhoz üres karakterláncot rendeltünk, akkor az Output állományt nyitja meg írásra – erre a program indulásakor nincs szükség az automatikus megnyitás miatt. Típus nélküli állományok esetén a *RecSize* paraméterrel a rekordméret adható meg, egyébként az 128 bájt lesz.

### **Rmdir** (S:String)

Megszünteti az  $S$  paraméterben megadott alkatalógust.  $S$  tartalmazhat lemezegységnevet és útvonalleírást is. Megfelel a DOS RD parancsának. Ha

a kitörlendő katalógus nem létezik vagy nem üres, B/K hiba lép fel.

**Round** (X:Real) : LongInt

$X$  valós kifejezést a legközelebbi egészre kerekíti. Futási hiba lép fel, ha  $X$  kerekített értéke nem esik bele a LongInt típus értékészletébe. A .5-re végződő számokat a nagyobb abszolút értékű egészre kerekíti.

**RunError** [ (ErrorCode:Word) ]

Leállítja a program futását és futási hibát generál. *ErrorCode* opcionális paraméterrel a futási hiba kódja adható meg. Ha hiányzik, a kód 0 lesz.

**Seek** (Var F ; N:LongInt)

$F$  típusos vagy típus nélkül állomány mutatóját az  $N$ -edik komponensre állítja. Az első komponens sorszáma mindig 0. Ha  $N > \text{FileSize}(F)$ , akkor az állomány a legközelebbi fráskor a megadott sorszámú komponensig bővül, olvasáskor B/K hiba lép fel. Az állománynak nyitva kell lennie. Típus nélkül állományok esetén a komponens mérete 128 bájtt, vagy a nyitáskor megadott.

**SeekEof** [ (Var F:Text) ] : Boolean

A függvény az  $F$  szöveges állomány következő TAB, szóköz és sorvégjel karaktereit átugorja, és értéke True lesz, ha a mutató az állományvégjelre került, egyébként False. Ha  $F$  hiányzik, az input állományról van szó. Az állománynak nyitva kell lennie.

**SeekEoln** [ (Var F:Text) ] : Boolean

A függvény az  $F$  szöveges állomány következő TAB vagy szóköz karaktereit átugorja, és értéke True lesz, ha a mutató sorvégjelre került, egyéb-

## SYSTEM EGYSÉG

ként False. Ha *F* hiányzik, az Input állományról van szó. Az állománynak nyitva kell lennie.

**Seg (X) : Word**

Visszaadja *X* szegmenscímét. *X* egy változó vagy szubrutin azonosítója.

**SetTextBuf (Var F:Text; Var Buf [ ; Size:Word] )**

Hozzárendeli *F* szöveges állományváltozóhoz az eredeti 128 bájtos B/K puffer helyett *Buf* átmeneti puffert. *Size* opcionális paraméterrel lehet megadni a puffer méretét; ha hiányzik, akkor *Buf* mérete határozza meg a puffer hosszát. Nyitott állományra csak közvetlenül a Reset, Rewrite vagy Append eljárás után ajánlatos használni – egy esetleges adatvesztés miatt. Az eljárás hatása a következő, *F*-re vonatkozó Assign-ig, illetve a program végéig tart. CON perifériáról történő bevitel során az eljárással korlátozhatjuk a bevitelhető karakterek számát.

- ! *Buf* méretét és *Size* értékét vegyük mindig két bájttal nagyobbra, a sorvégjel (CR/LF) miatt!
- ! Tapasztalataink szerint csak a Crt egység használata mellett van hatása a *Size* paraméternek.

**Sin (X:Real) : Real**

*X* valós kifejezés (radlán) szinuszát adja vissza.

**SizeOf (X) : Word**

*X* típus vagy változó fizikai méretét adja vissza, bájtokban. VMT-vel rendelkező objektum típusú változók esetén a VMT-ben tárolt méret (azaz a VMT első két bájtlján tárolt szó) lesz a függvény értéke<sup>5.5</sup>.

**SPtr** : Word

A veremmutató aktuális eltolási címét (SP regiszter tartalmát) adja vissza.

**Sqr** (X) : X típusa

X egész vagy valós típusú kifejezés négyzetét adja vissza.

**Sqrt** (X:Real) : Real

X valós kifejezés négyzetgyökét adja vissza.

**SSeg** : Word

A veremmutató szegmenscímét (SS regiszter tartalmát) adja vissza.

**Str** (X [ : Width [ : Decimals]] ; Var S:String)

X egész vagy valós típusú kifejezés értékét az őt reprezentáló karakterláncá alakítja és azt visszaadja az S paraméterben. *Width* a mezőszélességet, valós típusú értékek esetében *Decimals* a tizedesek számát határozza meg. A karaktereket a megadott mezőben jobbra rendezve helyezi el, bevezető szóközök után. Ha *Width* hiányzik vagy értéke túl kicsi, akkor X-et annyi karakteren ábrázolja, amennyi szükséges.

**Succ** (X) : X típusa

X sorszámozott típusú kifejezés sorszám szerinti következő értékét adja vissza.

**Swap** (X) : X típusa

X Integer vagy Word típusú kifejezés értékét az alsó és felső bájt felcserélésével adja vissza.

## SYSTEM EGYSÉG

---

**Trunc** (X:Real) : LongInt

*X* valós kifejezés nulla felé kerekített egész részét adja vissza. Futási hibát okoz, ha az érték nem esik bele a LongInt típus értékkészletébe.

**Truncate** (Var F)

Levágja az *F* bármilyen típusú állomány komponenselt az aktuális pozíciótól (Eof(*F*) True lesz). Az állománynak nyitva kell lennie.

**TypeOf** (objektumváltozó) : Pointer<sup>5.5</sup>

Visszaadja az *objektumváltozó*-hoz tartozó virtuális metódusok táblázatának (VMT) címét.

**UpCase** (Ch:Char) : Char

A *Ch* karakternek megfelelő nagybetűt adja vissza. Ha *Ch* értéke nem esik az 'a'..'z' intervallumba, akkor a függvény értéke megegyezik *Ch*-val.

**Val** (S:String; Var V; Var Code:Integer)

*S* karakterlánc típusú kifejezést *V* típusának megfelelően numerikussá alakítva elhelyezi *V* egész vagy valós típusú változóban. *Code* az átalakítás során talált első hibás karakter pozícióját tartalmazza visszatéréskor, egyébként értéke 0. Hiba esetén *V* nem változik. Az eljárás a bevezető szóközöket és TAB karaktereket átugorja. Ha az átalakított érték nem esik bele a *V* típusa szerinti értékkészletbe, akkor \$R+ esetén ez futási hibát okoz, míg \$R- mellett a valós vagy LongInt típusú *V* értéke definiálatlan és *Code* nullától eltérő lesz; egyéb típusoknál *V* túlcscordul és *Code* értéke 0 lesz.

**Write** ( / Var F:Text ; / V1 [ ,V2 ...] ) (szöveges állomány)

Az eljárás kírja *F* szöveges állományba az aktuális pozíciótól *V1* [*V2* ...] kifejezés(ek) értékét. Az állománymutató az utolsó kírít karakter mögé áll. Ha *F* hiányzik, akkor az Output állományba ír. Az eljárással egész, valós, karakteres, karakterlánc és logikai típusú értékek írhatók ki. Mezőszéleség és valós típusok esetében a tizedesek száma is megadható az Str eljáráshoz hasonlóan.

**Write** (F, V1 [ ,V2 ...] ) (típusos állomány)

*F* típusos állományba kírja *V1* [*V2* ...] változó(k) értékét. Az állománymutató minden írás után automatikusan a következő komponensre lép. A változó(k) típusának és *F* alaptípusának egyezniük kell. Ha az állománymutató az utolsó komponens után áll, bővül az állomány.

**WriteLn** ( / Var F:Text; / V1 [ ,V2 ...] )

Csak szöveges állományokra használható. Megfelel a Write eljárásnak, de a kifejezés(ek) után még sorvégjelet (CR/LF) is kírít. Kifejezés(ek) hiányában csak sorvégjelet ír ki.

### Megjegyzés:

A System egységet nem kell sem programokban, sem saját egységekben a USES kulcsszó után megadni, mert a fordító automatikusan minden programhoz és egységhez hozzászerkeszti.

## Konstansok

Háttér- és tintaszínek:

**Black** = 0 ;  
**Blue** = 1 ;  
**Green** = 2 ;  
**Cyan** = 3 ;  
**Red** = 4 ;  
**Magenta** = 5 ;  
**Brown** = 6 ;  
**LightGray** = 7 ;

További tintaszínek:

**DarkGray** = 8 ;  
**LightBlue** = 9 ;  
**LightGreen** = 10 ;  
**LightCyan** = 11 ;

**LightRed** = 12 ;  
**LightMagenta** = 13 ;  
**Yellow** = 14 ;  
**White** = 15 ;  
**Blink** = 128 ; (tintaszínhez adandó)

Szöveges üzemmódok:

**BW40** = 0 ;  
**BW80** = 2 ;  
**CO40** = 1 ;  
**CO80** = 3 ;  
**C40** = 1 ;  
**C80** = 3 ;  
**Mono** = 7 ;  
**Font8x8** = 256 ;

## Változók

**CheckBreak** : Boolean ;  
**CheckEOF** : Boolean ;  
**CheckSnow** : Boolean ;  
**DirectVideo** : Boolean ;

**LastMode** : Word ;  
**TextAttr** : Byte ;  
**WindMin** : Word ;  
**WindMax** : Word ;

### • CheckBreak

Ha értéke True, a program képernyőre történő íráskor megszakítható a Ctrl-Break billentyű-kombinációval, egyébként nem (keretrendszerben futtatott programok mindig megszakíthatók)



## CRT EGYSÉG

---

- **CheckEOF**  
Ha értéke True, a ^Z leütése hatásos, egyébként nem
- **CheckSnow**  
Ha értéke True, CGA monitorvezérlő esetén elkerülhető kírás alatt a képernyő „kásásodása”; csak DirectVideo=True mellett hatásos
- **DirectVideo**  
Ha értéke True, a kírások közvetlenül a video memóriába történnek, egyébként a program a BIOS-on keresztül ír (ez utóbbi áll fenn, ha az egységet nem használja a program)
- **LastMode**  
A TextMode eljárás hívásakor érvényes képernyő üzemmód eltárolására szolgál
- **TextAttr**  
A legközelebbi képernyőre történő írás attribútumát tartalmazza – 0..3 bit: tintaszín; 4..6 bit: háttérszín; 7. bit: villogás
- **WindMin, WindMax**  
Az aktuális ablak koordinátái (felső bájtt: Y, alsó bájtt: X koordináta)

Az egység inicializáló részében CheckBreak, CheckSnow és DirectVideo True, míg CheckEOF False értéket kap; LastMode, TextAttr, WindMin és WindMax a program indításakor érvényes értékeket vesz fel.

## Eljárások, függvények

### AssignCrt (Var F:Text)

*F* Text típusú állományváltozót hozzárendell a Turbo Pascal-ban definiált CRT perifériához. Használatával gyorsabb képernyőkezelést lehet elérni. A szabványos Input és Output állományokat az egység inicializáló része automatikusan hozzárendell a CRT-hez és megnyitja.

### CirEol

Az aktuális háttérszínnel törli a kurzor és a sor vége közötti karaktereket. Az eljárás ablakrelatív; a kurzor pozíciója változatlan marad.

### CirScr

Az aktuális háttérszínnel törli az aktuális szöveges ablakot, és a kurzort az ablak bal felső sarkába, az (1,1) pozícióba viszi.

### Delay (Ms:Word)

*Ms* milliszekundum késleltetést vált ki az eljárás.

### DelLine

A kurzor sorát törli, és az aktuális szöveges ablak további sorait feljebb pergeti. Az ablak utolsó sorát az aktuális háttérszínnel törli. A kurzor helye nem változik.

### GoToXY (X,Y:Byte)

A kurzort az aktuális szöveges ablak (*X*,*Y*) karakter-pozíciójába viszi. *X* a vízszintes (oszlop), *Y* a függőleges (sor) koordináta. Az eljárás ablakrelatív.

## CRT EGYSÉG

---

### **HighVideo**

Nagy intenzitásúra (fényesre) változtatja az aktuális tintaszínt szöveges üzemmódban. Hatása a következő klírás alkalmával válik láthatóvá. Az eljárás TextAttr 3. bitjét állítja 1-re.

### **InsLine**

A kurzor sorát, valamint az aktuális ablak további sorait lejjebb pergeti. A kurzor sorába beszúr egy üres sort az aktuális háttérszínnel. Az ablak utolsó sora elvész; a kurzor helye nem változik.

### **KeyPressed** : Boolean

A függvény értéke True, ha a billentyűzet pufferben van kiolvásra várakozó karakter, egyébként False. A puffer tartalma nem változik. A függvény nem érzékeli az ALT, SHIFT, CTRL, NUM LOCK és CAPS LOCK módosító billentyűket önmagukban, mivel ezek kódjai nem kerülnek be a pufferbe. Bizonyos billentyűk és kombinációk kettős kódot generálnak (lásd BILLENTYŰZET KÓDTÁBLA).

### **LowVideo**

Kis intenzitásúra változtatja az aktuális tintaszínt szöveges üzemmódban. Hatása a következő klírás alkalmával válik láthatóvá. Az eljárás TextAttr 3. bitjét állítja 0-ra.

### **NormVideo**

Visszaállítja a program indításakor a kurzor pozíciójában detektált szöveges háttér- és tintaszínt.

**NoSound**

Kikapcsolja a beépített hangszórót.

**ReadKey** : Char

Beolvas a billentyűzetről egy karaktert, mely a képernyőn nem jelenik meg. A billentyűzet puffer egy karakterrel rövidebb lesz. Ha a függvény hívásakor a billentyűzet pufferben van kielvételre várakozó karakter (KeyPressed=True), akkor a függvény rögtön visszatér, egyébként billentyűleütésre vár. Bizonyos billentyűk illetve kombinációk esetében két karakter kerül a pufferbe: #0 majd a billentyű saját (scan-) kódja (lásd BILLYENTŰZET KÓDTÁBLA). Ha CheckBreak értéke False, akkor a Ctrl-Break leütése #3-as karaktert generál, egyébként a program futása megszakad.

**Sound** (Hz:Word)

Bekapcsolja a belső hangszórót, mely *Hz* frekvenciájú hangot bocsát ki. A hangszóró csak a NoSound eljárással kapcsolható ki. A normál A (a1) hang frekvenciája 440 Hz. Minden féhang az előző  $12\sqrt{2}$ -szöröse.

**TextBackground** (Color : Byte)

Szöveges üzemmódban a háttérszínt (TextAttr 4.6 bitjel) *Color*-ra (0..7) állítja. 7-nél nagyobb érték esetén a szín *Color* MOD 8 lesz. (Színkonstansokat lásd Konstansok – Háttér- és tintaszínek.)

**TextColor** (Color : Byte)

Szöveges üzemmódban a tintaszínt (TextAttr 0.3 bitjel) *Color*-ra (0..15) állítja. Ha *Color*-hoz 128-at adunk, a kilírt szöveg villogni fog (TextAttr 7. bitje). (Színkonstansokat lásd Konstansok – Háttér- és tintaszínek, Továb-

## CRT EGYSÉG

---

bi tintaszínek.)

### **TextMode** (Mode : Word)

*Mode* szöveges üzemmódba állítja a képernyőt. A szöveges ablak az egész képernyő lesz; *TextAttr* 3. bitjét 0-ra, színes mód esetén *CheckSnow* és *DirectVideo* változót True értékűre állítja. Átállítás előtt az aktuális üzemmódot elmenti *LastMode* változóba.

### **WhereX** : Byte

Szöveges üzemmódban a kurzor X (vízszintes) karakter-pozícióját adja vissza. A függvény értéke ablakrelatív.

### **WhereY** : Byte

Szöveges üzemmódban a kurzor Y (függőleges) karakter-pozícióját adja vissza. A függvény értéke ablakrelatív.

### **Window** (X1,Y1,X2,Y2:Byte)

Szöveges üzemmódban az ablakot állítja be (*X1,Y1*) bal felső, (*X2,Y2*) jobb alsó sarokpozíciókkal. Az aktuális ablak alapértelmezésben az egész képernyő (pl.: (1,1,80,25)).

## Változó

Lst : Text ;

Az egység inicializáló része automatikusan hozzárendeli az első nyomtatót (LPT1-et, vagy másképpen PRN-t) Lst-hez, és megnyitja írásra azt.

## Típusok

Keresőrekord (FindFirst és FindNext):

**SearchRec =**

```
Record
  Fill : Array[1..21] Of Byte ;
  Attr : Byte ;
  Time : LongInt ;
  Size : LongInt ;
  Name : String[12] ;
End ;
```

Típusos és típusnélküli állomány adatai:

**FileRec =**

```
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 ;
```

Szöveges B/K puffer:

**TextBuf = Array[0..127] Of Char ;**

Szöveges állomány adatai:

**TextRec =**

```
Record
  Handle : Word ;
  Mode : Word ;
  BufSize : Word ;
  Private : Word ;
  BufPos : Word ;
  BufEnd : Word ;
  BufPtr : ^TextBuf ;
  OpenFunc : Pointer ;
  InOutFunc : Pointer ;
  FlushFunc : Pointer ;
  CloseFunc : Pointer ;
  UserData : Array[1..16] Of Byte ;
  Name : Array[0..79] Of Char ;
  Buffer : TextBuf ;
End ;
```

## DOS EGYSÉG

Dátumrekord:

**DateTime =**

Record

**Year** : Word ;

**Month** : Word ;

**Day** : Word ;

**Hour** : Word ;

**Min** : Word ;

**Sec** : Word ;

End ;

Parancssor karakterlánc:

**ComStr =** String[127] ;

Útvonal karakterlánc:

**PathStr =** String[79] ;

Katalógus karakterlánc:

**DirStr =** String[67] ;

Állománynév karakterlánc:

**NameStr =** String[8] ;

Kiterjesztés karakterlánc:

**ExtStr =** String[4] ;

CPU regiszterek (MsDos és Intr):

**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 ;



## Konstansok

Flags regiszter bitjel:

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

Állományok állapotai:

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

Állomány-attribútumok:

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

## Változó

**DOSError** : Integer ;

értékelnek jelentése:

2 Az állomány nem található  
 3 Az útvonal nem található  
 5 Tiltott hozzáférés  
 6 Érvénytelen kezelőszám (handle)

8 Kevés a memória  
 10 Érvénytelen környezet  
 11 Érvénytelen formátum  
 18 Nincs több állomány

## Eljárások, függvények

### **DiskFree** (Drive:Word) : LongInt

Visszaadja a *Drive* lemezegységben lévő lemez szabad területének méretét. Ha *Drive* értéke 0, akkor az aktuális, ha 1 az A.; ha 2 a B.; stb. lemezegységről van szó. Ha a lemezegység száma (*Drive* paraméter) hibás, vagy a lekérdezés sikertelen, akkor a függvény értéke -1 lesz.

### **DiskSize** (Drive:Word) : LongInt

Visszaadja a *Drive* lemezegységben lévő lemez teljes kapacitását. Ha *Drive* értéke 0, akkor az aktuális, ha 1 az A.; ha 2 a B.; stb. lemezegységről van szó. Ha a lemezegység száma (*Drive* paraméter) hibás, vagy a lekérdezés sikertelen, akkor a függvény értéke -1 lesz.

### **DosExitCode** : Word

Az Exec eljárással hívott program visszatérési kódját adja vissza. Az alsó bájt tartalmazza a program által küldött kódot – a Halt vagy Keep eljárásnak adott paramétert –, míg a felső bájt értéke:

- 0 ha normálisan lefutott az alprogram,
- 1 ha Ctrl-C -vel megszakították,
- 2 ha futási hibával állt le, és
- 3 ha Keep eljárással fejeződött be a futása.

### **DosVersion** : Word

Visszaadja a DOS verziószámát. Az alsó bájt tartalmazza a fő-, a felső bájt az al-verziószámot.

**EnvCount** : Integer

A DOS környezeti változók számát adja vissza.

**EnvStr** (Index:Integer) : String

Az *Index*-edlik DOS környezeti változó értékét adja vissza. Az első DOS környezeti változó száma 1. Ha *Index* értéke kisebb, mint 1, vagy nagyobb, mint EnvCount értéke, akkor a függvény értéke üres karakterlánc lesz.

**Exec** (Path,CmdLine:String)

A *Path* paraméterben megadott programot *CmdLine* parancssorral futtatja. *Path* tartalmazhat lemezegységnevet és útvonalleírást is. A heap maximális felső határát akkorára kell lecsökkenteni, hogy a futtatandó külső program beférjen a memóriába (lásd \$M direktíva). Az eljárás eredményességét DOSError változó tartalmazza a hívás után – ha értéke 0, a hívott program visszatérési kódja a DosExitCode függvényvel lekérdezhető.

! A COMMAND.COM futtatásakor az első parancssor-paraméter '/C' kell legyen!

! Hívása előtt és után használjuk a SwapVectors eljárást!

**FExpand** (Path:PathStr) : PathStr

A paraméterként átadott *Path* fizikai állománynevet teljes – lemezegységnevet, útvonalleírást és állománynevet tartalmazó – állomány-specifikációvá alakítva adja vissza. A PathStr típust az egység definiálja.

**FindFirst** (Path:String; Attr:Word; Var S:SearchRec)

A *Path* paraméterben meghatározott normál és *Attr* attribútumú (lásd előre deklarált állomány-attribútumok) katalógus-bejegyzések közül kikere-

## DOS EGYSÉG

---

si az elsőt, és elhelyezi az *S* változóban. *Path* tartalmazhat lemezegységnevet és útvonalleírást is. Az eljárás eredményességét *DOSError* változó tartalmazza a hívás után. A *SearchRec* típust az egység definiálja.

### **FindNext** (Var *S*:*SearchRec*)

Az előzőleg *FindFirst* eljárással definiált katalógus-bejegyzések közül a következőt adja vissza az *S* paraméterben. Az eljárás eredményességét *DOSError* változó tartalmazza a hívás után. A *SearchRec* típust az egység definiálja.

### **FSearch** (Path:*PathStr*; *DirList*:*String*) : *PathStr*

A *Path* paraméterben megadott fizikai állományt megkeresi az aktuális lemezegység aktuális katalógusában, majd a *DirList* paraméterrel meghatározott katalógusokban. Ha megtalálta, akkor a teljes állomány-specifikációt adja vissza, egyébként a függvény értéke üres karakterlánc lesz. A *DirList* paraméterben az egyes útvonalakat egymástól pontosvesszővel elválasztva kell megadni, mint a DOS *PATH* környezeti változójánál. A *PathStr* típust az egység definiálja.

### **FSplit** (Path:*PathStr*; Var *Dir*:*DirStr*; Var *Name*:*NameStr*; Var *Ext*:*ExtStr*)

A *Path* paraméterben megadott állomány-specifikációt komponenseire bontva adja vissza. *Dir* tartalmazza a lemezegységkódot és az útvonalat, *Name* az állomány nevét, *Ext* pedig a kiterjesztést az elválasztó ponttal együtt. Ha *Path*-ban nem adunk meg egy komponenst, akkor az annak megfelelő változó üres karakterlánc lesz. A *PathStr*, *DirStr*, *NameStr* és *ExtStr* típusokat az egység definiálja.

**GetCBreak** (Var Break:Boolean)

*Break* paraméterében visszaadja, hogy a DOS Ctrl-Break ellenőrzése aktív-e, vagy sem. Ha értéke False – vagyis az ellenőrzés inaktív –, a DOS csak periféria-műveletek hívásakor fogadja el a program Ctrl-Break-kel történő megszakítását, míg True esetén minden rendszer-híváskor.

**GetDate** (Var Year,Month,Day,DayOfWeek:Word)

Az operációs rendszer szerinti aktuális dátumot adja vissza paraméterekben: *Year* az évszámot (1980..2099), *Month* a hónap számát (1..12), *Day* a nap számát (1..31) és *DayOfWeek* a hét napjának sorszámát (0 a vasárnap, 1 a hétfő, stb.) tartalmazza visszatéréskor.

**GetEnv** (EnvVar:String) : String

Az *EnvVar* paraméterben meghatározott DOS környezeti változó értékét adja vissza. Ha az nem létezik, a függvény értéke üres karakterlánc lesz.

**GetFAttr** (Var F; Var Attr:Word)

Az *F*-hez rendelt fizikal állomány attribútumát adja vissza az *Attr* paraméterben (lásd Konstansok – Állomány-attribútumok). Az állománynak nem szabad nyitva lennie az eljárás hívásakor.

**GetFTime** (Var F; Var Time:LongInt)

Az *F*-hez rendelt fizikal állomány utolsó írási dátumát és idejét adja vissza pakolt formában a *Time* paraméterben. Ez az UnpackTime eljárással szétpakolható. Az állománynak nyitva kell lennie.

**GetIntVec** (IntNo:Byte; Var Vector:Pointer)

Az *IntNo* paraméterrel meghatározott megszakítási rutin címét adja visz-

## DOS EGYSÉG

---

sza a *Vector* paraméterben.

### **GetTime** (Var Hour,Minute,Second,Sec100:Word)

Paraméterekben visszaadja az operációs rendszer szerinti aktuális időt. *Hour* tartalmazza az órákat (0..23), *Minute* a perceket (0..59), *Second* a másodperceket (0..59), *Sec100* a századmásodperceket (0..99).

### **GetVerify** (Var Verify:Boolean)

Visszaadja a DOS VERIFY jelzőjének aktuális értékét. Ha *Verify* True értékű, akkor minden lemez-írásakor külön ellenőrzés is történik, egyébként nem.

### **Intr** (IntNo:Byte; Var Regs:Registers)

Meghívja az *IntNo* paraméterrel meghatározott megszakítást. A megszakítás hívása előtt a processzor AX, BX, CX, DX, BP, SI, DI, DS és ES regisztereit feltölti a *Regs* rekord megfelelő mezőivel, majd visszatéréskor ezeket, valamint a Flags regisztert visszatölti *Regs*-be. A Registers típust az egység definiálja.

! Azon megszakításokat, melyek függenek SP és SS értékétől, illetve ezeket megváltoztatják, **nem szabad meghívni!**

### **Keep** (ExitCode:Word)

Leállítja a program futását, a vezérlést visszaadja a hívó programnak, illetve az operációs rendszernek, de a program a kód-, adat- és verem szegmensével, valamint a heap-pel együtt a memóriában marad (rezidens kiszállás). A kiszállási kódot *ExitCode*-dal kell megadni, melyet a hívó programból a *DosExitCode* függvénnyel vagy batch állományban ERROR-LEVEL-lel lekérdezhetünk.

I Ha a heap méretét nem korlátozzuk rezidens programokban, a rendszer – szabad memória hlányában – lefagy.

### **MsDos** (Var Regs:Registers)

A \$21-es megszakítást hívja meg (DOS belépési pont). A megszakítás hívása előtt a processzor AX, BX, CX, DX, BP, SI, DI, DS és ES regiszterrel feltölti a *Regs* rekord megfelelő mezőivel, majd visszatéréskor ezeket, valamint a *Flags* regisztert visszatölti *Regs*-be. A *Registers* típus az egység definiálja.

I Azon funkciókat, melyek függenek SP és SS értékétől, illetve ezeket megváltoztatják, **nem szabad meghívni!**

### **PackTime** (Var DT:DateTime; Var Time:LongInt)

*DT* rekord tartalmát pakoltan elhelyezi *Time* paraméterben. A *DateTime* típus az egység definiálja.

### **SetCBreak** (Break:Boolean)

Az eljárással a DOS Ctrl-Break ellenőrzése be- vagy kikapcsolható. Ha *Break* értéke *False*, a programok csak periféria-műveletek hívásakor szakíthatók meg Ctrl-Break-kel, míg *True* esetén minden rendszer-híváskor.

### **SetDate** (Year,Month,Day:Word)

Beállítja az operációs rendszer dátumát. *Year* tartalmazza az évszámot (1980..2099), *Month* a hónap (1..12) és *Day* a nap számát (1..31). Ha a megadott dátum érvénytelen, akkor az eljárás hatástalan.

### **SetFAttr** (Var F; Attr:Word)

Az *F*-hez rendelt fizikai állomány attribútumát megváltoztatja az *Attr* pa-

## DOS EGYSÉG

---

raméterben megadottra (lásd Konstansok – Állomány-attribútumok). Az állománynak zárva kell lennie.

### **SetFTime** (Var F; Time:LongInt)

Beállítja az *F*-hez rendelt fizikai állomány utolsó írásának dátumát és idejét, melyet a *Time* paraméterben pakolt formában adunk meg. Az állománynak nyitva kell lennie. *Time* értéke a PackTime eljárással állítható elő.

### **SetIntVec** (IntNo:Byte; Vector:Pointer)

Beállítja az *IntNo* sorszámú megszakítási vektort a *Vector* paraméterrel meghatározott címre.

### **SetTime** (Hour,Minute,Second,Sec100:Word)

Beállítja az operációs rendszer szerinti időt. *Hour* az órák (0..23), *Minute* a percek (0..59), *Second* a másodpercek (0..59) és *Sec100* a századmásodpercek (0..99) értékét tartalmazza híváskor. Ha a megadott idő érvénytelen, az eljárás hatástalan.

### **SetVerify** (Verify:Boolean)

Beállítja a DOS VERIFY jelzőjének aktuális értékét. Ha *Verify* True értékű, akkor minden lemez-írásakor külön ellenőrzés is történik, egyébként nem.

### **SwapVectors**

A System egységben deklarált SaveIntXX – megszakítási vektorokat tartalmazó – változók értékét felcseréli a hozzájuk tartozó megszakítási vektorokkal. Tipikusan az Exec eljárás hívása előtt és után használatos. A fordító a főprogramhoz olyan kódot generál, mely a deklarált megszakítási vektorokat induláskor elmenti, kilépéskor pedig visszaállítja.



**UnpackTime** (Time:LongInt; Var DT:DateTime)

A *DT* rekordváltozóba szétpakolja a *Time* paraméterrel meghatározott pakolt dátumot és időt. A *DateTime* típus az egység definiálja.

## Típusok

Vonaljellemzők:

*stílus,*  
*felhasználói minta,*  
*vastagság*

**LineStyleType** =

Record  
**LineStyle** : Word ;  
**Pattern** : Word ;  
**Thickness** : Word ;  
End ;

Szövegjellemzők:

*betűtípus,*  
*szövegállás,*  
*karakterméret,*  
*vízszintes igazítás,*  
*függőleges igazítás*

**TextSettingsType** =

Record  
**Font** : Word ;  
**Direction** : Word ;  
**CharSize** : Word ;  
**Horiz** : Word ;  
**Vert** : Word ;  
End ;

Kitöltési jellemzők:

*stílus,*  
*szín*

**FillSettingsType** =

Record  
**Pattern** : Word ;  
**Color** : Word ;  
End ;

Pontkoordináta:

**PointType** =

Record  
**X** : Integer ;  
**Y** : Integer ;  
End ;

Palettajellemzők:

*méret,*  
*színek*

**PaletteType** =

Record  
**Size** : Byte ;  
**Colors** : Array[0..MaxColors] Of ShortInt ;  
End ;

## GRAPH EGYSÉG

Grafikus ablakjellemzők:

*bal felső sarok,  
jobb alsó sarok,  
grafika vágás*

**ViewPortType** =

Record

**X1** : Integer ;

**Y1** : Integer ;

**X2** : Integer ;

**Y2** : Integer ;

**Clip** : Boolean ;

End ;

Ívjellemzők:

*középpont,*

*kezdőpont,*

*végpont*

**ArcCoordsType** =

Record

**X** : Integer ;

**Y** : Integer ;

**XStart** : Integer ;

**YStart** : Integer ;

**XEnd** : Integer ;

**YEnd** : Integer ;

End ;

Kitöltési minta:

**FillPatternType** = Array[1..8] Of Byte ;

## Konstansok

Grafikus hibakódok:

```
grOk = 0 ;
grNoInItGraph = -1 ;
grNotDetected = -2 ;
grFileNotFound = -3 ;
grInvalidDriver = -4 ;
grNoLoadMem = -5 ;
grNoScanMem = -6 ;
grNoFloodMem = -7 ;
grFontNotFound = -8 ;
grNoFontMem = -9 ;
grInvalidMode = -10 ;
grError = -11 ;
grIOError = -12 ;
grInvalidFont = -13 ;
grInvalidFontNum = -14 ;
```

Grafikus meghajtók:

```
Detect = 0 ;
CGA = 1 ;
MCGA = 2 ;
EGA = 3 ;
EGA64 = 4 ;
EGAMono = 5 ;
```

```
IBM8514 = 6 ;
HercMono = 7 ;
ATT400 = 8 ;
VGA = 9 ;
PC3270 = 10 ;
CurrentDriver = -128 ;
```

Grafikus üzemmódok:

```
CGAC0 = 0 ;
CGAC1 = 1 ;
CGAC2 = 2 ;
CGAC3 = 3 ;
CGAHI = 4 ;
MCGAC0 = 0 ;
MCGAC1 = 1 ;
MCGAC2 = 2 ;
MCGAC3 = 3 ;
MCGAMed = 4 ;
MCGAHI = 5 ;
EGALo = 0 ;
EGAHI = 1 ;
EGA64Lo = 0 ;
EGA64HI = 1 ;
EGAMonoHI = 3 ;
```

## GRAPH EGYSÉG

HercMonoHI = 0 ;  
ATT400C0 = 0 ;  
ATT400C1 = 1 ;  
ATT400C2 = 2 ;  
ATT400C3 = 3 ;  
ATT400Med = 4 ;  
ATT400HI = 5 ;  
VGALo = 0 ;  
VGAMed = 1 ;  
VGAHI = 2 ;  
PC3270HI = 0 ;  
IBM8514Lo = 0 ;  
IBM8514Hi = 1 ;

Normál színek:





Black = 0 ;  
Blue = 1 ;  
Green = 2 ;  
Cyan = 3 ;  
Red = 4 ;  
Magenta = 5 ;  
Brown = 6 ;  
LightGray = 7 ;  
DarkGray = 8 ;  
LightBlue = 9 ;  
LightGreen = 10 ;

LightCyan = 11 ;  
LightRed = 12 ;  
LightMagenta = 13 ;  
Yellow = 14 ;  
White = 15 ;



EGA színek:

EGABlack = 0 ;  
EGABlue = 1 ;  
EGAGreen = 2 ;  
EGACyan = 3 ;  
EGARed = 4 ;  
EGAMagenta = 5 ;  
EGABrown = 20 ;  
EGALightGray = 7 ;  
EGADarkGray = 56 ;  
EGALightBlue = 57 ;  
EGALightGreen = 58 ;  
EGALightCyan = 59 ;  
EGALightRed = 60 ;  
EGALightMagenta = 61 ;  
EGAYellow = 62 ;  
EGAWhite = 63 ;

## Vonalstílus:

**SolidLn** = 0 ;   
**DottedLn** = 1 ;   
**CenterLn** = 2 ;   
**DashedLn** = 3 ;   
**UserBitLn** = 4 ; (felhasználói)

## Vonalvastagság:

**NormWidth** = 1 ;   
**ThickWidth** = 3 ; 

## Betűtípus:

**DefaultFont** = 0 ; (bittérképes)  
**TriplexFont** = 1 ;  
**SmallFont** = 2 ;  
**SansSerifFont** = 3 ;  
**GothicFont** = 4 ;

## Szövegállás:

**HorizDir** = 0 ;  
**VertDir** = 1 ;

## Felhasználói karakterméret:

**UserCharSize** = 0 ;

## Szöveglgázítás:

**LeftText** = 0 ;  
**CenterText** = 1 ;  
**RightText** = 2 ;  
**BottomText** = 0 ;  
**TopText** = 2 ;

## Grafika vágás:

**ClipOn** = True ;  
**ClipOff** = False ;

## Tetőrajzolás:



**TopOn** = True ;  
**TopOff** = False ;

## Kitöltési stílus:

**EmptyFill** = 0 ;  
**SolidFill** = 1 ;  
**LineFill** = 2 ;  
**LtSlashFill** = 3 ;  
**SlashFill** = 4 ;  
**BkSlashFill** = 5 ;  
**LtBkSlashFill** = 6 ;  
**HatchFill** = 7 ;  
**XHatchFill** = 8 ;  
**InterleaveFill** = 9 ;



## GRAPH EGYSÉG

WideDotFill = 10 ;   
CloseDotFill = 11 ;   
UserFill = 12 ; (felhasználói)

Legnagyobb színkód érték:  
MaxColors = 15 ;

Rajzolási mód:  
CopyPut = 0 ;  
XORPut = 1 ;  
ORPut = 2 ;  
ANDPut = 3 ;  
NOTPut = 4 ;

### Változók

GraphGetMemPtr : Pointer ;

GraphFreeMemPtr : Pointer ;

### Eljárások, függvények

- A külön nem jelölt szubrutinok (funkciójuktól függően) mindig az aktuális beállításokkal dolgoznak (alapértelmezéseket lásd GraphDefaults):
  - háttérszín (lásd SetBkColor)
  - tintaszín (lásd SetColor)
  - vonaljellemzők (lásd SetLineStyle)
  - kitöltési jellemzők (lásd SetFillStyle, SetFillPattern)
  - rajzolási mód (lásd SetWriteMode) – csak egyenes vonalak esetében
  - szövegjellemzők (lásd SetTextStyle, SetTextJustify)
- A szögek fokokban értendők, az óra járásával ellentétesen növekszenek. (0°: → , 90°: ↑ , 180°: ← , 270°: ↓ , ... periodikus.)
- A koordináták mindig ablakrelatívák, kivéve a SetViewPort-ban.
- A G-vel jelölt szubrutinok hívásakor grafikus üzemmódban kell lennie a gépnek.
- A hibakódok a GraphResult függvénnyel lekérdezhetők.

- Arc** (X,Y:Integer; StAngle, EndAngle, Radius:Word) **G**  
 Körívet rajzol (X,Y) középpont köré *Radius* sugárral. A körív *Stangle* szög-  
 től *EndAngle* szögig rajzolódik. A legutóbb húzott ív adatai a *GetArcCoords*  
 eljárással lekérdezhetők.
- Bar** (X1,Y1,X2,Y2:Integer) **G**  
 Keretvonal nélkül kitöltött négyszöget rajzol. (X1,Y1) és (X2,Y2) a négyszö-  
 get definiáló két átellenes sarok koordinátái.
- Bar3D** (X1,Y1,X2,Y2:Integer; Depth:Word; Top:Boolean) **G**  
 Háromdimenziós, keretvonalal határolt téglatestet rajzol. (X1,Y1) és (X2,Y2)  
 az elülső lap két átellenes sarkának koordinátái; *Depth* a mélység; ha *Top*  
 értéke True, akkor a téglatest tetejét is megrajzolja, egyébként nem (lásd  
 Konstansok – Tetőrajzolás). Az eljárás a téglatest elülső lapját kitölti.
- Circle** (X,Y:Integer; Radius:Word) **G**  
 (X,Y) középpont köré *Radius* sugárral kört rajzol.
- ClearDevice** **G**  
 Törli a teljes képernyőt és a grafikus kurzort az aktuális ablak (0,0) pontjá-  
 ba viszi.
- ClearViewPort** **G**  
 Törli az aktuális ablakot, a grafikus kurzort a (0,0) pontba viszi.
- CloseGraph** **G**  
 Visszaállítja a grafikus rendszer Indítása előtti szöveges üzemmódot és fel-  
 szabadítja a heap-ben a grafikus rendszer számára lefoglalt területet.



## GRAPH EGYSÉG

---

### **DetectGraph** (Var GraphDriver,GraphMode:Integer)

Visszaadja *GraphDriver* és *GraphMode* paramétereiben a detektált grafikus meghajtót és annak legnagyobb felbontású üzemmódját. Ha a gépben nincs grafikus meghajtó, akkor mind *GraphDriver*, mind a *GraphResult* függvény  $-2$  értékű lesz (*grNotDetected*).

### **DrawPoly** (NumPoints:Word; Var PolyPoints)

G

A *NumPoints* paraméterben meghatározott számú, *PolyPoints* változóban sorban megadott töréspontokat összeköti – sokszöget rajzol. Minden töréspont két *Word* típusú értékből áll: az *X* és *Y* koordinátából (ezek megadásához használható az egység által definiált *PointType* típus).

### **Ellipse** (X,Y:Integer; StAngle,EndAngle,XRadius,YRadius:Word)

G

(*X,Y*) középpont köré *XRadius* vízszintes és *YRadius* függőleges sugárral ellipszist ívet rajzol *StAngle* szögtől *EndAngle* szögig. A legutóbb húzott ív adatai a *GetArcCoords* eljárással lekérdezhetők.

### **FillEllipse** (X,Y:Integer; XRadius,YRadius:Word)

G

Kitöltött ellipszist rajzol (*X,Y*) középpont köré *XRadius* vízszintes, *YRadius* függőleges sugárral. Az eljárás egyben a körvonalat is megrajzolja.

### **FillPoly** (NumPoints:Word; Var PolyPoints)

G

Megfelel a *DrawPoly* eljárásnak, de a sokszöget ki is tölti.

I Ha a töréspontok által definiált vonalak keresztezik egymást, vagy a sokszög nem zárt, a kitöltés lehet hiányos, illetve túlsordulhat.

**FloodFill** (X,Y:Integer; Border:Word)

G

A képernyő azon *Border* színnel határolt területét tölti ki, melynek egyik pontját az (X,Y) koordinátákkal adjuk meg. Ha a határolószín nem folytonos, a kitöltés a területen kívül is folytatódik.

I Az eljárás IBM8514 meghajtóval nem működik.

**GetArcCoords** (Var ArcCoords:ArcCoordsType)

G

Az utolsó Arc vagy Ellipse eljárással rajzolt ív adatait (közép-, kezdő- és végpont koordináták) elhelyezi az *ArcCoords* paraméterben. Az *ArcCoordsType* típust az egység definiálja.

**GetAspectRatio** (Var XAsp,YAsp:Word)

G

Minden grafikus meghajtó és üzemmód rendelkezik bizonyos torzítással, melyet az eljárás által visszaadott értékekből lehet kiszámítani. Az arányossági tényező (*XAsp Div YAsp*) felhasználható a torzítás korrigálására.

**GetBkColor** : Word

G

Visszaadja, hogy a paletta mely színe (Palette[x]) az aktuális grafikus háttérszín.

**GetColor** : Word

G

Visszaadja, hogy a paletta mely színe (Palette[x]) az aktuális grafikus tintaszín.

**GetDefaultPalette** (Var Pal:PaletteType)

G

Visszaadja *Pal* paraméterben azt a palettabeállítást, mely a grafikus rendszer indításakor volt érvényes. A *PaletteType* típust az egység definiálja.

## GRAPH EGYSÉG

**GetDriverName** : String

**G**

Visszaadja az aktív grafikus meghajtó nevét, mint karakterlánc típusú értéket.

**GetFillPattern** (Var FillPattern:FillPatternType)

**G**

Az előzőleg SetFillPattern eljárással beállított kitöltési mintát adja vissza a *FillPattern* paraméterben. A *FillPatternType* típust az egység definiálja. Ha a grafikus üzemmód indítása óta nem hívtuk a SetFillPattern eljárást, akkor \$FF értékkel feltöltött tömböt kapunk.

**GetFillSettings** (Var FillInfo:FillSettingsType)

**G**

Az előzőleg SetFillStyle eljárással beállított kitöltési jellemzőket (stílus és szín) adja vissza a *FillInfo* paraméterben. A *FillSettingsType* típust az egység definiálja. Ha a *FillInfo.Pattern* = *UserFill*, akkor a felhasználói bitmintát a *GetFillPattern* eljárással kérdezhetjük le.

**GetGraphMode** : Integer

**G**

Visszaadja az előzőleg *InitGraph* vagy *SetGraphMode* eljárással beállított grafikus üzemmódot. A függvény értéke 0 és 5 között lehet, az aktív meghajtó függvényében.

**GetImage** (X1,Y1,X2,Y2:Integer; Var BitMap)

**G**

A grafikus képernyő (*X1,Y1*) és (*X2,Y2*) pontok által meghatározott négyzetes területet elmenti *BitMap* változóba. *BitMap* első négy bájtja az elmentett terület vízszintes és függőleges méretét tartalmazza. A szükséges memóriaméret az *ImageSize* függvénnyel lekérdezhető, és nem haladhatja meg a 64K-t.

**GetLineSettings** (Var LineInfo:LineSettingsType)

G

Az előzőleg SetLineStyle eljárással beállított vonaljellemzőket (stílus, felhasználói minta és vastagság) adja vissza a *LineInfo* paraméterben. A LineSettingsType típust az egység definiálja.

! Ha LineInfo.LineStyle <> UserBitLn, akkor LineInfo.Pattern értéke definiálatlan.

**GetMaxColor** : Word

G

Visszaadja a legnagyobb palettaindexet, melyet a SetColor eljárásnak át lehet adni.

**GetMaxMode** : Word

G

Visszaadja az aktuális grafikus meghajtó legnagyobb értékű üzemmódját.

**GetMaxX** : Integer

G

Visszaadja az aktuális grafikus meghajtón és üzemmódon érvényes legnagyobb X pozíció értékét.

**GetMaxY** : Integer

G

Visszaadja az aktuális grafikus meghajtón és üzemmódon érvényes legnagyobb Y pozíció értékét.

**GetModeName** (ModeNumber:Word) : String

G

Visszaadja az aktív meghajtóra érvényes, *ModeNumber* paraméterrel meghatározott grafikus üzemmód nevét, mint karakterlánc típusú értéket.

**GetModeRange** (GraphDriver:Integer; Var LoMode,HiMode:Integer)

A *GraphDriver* paraméterrel meghatározott grafikus meghajtón érvényes

## GRAPH EGYSÉG

---

legkisebb és legnagyobb üzemmód értékét adja vissza a *LoMode* illetve *HiMode* paraméterekben. Ha a *GraphDriver* paraméter értéke érvénytelen, mind *LoMode*, mind *HiMode* -1 értékű lesz.

**GetPalette** (Var Palette:PaletteType)

G

*Palette* paraméterben visszaadja a paletta méretét (azaz a színek számát) és a palettaindexekhez tartozó színeket. A *PaletteType* típust az egység definiálja.

**GetPaletteSize** : Word

G

Visszaadja az aktuális grafikus meghajtó és üzemmód esetében, hogy a paletta hány színt tartalmaz (azaz a paletta méretét).

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

G

Visszaadja az (X,Y) pozícióban lévő képpont palettaszínét.

**GetTextSettings** (Var TextInfo:TextSettingsType)

G

Visszaadja *TextInfo* paraméterben az aktuális szövegjellemzőket (betűtípus, szövegállás, karakterméret, vízszintes és függőleges igazítás), melyeket előzőleg a *SetTextStyle*, *SetTextJustify*, *SetUserCharSize* eljárásokkal állíthattunk be. A *TextSettingsType* típust az egység definiálja.

**GetViewSettings** (Var ViewPort:ViewPortType)

G

*ViewPort* paraméterben visszaadja az aktuális grafikus ablak jellemzőit, melyeket a *SetViewPort* eljárással állíthattunk be. A *ViewPortType* típust az egység definiálja.

**GetX** : Integer G

A grafikus kurzor X koordinátáját adja vissza.

**GetY** : Integer G

A grafikus kurzor Y koordinátáját adja vissza.

**GraphDefaults** G

Alaphelyzetbe állítja a grafikus rendszer következő jellemzőit:

- grafikus ablak – X1=0, Y1=0, X2=GetMaxX, Y2=GetMaxY, vágás=True
- kurzor – X=0, Y=0
- paletta – méret=GetMaxColor+1, szín(nel): Colors[n]=n (n: 0..méret-1)
- háttér- és tintaszín – Palette[0] és Palette[GetMaxColor]
- vonaljellemzők:
  - stílus = SolidLn
  - felhasználói minta = \$FF
  - vastagság = NormWidth
- kitöltési jellemzők:
  - stílus = SolidFill
  - szín = GetMaxColor
  - minta = \$FF tömb
- szövegjellemzők:
  - betűtípus = DefaultFont
  - szövegállás = HorizDir
  - karakterméret = 1
  - vízszintes igazítás = LeftText
  - függőleges igazítás = BottomText

**GraphErrorMsg** (ErrorCode:Integer) : String

Visszaadja az *ErrorCode* paraméterhez tartozó grafikus hiba szövegét.

# GRAPH EGYSÉG

**GraphResult** : Integer

**G**

Az alábbi grafikus műveletek hibakódját adja vissza:

Bar	FloodFill	RegisterBGIDriver	SetLineStyle
Bar3D	GetGraphMode	RegisterBGIFont	SetPalette
ClearViewPort	ImageSize	SetAllPalette	SetTextJustify
CloseGraph	InitGraph	SetFillPattern	SetTextStyle
DetectGraph	InstallUserDriver	SetFillStyle	
DrawPoly	InstallUserFont	SetGraphBufSize	
FillPoly	PieSlice	SetGraphMode	

! Hívása után a grafikus hibakód törlődik!

## A grafikus hibakódok jelentése:

- 0 Nincs hiba
- 1 A grafikus rendszer nem installált
- 2 Grafikus kártya nincs a gépben
- 3 A grafikus meghajtó (.BGI állomány) nem található
- 4 A grafikus meghajtó (.BGI állomány) hibás
- 5 Nincs elég memória a meghajtó (.BGI állomány) betöltéséhez
- 6 Kitöltés közben elfogyott a memória (scan fill)
- 7 Kitöltés közben elfogyott a memória (flood fill)
- 8 A betűtípus (.CHR állomány) nem található
- 9 Nincs elég memória a betűtípus (.CHR állomány) betöltéséhez
- 10 Érvénytelen üzemmód a megadott grafikus kártyán
- 11 Grafikus hiba
- 12 Grafikus B/K hiba

- 13 A betűtípus (.CHR állomány) hibás
- 14 Érvénytelen betűtípus-szám
- 15 Érvénytelen eszközsorszám

**ImageSize** (X1,Y1,X2,Y2:Integer) : Word

**G**

Visszaadja, hogy mekkora memória szükséges az (X1,Y1) és (X2,Y2) koordináták által meghatározott grafikus képernyőterület GetImage eljárással történő elmentéséhez – beleértve a kép méreteit tartalmazó első négy bajtot is. Ha a képernyőterület elmentéséhez 64 kilobájtól nagyobb memóriára lenne szükség, a függvény értéke 0 lesz.

**InitGraph** (Var GraphDriver,GraphMode:Integer; DrivePath:String)

Grafikus üzemmódba állítja a gépet. *GraphDriver* a grafikus meghajtó, *GraphMode* az üzemmód kódját tartalmazza híváskor. A grafikus meghajtó rutint tartalmazó (.BGI) állomány útvonalleírását (aktuális katalógus = ' ') a *DrivePath* paraméterben kell megadni, amennyiben a meghajtót nem töltöttük be a RegisterBGIDriver eljárással. Ha *GraphDriver* változó értéke 0 (Detect), akkor az eljárás a grafikus kártya függvényében indítja az üzemmódot – a legnagyobb felbontással. Az alapbeállításokat lásd a GraphDefaults eljárásnál.

**InstallUserDriver** (Name:String; AutoDetectPtr:Pointer) :Word ;

Felhasználói grafikus meghajtó rutint installál, és visszaadja a hozzátartozó meghajtószámot. *Name* a meghajtó rutin fizikai állománynevéét, *AutoDetectPtr* az opcionális detektáló függvény címét tartalmazza híváskor. Ha nincs detektáló függvény, akkor Nil értéket kell átadni *AutoDetectPtr*-nek. Ha a belső grafikus meghajtó tábla betelt, a függvény értéke -11 lesz.



## GRAPH EGYSÉG

---

**InstallUserFont** (FontFileName:String) : Integer

Felhasználói grafikus betűtípust installál, és visszaadja a hozzátartozó betűtípus-számot. *FontFileName* a betűképeket tartalmazó fizikai állomány nevét kapja meg híváskor. Ha a belső betűtípus tábla betelt, a függvény értéke 0 lesz.

**Line** (X1,Y1,X2,Y2:Integer)

G

(*X1,Y1*) és (*X2,Y2*) pontok közé egyenest húz. Az eljárás hívása után a grafikus kurzor koordinátája változatlan marad.

**LineRel** (DX,DY:Integer)

G

Egyenest húz a grafikus kurzor aktuális pozíciójából az attól (*DX,DY*) relatív távolságra lévő pontba. A grafikus kurzor a vonal végére kerül.

**LineTo** (X,Y:Integer)

G

Egyenest húz a grafikus kurzor pozíciójából az (*X,Y*) pontba. A grafikus kurzor az (*X,Y*) pontba kerül.

**MoveRel** (DX,DY:Integer)

G

A grafikus kurzort az attól (*DX,DY*) relatív távolságra lévő pontba viszi, rajzolás nélkül.

**MoveTo** (X,Y:Integer)

G

A grafikus kurzort az (*X,Y*) pontba viszi, rajzolás nélkül.

**OutText** (TextString:String)

G

*TextString* szöveget megjeleníti az aktuális szövegjellemzőkkel a grafikus kurzor pozíciójában. A grafikus ablakból esetlegesen kieső szövegrészt le-

vágja. A grafikus kurzor csak akkor kerül a szöveg végére, ha vízszintesen ( $Direction=HorizDir$ ) és balra igazítva ( $Horiz=LeftText$ ) írtunk ki.

**OutTextXY** ( $X,Y:Integer$ ;  $TextString:String$ ) **G**

Megfelel az *OutText* eljárásnak, de a szöveget az  $(X,Y)$  pozícióban jeleníti meg.

**PieSlice** ( $X,Y:Integer$ ;  $StAngle,EndAngle,Radlus:Word$ ) **G**

Körcikket rajzol  $(X,Y)$  középponttal, *Radius* sugárral *StAngle* szögű *EndAngle* szöglig, és kitölti azt.

**PutImage** ( $X,Y:Integer$ ;  $Var BitMap$ ;  $BitBit:Word$ ) **G**

Megjeleníti a képernyő  $(X,Y)$  pozíciójában (bal felső sarok) a *BitMap* által meghatározott képet, melyet előzőleg a *GetImage* eljárással mentettünk el. *BitBit* határozza meg a megjelenítés módját (MOV: direkt másolás; XOR, OR, AND: logikai műveletek a másolandó és eredeti képpont között; NOT: inverz kép).

**PutPixel** ( $X,Y:Integer$ ;  $Pixel:Word$ ) **G**

*Pixel* színűre állítja az  $(X,Y)$  koordinátájú képernyőpontot.

**Rectangle** ( $X1,Y1,X2,Y2:Integer$ ) **G**

Négyszöget rajzol.  $(X1,Y1)$ ,  $(X2,Y2)$  két átellenes sarok koordinátái.

**RegisterBGIDriver** ( $Driver:Pointer$ ) :  $Integer$

Feljegyzi a felhasználó által betöltött vagy a programhoz szerkesztett grafikus meghajtót, és visszaadja a hozzárendelt meghajtószámot. *Driver* paraméternek a meghajtórutin memóriabeli címét kell átadni.

## GRAPH EGYSÉG

### RegisterBGIFont (Font:Pointer) : Integer

Feljegyzli a felhasználó által betöltött vagy a programhoz szerkesztett betűtípust, és visszaadja a hozzárendelt betűtípus-számot. *Font* paraméternek a betűtípust leíró bájtsorozat memóriabeli címét kell átadni.

### RestoreCrtMode

G

Visszaállítja a képernyőt az InitGraph eljárás által detektált szöveges üzemmódba. A grafikus rendszer számára lefoglalt heap-területet nem szabadítja fel – a SetGraphMode eljárással a grafikus üzemmód visszaállítható.

### Sector (X,Y:Integer ; StAngle,EndAngle,XRadius,YRadius:Word)

G

(X,Y) középpont köré *XRadius* vízszintes és *YRadius* függőleges sugárral ellipszis cikket rajzol *StAngle* szögtől *EndAngle* szögig, majd kitölti azt.

### SetActivePage (Page:Word)

G

Az aktuális grafikus lapot *Page*-re (0 .. az üzemmódban érvényes maximális lapszám) állítja. Minden további grafikus utasítás az új lapra vonatkozik majd. A lapot a SetVisualPage eljárással tehetjük láthatóvá. Érvénytelen lapszám esetén az eljárás hatástalan.

### SetAllPalette (Var Palette)

G

Megváltoztatja a színpaletta színeit. A *Palette* változó első bájta a hossz (2 .. az üzemmódban érvényes színek száma). Ezt követik az egyes színértékek (-1 .. az üzemmódban érvényes legnagyobb szín száma – lásd Konstansok – Normál és EGA színek). Ha a színérték -1, akkor az illető palettaszín nem változik. A változás a képernyőn azonnal látható lesz. *Palette* változó lehet PaletteType típusú, melyet az egység definiál.

**SetAspectRatio** (XAsp,YAsp:Word)

G

*XAsp* és *YAsp* megadásával korrigálható a képernyő torzítása. *XAsp* DIV *YAsp* az arányossági tényező. Lásd *GetAspectRatio*.

**SetBkColor** (Color:Word)

G

Beállítja a grafikus háttérszintet az aktuális paletta *Color* sorszámú színére (0 .. az üzemmódban érvényes legnagyobb palettaindex). Ha *Color* értéke 0, a háttér színe fekete lesz.

**SetColor** (Color:Word)

G

Beállítja a grafikus tintaszínt a paletta *Color* sorszámú színére (0 .. az üzemmódban érvényes legnagyobb palettaindex).

**SetFillPattern** (Pattern:FillPatternType; Color:Word)

G

Beállítja a *Pattern* paraméterrel megadott felhasználói kiegészítési mintát és annak *Color* paraméterrel meghatározott színét (0 .. az üzemmódban érvényes legnagyobb palettaindex), melyet a *FillPoly*, *FloodFill*, *Bar*, *Bar3D* és *PieSlice* eljárások használnak fel. A *FillPatternType* típust az egység definiálja. Beállítás után nem kell külön meghívni a *SetFillStyle* eljárást.

**SetFillStyle** (Pattern:Word; Color:Word)

G

Beállítja az előre definiált kiegészítési minták közül a *Pattern* paraméterrel meghatározottat (0 .. 12 – lásd Konstansok – Kiegészítési stílus) és annak *Color* paraméterrel megadott színét (0 .. az üzemmódban érvényes legnagyobb palettaindex), melyet a *FillPoly*, *Bar*, *Bar3D* és *PieSlice* eljárások használnak fel. Felhasználói minta (*Pattern* = 12) a *SetFillPattern* eljárással definiálható (alapértelmezés \$FF tömb).

## GRAPH EGYSÉG

---

### **SetGraphBufSize** (BufSize:Word)

Beállítja a kitöltéseknél használt grafikus puffer méretét *BufSize*-ra. Az eljárásnak csak az *InitGraph* hívása előtt van hatása. Az alapméret 4 kilobájt, mely egy kb. 650 törésponttal rendelkező sokszög kitöltéséhez elegendő.

### **SetGraphMode** (Mode:Integer)

Grafikus üzemmódba állítja vissza a gépet és törli a grafikus képernyőt. *Mode* paraméter (0 .. a meghajtón érvényes legnagyobb üzemmód – lásd Konstansok – Grafikus üzemmódok) határozza meg a már inicializált grafikus meghajtó üzemmódját. Az eljárás alaphelyzetbe állítja az összes grafikus jellemzőt (lásd *GraphDefaults*).

### **SetLineStyle** (LineStyle:Word; Pattern:Word; Thickness:Word) **G**

Beállítja a *LineStyle* paraméterben (0 .. 4 – lásd Konstansok – Vonal stílus) megadott vonalstílust és a *Thickness* paraméterrel (1,3 – lásd Konstansok – Vonalvastagság) meghatározott vonalvastagságot. Amennyiben *LineStyle* értéke 4 (*UserBitLn*), *Pattern*-ben kell megadni a felhasználói mintát (a szó 1 értékű bitjel rajzolódna ki), egyébként az eljárás figyelmen kívül hagyja a paramétert.

### **SetPalette** (ColorNum:Word; Color:ShortInt) **G**

Megváltoztatja a színpaletta *ColorNum* (0 .. maximális palettaindex) szerinti színét *Color*-ra (0 .. az üzemmódban érvényes legnagyobb szín száma – lásd Konstansok – Normál és EGA színek). A változás a képernyőn azonnal látható lesz.

**SetRGBPalette** (ColorNum,Red,Green,Blue:Integer)

IBM8514 és VGA grafikus meghajtók esetében a színpaletta *ColorNum* szerinti színének alapszín értékelt állítja be. *Red* a vörös, *Green* a zöld és *Blue* a kék komponens intenzitását határozza meg. Az eljárás *Red*, *Green* és *Blue* alsó bájtjának felső hat bitjét használja fel.

**SetTextJustify** (Horiz,Vert:Word)

G

Grafikus üzemmódban *Horiz* a vízszintes (0 .. 2), *Vert* a függőleges (0 .. 2) – lásd Konstansok – Szöveglgazítás) szöveglgazítást állítja be.

**SetTextStyle** (Font,Direction,CharSize:Word)

G

Beállítja a grafikus szöveg jellemzőit. *Font* (0 .. 4 – lásd Konstansok – Betűtípus) határozza meg a betűtípust, *Direction* (0,1 – lásd Konstansok – Szövegállás) a szövegállást és *CharSize* a méretszorozót (0 .. 10) – ez utóbbi alaphelyzetben bittréképes betűtípus esetén 1, egyéb betűtípusoknál 4. Ha *CharSize* 0 (lásd Konstansok – Felhasználói karakterméret), akkor a *SetUserCharSize* eljárással megadott méret lesz érvényes.

**SetUserCharSize** (MultX,DivX,MultY,DivY:Word)

G

Beállítja a nem bittréképes karakterek vízszintes és függőleges méretét. *MultX:DivX* hányadossal a vízszintes, *MultY:DivY* hányadossal a függőleges alpméretet szorozza meg.

**SetViewPort** (X1,Y1,X2,Y2:Integer; Clip:Boolean)

G

Beállítja az aktív grafikus ablakot. (X1,Y1) az ablak bal felső, (X2,Y2) a jobb alsó sarkát határozza meg. Az ablak bal felső sarka lesz ezután a (0,0) pont. Ha *Clip* False (lásd Konstansok – Grafika vágás), akkor az ablakon kívül-

## GRAPH EGYSÉG

---

re eső rajzolatok is megjelennek, egyébként nem.

**SetVisualPage** (Page:Word)

**G**

A látható grafikus képernyőlapot *Page*-re (0 .. az üzemmódban érvényes legnagyobb lap száma) állítja. Érvénytelen lapszám esetén az eljárás hajtástalan. Az aktuális lapot – amelyen az eljárások rajzolnak – a *SetActivePage* állítja be.

**SetWriteMode** (WriteMode:Integer)

**G**

Beállítja a vonalak rajzolási módját (0:MOV, 1:XOR művelet – lásd Konstanstok – Rajzolási mód), melyet a *DrawPoly*, *Line*, *LineRel*, *LineTo* és *Rectangle* eljárások használnak fel.

**TextHeight** (TextString:String) : Word

**G**

Visszaadja a *TextString* szöveg magasságát – képpontokban.

**TextWidth** (TextString:String) : Word

**G**

Visszaadja a *TextString* szöveg szélességét – képpontokban.

**Típus**<sup>5.5</sup>

**ovrReadFunc** = Function(OvrSeg:Word) : Integer ;

**Konstansok**

Overlay hibakódok:

**ovrOk** = 0 ;

**ovrError** = -1 ;

**ovrNotFound** = -2 ;

**ovrNoMemory** = -3 ;

**ovrIOError** = -4 ;

**ovrNoEMSDriver** = -5 ;

**ovrNoEMSMemory** = -6 ;

**Változók**

**ovrResult** : Integer ;

**ovrFileMode** : Byte ;<sup>5.5</sup>

**ovrLoadCount** : Word ;<sup>5.5</sup>

**ovrReadBuf** : OvrReadFunc ;<sup>5.5</sup>

**ovrTrapCount** : Word ;<sup>5.5</sup>

**ovrResult** értékeinek jelentése:

- 0 Nincs hiba
- 1 Overlay-kezelési hiba
- 2 Az overlay állomány nem található
- 3 Kevés a memória az overlay puffer számára
- 4 Overlay állomány B/K hiba
- 5 Nincs EMS kártya a gépben
- 6 Kevés az EMS memória



## OVERLAY EGYSÉG

---

### Eljárások, függvények

- Az Overlay egység minden eljárása és függvénye állítja az **ovrResult** változót – lásd értékeinek jelentését.

### OvrClearBuf

Törli az overlay puffer tartalmát. Ha egy overlay egységből hívjuk az eljárást, akkor a puffer törlése után ismét betölti az egységet.

### OvrGetBuf : LongInt

Visszaadja az overlay puffer aktuális méretét. Alapértelmezés: a legnagyobb overlay egység mérete (a puffer lehet 64K-nál nagyobb).

### OvrGetRetry : LongInt<sup>5.5</sup>

Visszaadja az overlay puffer próbálkozási területének aktuális méretét. Lásd OvrSetRetry.

### OvrInIt (FileName:String)

Alaphelyzetbe állítja az overlay-kezelőt és megnyitja a *FileName*, overlay egységeket tartalmazó fizikai állományt. Ha *FileName* nem tartalmaz lemezegységnevet és/vagy útvonalleírást, az állományt először az aktuális, majd az .EXE állományt tartalmazó katalógusban keresi (DOS 3.x verziótól), végül a DOS PATH környezeti változójával meghatározott katalógusokban.

I Az eljárást meg kell hívni még az első lapozás, illetve bármely egyéb heap-foglaló művelet végrehajtása előtt.

Az overlay-állomány hozzámásolható a programhoz (Copy DOS parancs /B opcióval)<sup>5.5</sup>. Ekkor overlay-állományként a programot kell megadni, de

fordításkor a D/Standalone debugging opciónak Off állapotban kell lennie!

## OvrInitEMS

Ha az EMS meghajtó aktív és elegendő az EMS memória, betölti abba az overlay állományt. Ezután a lemeztől való betöltést a gyors memóriamásolás váltja fel. A program végén az EMS automatikusan felszabadul.

## OvrSetBuf (BufSize:Longint)

Beállítja az overlay puffer méretét. *BufSize* értékének nagyobb-egyenlőnek kell lennie, mint a kezdeti pufferméret, és kisebb-egyenlőnek, mint  $MemAvail + OvrGetBuf$ .

! Az eljárás hívásakor a heap-nek üresnek kell lennie: szüntessük meg az összes dinamikus változót, zárjuk le a grafikus üzemmódot.

## OvrSetRetry (Size:Longint) <sup>5.5</sup>

Beállítja *Size* méretűre az overlay puffer próbálkozási területét. Ha a próbálkozási terület elegendő egy újabb overlay egység betöltéséhez, akkor az overlay-kezelő nem törli a már betöltött lapot.

## Változók

**Kbd** : Text ;

**CBreak** : Boolean Absolute CheckBreak ;

## Eljárások, függvények

**HighVideo**

**IOResult** : Integer

**LongFilePos**(Var F) : Real

**LongFileSize**(Var F) : Real

**LongSeek**(Var F; Pos : Real)

**LowVideo**

**MemAvail** : Integer

**MaxAvail** : Integer

**NormVideo**

### Megjegyzés:

A Turbo3 egység használatához a Crt egység is szükséges, melyet a USES kulcsszó után még a Turbo3 előtt kell megadni.

**Konstansok****North** = 0 ;**East** = 90 ;**South** = 180 ;**West** = 270 ;**Eljárások, függvények****Arc**(X,Y,Angle,Radius,Color : Integer)**Back**(Dist : Integer)**Circle**(X,Y,Radius,Color : Integer)**ClearScreen****ColorTable**(C1,C2,C3,C4 : Integer)**Draw**(X1,Y1,X2,Y2,Color : Integer)**FillPattern**(X1,Y1,X2,Y2,Color : Integer)**FillScreen**(Color : Integer)**FillShape**(X,Y,FillCol,BorderCol : Integer)**Forwd**(Dist : Integer)**GetDotColor**(X,Y: Integer) : Integer**GetPic**(Var Buffer; X1,Y1,X2,Y2 : Integer)**GraphBackground**(Color : Integer)**GraphColorMode****GraphMode****GraphWindow**(X1,Y1,X2,Y2 : Integer)**Heading** : Integer**HideTurtle****HIRes****HIResColor**(Color : Integer)

## GRAPH3 EGYSÉG

---

Home

NoWrap

Palette(N : Integer)

Pattern(Var P)

PenDown

PenUp

Plot(X,Y,Color : Integer)

PutPic(Var Buffer; X,Y : Integer)

SetHeading(Angle : Integer)

SetPenColor(Color : Integer)

SetPosition(X,Y : Integer)

ShowTurtle

TurnLeft(Angle : Integer)

TurnRight(Angle : Integer)

TurtleDelay(Delay : Integer)

TurtleThere : Boolean

TurtleWindow(X,Y,W,H : Integer)

Wrap

XCor : Integer

YCor : Integer

### Megjegyzés:

A Graph3 egység használatához a Crt egység is szükséges, melyet a USES kulcsszó után még a Graph3 előtt kell megadni.

- 1 Elfogyott a memória
  - ! a memóriában lévő extra programot szüntesse meg (talán kétszer hívta a TURBO-t)
  - ! a programot fordítsa lemezre (C/Destination → DISK)
  - ! a programot szerkessze lemezen (O/C/Link buffer → DISK)
- 2 A fordító azonosítót vár
- 3 Az azonosító nincs deklaráva
- 4 Az azonosító már deklaráva van ebben a blokkban
- 5 Szintaktikus hiba: Illegális karakter
- 6 Hiba a valós típusú konstansban
- 7 Hiba az egész típusú konstansban
  - ! nagy valós egészeket tizedessel kell megadni (pl. 65540.0)
- 8 A karakterlánc konstans túlnyúlik a soron
  - esetleg hiányzik a záró aposztróf
- 9 Túl sok az egymásba ágyazott (Include) forrásállomány
  - maximálisan 5 illetve 15<sup>5.5</sup> lehet
- 10 A forrásállomány nem várt vége
  - END. előtt vége van a programnak (pl. Begin End párhiba)
  - egy Include állomány nem teljes
  - esetleg nem zárt le egy megjegyzést
- 11 Túl hosszú a sor
  - maximum 126 karakteres lehet
- 12 A fordító típusazonosítót vár
- 13 Túl sok a nyitott állomány
  - az egyszerre megnyitható állományok maximális száma a DOS CONFIG.SYS állományában adható meg

## FORDÍTÁSI HIBAKÓDOK

---

- 14 Érvénytelen állománynév vagy útvonalleírás
- 15 Az állomány nincs az aktuális/keresési katalógusban
- 16 Tele van a lemez
  - ! töröljön ki állományokat
  - ! használjon új lemezt
- 17 Érvénytelen fordítási direktíva
  - szintaktikai hiba
  - globális direktívát talált a fordító a programblokkban
- 18 Túl sok az állomány
  - ! Csökkentse az egymásba ágyazott állományok számát, neveik hosszúságát
- 19 Definiálatlan típus Pointer deklarációban
- 20 A fordító változó-azonosítót vár
- 21 Rossz típusdeklaráció
- 22 Túl nagy a struktúrált (összetett) típus
  - maximum 65520 bájt lehet a mérete
- 23 A halmaz alaptípusa hibás
  - az elemek rendszámál 0 és 255 között lehetnek
- 24 Állomány komponense nem lehet állomány és objektum<sup>5.5</sup>
- 25 A String deklarált maximális hossza 1 és 255 közé eshet
- 26 Típussegylezési hiba
  - értékadás során a változó és a kifejezés típusa nem kompatibilis
  - a formális és az aktuális paraméterek típusa nem kompatibilis
  - az Indextípus és az Indexkifejezés nem kompatibilis
  - a kifejezés operandusai nem kompatibilisek
- 27 Az intervallum alaptípusa nem sorszámozott

- 28 Az intervallum alsó határa nagyobb, mint a felső
- 29 A fordító sorszámozott típust vár
- 30 A fordító egész konstanst vár
- 31 A fordító konstanst vár
- 32 A fordító egész vagy valós konstanst vár
- 33 A fordító típusazonosítót vár
- 34 A függvény típusa csak egyszerű, String és Pointer lehet
- 35 A fordító címkét vár
- 36 A fordító a BEGIN kulcsszót várja
- 37 A fordító az END kulcsszót várja
- 38 A fordító egész típusú kifejezést vár
- 39 A fordító sorszámozott típusú kifejezést vár
- 40 A fordító Boolean típusú kifejezést vár
- 41 A műveletnek nem lehetnek ilyen operandusai
- 42 Szintaktikai hiba a kifejezésben
- 43 Hibás értékadás
  - állomány- és típus nélküli paraméter nem szerepelhet értékadó utasítás bal oldalán
  - függvény-azonosítónak csak a függvény blokkjában lehet értéket adni
- 44 A fordító a rekordváltozó egy mező-azonosítóját várja
- 45 A tárgykódot tartalmazó (object) állomány 64K-nál nagyobb
- 46 Hibás EXTERNAL definíció
  - az EXTERNAL eljárás definíciójának nem felel meg PUBLIC definíció a tárgykódot tartalmazó állomány(ok)ban
- 47 A tárgykódot tartalmazó állomány érvénytelen rekordot tartalmaz



## FORDÍTÁSI HIBAKÓDOK

---

- 48 Túl nagy a kódszegmens
  - a program és az egységek kódja egyenként maximum 65520 bájt lehet
- 49 Túl nagy az adatszeglens
  - a program és az egységek globális változóinak együttes mérete maximum 65520 bájt lehet
- 50 A fordító a DO kulcsszót várja
- 51 Érvénytelen PUBLIC deklaráció
- 52 Érvénytelen EXTRN deklaráció
- 53 Túl sok az EXTRN deklaráció
  - maximum 256 lehet
- 54 A fordító az OF kulcsszót várja
- 55 A fordító az INTERFACE kulcsszót várja
- 56 Érvénytelen memóriacím
  - a tárgykódot tartalmazó állományban érvénytelen memóriacímre való hivatkozás szerepel
- 57 A fordító a THEN kulcsszót várja
- 58 A fordító a TO vagy DOWNTO kulcsszót várja
- 59 A deklaráció hiányzik
  - FORWARD direktívával deklarált szubrutin blokkja hiányzik
  - az egység illesztő részében deklarált szubrutin blokkja hiányzik
- 60 Túl sok a programban vagy egységben deklarált szubrutin
  - maximum 512 lehet
- 61 Hibás típuskonverzió
  - itt csak változó típusát lehet konvertálni
  - a változó és a céltípus méretének meg kell egyeznie

- 62** Osztas nullával
- 63** Érvénytelen állománytípus az állománykezelő eljárásban
- 64** Ilyen típusú változó nem szerepelhet Read/Write eljárásban
- szöveges állományból beolvasni csak karakter, egész, valós és karakterlánc típusú változóba lehet
  - szöveges állományba klírnli karakter, egész, valós, karakterlánc és Boolean típusú kifejezéseket lehet
- 65** A fordító mutató típusú változót vár
- 66** A fordító karakterlánc típusú változót vár
- 67** A fordító karakterlánc típusú kifejezést vár
- 68** Az egységek egymásra hivatkozása nem megengedett
- 69** Egységnév ütközés
- a .TPU állományban található egységnév nem egyezik a USES kulcsszó után megadottal
- ! adja ugyanazt a nevet az egységnek és az őt tartalmazó forrásállományoknak
- 70** Az egység módosult
- a program vagy egység által használt egy vagy több egységben változtatás történt a legutóbbi fordítás óta
- ! fordítsa le programját a C/Make vagy C/Bulld menüponttal
- 71** Ilyen egységnév már szerepelt a USES kulcsszó után
- 72** A .TPU állomány hibás
- ! győződjön meg róla, hogy az valóban .TPU állomány-e
- 73** A fordító az Implementációls részt várja
- 74** A konstans típusa nem egyezik a CASE szelektorának típusával
- 75** A fordító rekord típusú változót vár

## FORDÍTÁSI HIBAKÓDOK

---

- 76 Intervallumon kívüli konstans
  - az állandó értéke kiesik a tartományból értékadásnál, paraméterátadásnál vagy tömbindexelésénél
- 77 A fordító állomány típusú változót vár
- 78 A fordító mutató típusú kifejezést vár
- 79 A fordító egész vagy valós típusú kifejezést vár
- 80 A címke nincs a blokkban
  - GOTO utasítással nem lehet kiugrani az aktuális blokkból
- 81 Ez a címke már létezik
- 82 A címke nincs definiálva
- 83 Hiba @ művelet
  - a @ művelet operandusa csak változó, típusos konstans, eljárás és függvény azonosítója lehet
- 84 A fordító a UNIT kulcsszót várja
- 85 A fordító a ; jelet várja
- 86 A fordító a : jelet várja
- 87 A fordító a , jelet várja
- 88 A fordító a ( jelet várja
- 89 A fordító a ) jelet várja
- 90 A fordító az = jelet várja
- 91 A fordító a := jelet várja.
- 92 A fordító a [ vagy a ( jelet várja
- 93 A fordító a ] vagy a .) jelet várja
- 94 A fordító a . jelet várja
- 95 A fordító a .. jelet várja
- 96 Túl sok a változó

- a program vagy egység globális változóinak, valamint egy szubrutin lokális változóinak összmérete nem haladhatja meg a 64K-t
- 97** A FOR utasítás cilusváltozója érvénytelen
- nem egyszerű típusú
  - nincs az aktuális szubrutin deklarációs részében deklarálva
- 98** A fordító egész típusú változót vár
- 99** Típusos állandó nem lehet állomány típusú
- 100** A karakterlánc és a karaktertömb hossza nem egyezik
- 101** A rekord típusú állandó mezősorrendje nem egyezik a deklarációval
- 102** A fordító szövegkonstanst vár
- 103** A fordító egész vagy valós típusú változót vár
- 104** A fordító sorszámozott típusú változót vár
- 105** Hiba az INLINE direktíva blokkjában
- változó hivatkozásnál a < operátor nem megengedett
- 106** A fordító karakter típusú kifejezést vár
- 107** Túl sok az adatáthelyezés
- a futtatható kód relokációs táblája meghaladja a 64K-t
  - a program túl nagy a szerkesztő (linker) számára
- ! bontsa programját több részre, és hívja az alprogramokat az EXEC eljárással
- 112** A CASE utasítás konstansa klesik a megengedett tartományból
- 113** Hibás utasítás
- 114** Az INTERRUPT típusú eljárást nem lehet direkt módon hívni
- 116** A 80x87 valós típusú változókat \$N+ mellett kell fordítani
- 117** Az adott címnek nem feleltethető meg forrásnyelvű utasítás
- 118** Az Include direktíva blokkon belül nem megengedett

## FORDÍTÁSI HIBAKÓDOK

---

- 120 A fordító a NIL kulcsszót várja
- 121 Érvénytelen minősítés
- nem tömböt indexel
  - nem rekord illetve objektum<sup>5.5</sup> típusú változó mezőjére hivatkozik
  - nem mutatóval mutat
- 122 Érvénytelen változóhivatkozás
- 123 Túl sok a szimbólum
- ! fordítsa programját \$D- mellett, vagy tördelje kisebb részekre
- 124 Túl hosszú végrehajtandó rész
- maximum 24K lehet
- ! írjon eljárásokat!
- 126 Állomány típusú paraméter csak változó paraméter lehet
- 127 Túl sok a feltételes szimbólum, vagy túl hosszú a nevük
- 128 A feltételes direktívának nincs párja
- 129 Hiányzik a \$ENDIF direktíva
- 130 Az O/C/Conditional defines menüpont szimbólumai helytelenek
- a Turbo Pascal nulla vagy több azonosítót vár szóközzel, vesszővel vagy kettősponttal elválasztva
- 131 A szubrutinfejek ütköznek
- Az itt deklarált szubrutin feje nem egyezik az INTERFACE részben vagy a FORWARD direktívánál megadottal
- 132 Kritikus lemez hiba
- például nincs a meghajtóban lemez
- 133 A kifejezés kiértékelhetetlen
- 134 A kifejezés helytelen
- 135 Érvénytelen formátum-megadás

- 136** Érvénytelen indirekt hivatkozás
- 137** Struktúrált (összetett) típusú változó itt nem használható
- 138** A System egység nélkül a kifejezés kiértékelhetetlen
- 139** A szimbólum csak a program futásakor válik elérhetővé
- 140** Túlcsoportosítás vagy nullával való osztás valós számításnál
- 141** Overlay szubrutint tartalmazó program lemezre fordítandó
- 142** Itt a @ operátor csak eljárás típusú változóval szerepelhet
- 143** Érvénytelen eljárás vagy függvény hivatkozás
- eljárás hívása kifejezésben
  - a szubrutin \$F+ mellett fordítandó és nem rendelkezhet INLINE vagy INTERRUPT direktívával, ha eljárás típusú változó kapja meg, mint értéket
- 144** Az egységet nem lehet overlay struktúrába szervezni
- valószínű, hogy fordítása nem \$O+ mellett történt
- 147** A fordító objektumot vár<sup>5.5</sup>
- 148** Lokális objektum nem megengedett<sup>5.5</sup>
- 149** A fordító a VIRTUAL kulcsszót várja<sup>5.5</sup>
- 150** A fordító metódus azonosítóját várja<sup>5.5</sup>
- 151** Konstruktor nem lehet virtuális<sup>5.5</sup>
- 152** A fordító konstruktor azonosítóját várja<sup>5.5</sup>
- 153** A fordító destruktorkonstruktor azonosítóját várja<sup>5.5</sup>
- 154** A FAIL eljárás csak konstruktorból hívható<sup>5.5</sup>

## DOS hibák

- 2 Az állomány nem található
  - Reset, Append, Rename vagy Erase hívásakor az állományváltóhoz rendelt fizikai állomány nem létezik, vagy neve érvénytelen
- 3 Az útvonal/katalógus nem található
  - Reset, Rewrite, Append, Rename vagy Erase hívásakor az állományváltóhoz rendelt fizikai állomány katalógusa nem létezik
  - ChDir, Mkdir vagy Rmdir hívásakor az útvonal érvénytelen, vagy a katalógus nem létezik
- 4 Túl sok állomány van egyszerre nyitva
  - Reset, Rewrite vagy Append hívásakor a nyitott állományok a DOS által megengedett számot túllépnék

! állítsa nagyobbra a CONFIG.SYS FILES értéket
- 5 Az állományhozzáférés tiltott
  - Reset vagy Append hívásakor a FileMode változó csak írást engedélyez, és az állomány írásvédett, vagy katalógus
  - Rewrite hívásakor a katalógus betelt, az állomány írásvédett vagy katalógus
  - Rename hívásakor az új néven már létezik állomány, vagy katalógus
  - Erase hívásakor az állomány írásvédett, vagy katalógus
  - Mkdir hívásakor a katalógus betelt, a név már létezik, vagy esz- közt specifikál a név
  - Rmdir hívásakor a katalógus nem üres, a név nem katalógust specifikál, vagy a főkatalógust specifikálja
  - Read, BlockRead hívásakor az állomány nem nyitott olvasásra

## FUTÁSI HIBAKÓDOK

- Write, BlockWrite hívásakor az állomány nem nyitott írásra
- 6 Érvénytelen kezelőszám (handle)
  - az állományváltozó valamilyen oknál fogva megsérült
- 12 Érvénytelen hozzáférési kód
  - Reset vagy Append hívásakor típusos vagy típusnélküli állományra a FileMode változó értéke érvénytelen
- 15 Érvénytelen meghajtószám
  - GetDir hívásakor a meghajtószám érvénytelen
- 16 Az aktuális katalógus nem szüntethető meg
  - Rmdir hívásakor a paraméter az aktuális katalógust specifikálja
- 17 Lemezek között nem lehet átnevezni
  - Rename hívásakor az új név más meghajtót specifikál

### **B/K hibák** (\$!- esetén az IOResult függvénnyel lekérdezhetők)

- 100 Lemezolvasási hiba
  - Read hívásakor a típusos állomány rekordmutatója az állomány vége mögé mutat.
- 101 Lemezírás hiba
  - Close, Write, WriteLn vagy Flush hívásakor a lemez betelt
- 102 Az állományhozzáférés tiltott
  - Reset, Rewrite, Append, Rename vagy Erase hívása előtt elmaradt az Assign
- 103 Az állomány nincs nyitva
  - Close, Read, Write, Seek, Eof, FilePos, FileSize, Flush, BlockRead vagy BlockWrite hívásakor az állomány nincs nyitva
- 104 Az állomány nincs megnyitva olvasásra
  - Read, Readln, Eof, Eoln, SeekEof vagy SeekEoln hívásakor a szö-



- veges állomány nincs megnyitva olvasásra
- 105 Az állomány nincs megnyitva írásra
- Write, Writeln hívásakor a szöveges állomány nincs megnyitva írásra
- 106 Érvénytelen numerikus formátum
- Read, Readln hívásakor a numerikus forma hibás

## Kritikus hibák

- 150 A lemez írásvédett
- 151 Ismeretlen egység
- 152 A lemezegység üzemképtelen
- 153 Ismeretlen parancs
- 154 CRC hiba
- 155 Hiba struktúrahossz lemezegységművelet hívásakor
- 156 Lemez–pozícionálási hiba
- 157 Ismeretlen adathordozó típus
- 158 A szektor nem található
- 159 A nyomtatóból kifogyott a papír
- 160 Hiba eszközre történő írás közben
- 161 Hiba eszközről történő olvasás közben
- 162 Hardver hiba

## Fatális, mindig programleállást okozó hibák

- 200 Oszttás nullával
- 201 \$R+ mellett fordított programok esetében értékhatár–túllépés
- a tömb valamely indexe kiesik a deklarált intervallumból
  - értékadásban a kifejezés értéke kiesik a változó értékkészletéből

## FUTÁSI HIBAKÓDOK

---

- az eljárás vagy függvény valamely aktuális paraméterének értéke kiesik a formális paraméter értékészletéből
  - a virtuális metódus objektumát nem készítettük elő a konstruktor meghívásával<sup>5.5</sup>
- 202 \$S+ mellett fordított programok esetében veremhiba
- az eljárás vagy függvény hívásakor nincs elég hely a veremben  
! növelje meg a verem méretét a \$M direktívával
- 203 Heap túlcsordulási hiba
- New vagy GetMem hívásakor nincs elég hely a heap-ben
- 204 Dispose vagy FreeMem hívásakor érvénytelen mutató-művelet
- a mutató értéke Nil
  - a mutató nem a heap-be mutat
  - a szabad területet nyilvántartó lista nem bővíthető
- 205 Lebegőpontos túlcsordulás
- 206 Lebegőpontos alulcsordulás (csak 80x87 használata esetén)
- 207 Érvénytelen lebegőpontos művelet
- a Trunc vagy Round által kapott valós érték nem alakítható át egészszé a Longint intervallumában
  - az Sqrt paramétere negatív
  - az Ln paramétere nulla vagy negatív
  - a 80x87-es saját veremtára túlcsordult
- 208 Az overlay-kezelő nincs installálva
- a program overlay szubrutint hív, installált overlay-kezelő nélkül  
! hívja az Ovrlnit eljárást a program elején
- 209 Overlay-állomány olvasási hiba

# BILLENTYŰZET KÓDTÁBLA

Billentyű	Normál	Shift	Ctrl	Alt
F1	0 59	0 84	0 94	0 104
F2	0 60	0 85	0 95	0 105
F3	0 61	0 86	0 96	0 106
F4	0 62	0 87	0 97	0 107
F5	0 63	0 88	0 98	0 108
F6	0 64	0 89	0 99	0 109
F7	0 65	0 90	0 100	0 110
F8	0 66	0 91	0 101	0 111
F9	0 67	0 92	0 102	0 112
F10	0 68	0 93	0 103	0 113
F11	0 133	0 135	0 137	0 139
F12	0 134	0 136	0 138	0 140
←	0 75	0 75	0 115	-
→	0 77	0 77	0 116	-
↑	0 72	0 72	-	-
↓	0 80	0 80	-	-
Home	0 71	0 71	0 119	-
End	0 79	0 79	0 117	-
PgUp	0 73	0 73	0 132	-
PgDn	0 81	0 81	0 118	-
Ins	0 82	0 82	-	-
Del	0 83	0 83	-	-
ESC	27	27	27	-
Backspace	8	8	127	-
TAB	9	0 15	-	-

# BILLENTYŰZET KÓDTÁBLA

Billentyű	Normál	Shift	Ctrl	Alt
A	97	65	1	0 30
B	98	66	2	0 48
C	99	67	3	0 46
D	100	68	4	0 32
E	101	69	5	0 18
F	102	70	6	0 33
G	103	71	7	0 34
H	104	72	8	0 35
I	105	73	9	0 23
J	106	74	10	0 36
K	107	75	11	0 37
L	108	76	12	0 38
M	109	77	13	0 50
N	110	78	14	0 49
O	111	79	15	0 24
P	112	80	16	0 25
Q	113	81	17	0 16
R	114	82	18	0 19
S	115	83	19	0 31
T	116	84	20	0 20
U	117	85	21	0 22
V	118	86	22	0 47
W	119	87	23	0 17
X	120	88	24	0 45
Y	121	89	25	0 21
Z	122	90	26	0 44

# BILLENTYŰZET KÓDTÁBLA

Billentyű	Normál	Shift	Ctrl	Alt
0	48	41	-	0 129
1	49	33	-	0 120
2	50	64	0 3	0 121
3	51	35	-	0 122
4	52	36	-	0 123
5	53	37	-	0 124
6	54	94	30	0 125
7	55	38	-	0 126
8	56	42	-	0 127
9	57	40	-	0 128
*	42	42	-	-
+	43	43	-	-
-	45	95	-	-
=	61	43	-	0 131
,	44	60	-	-
;	59	58	-	-
/	47	63	-	-
[	91	123	27	-
\	92	124	-	-
]	93	125	29	-
'	39	34	-	-
‘	96	126	-	-

# ASCII KÓDTÁBLÁZAT

Kód	Char	Kód	Char	Kód	Char	Kód	Char
0	^@ NUL	32	SPC	64	@	96	'
1	^A SOH	33	!	65	A	97	a
2	^B STX	34	"	66	B	98	b
3	^C ETX	35	#	67	C	99	c
4	^D EOT	36	\$	68	D	100	d
5	^E ENQ	37	%	69	E	101	e
6	^F ACK	38	&	70	F	102	f
7	^G BEL	39	'	71	G	103	g
8	^H BS	40	(	72	H	104	h
9	^I HT	41	)	73	I	105	i
10	^J LF	42	*	74	J	106	j
11	^K VT	43	+	75	K	107	k
12	^L FF	44	,	76	L	108	l
13	^M CR	45	-	77	M	109	m
14	^N SO	46	.	78	N	110	n
15	^O SI	47	/	79	O	111	o
16	^P DLE	48	0	80	P	112	p
17	^Q DC1	49	1	81	Q	113	q
18	^R DC2	50	2	82	R	114	r
19	^S DC3	51	3	83	S	115	s
20	^T DC4	52	4	84	T	116	t
21	^U NAK	53	5	85	U	117	u
22	^V SYN	54	6	86	V	118	v
23	^W ETB	55	7	87	W	119	w
24	^X CAN	56	8	88	X	120	x
25	^Y EM	57	9	89	Y	121	y
26	^Z SUB	58	:	90	Z	122	z
27	^[ ESC	59	;	91	[	123	{
28	^\ FS	60	<	92	\	124	
29	^] GS	61	=	93	]	125	}
30	^^ RS	62	>	94	^	126	~
31	^_ US	63	?	95	_	127	DEL

# ASCII KÓDTÁBLÁZAT

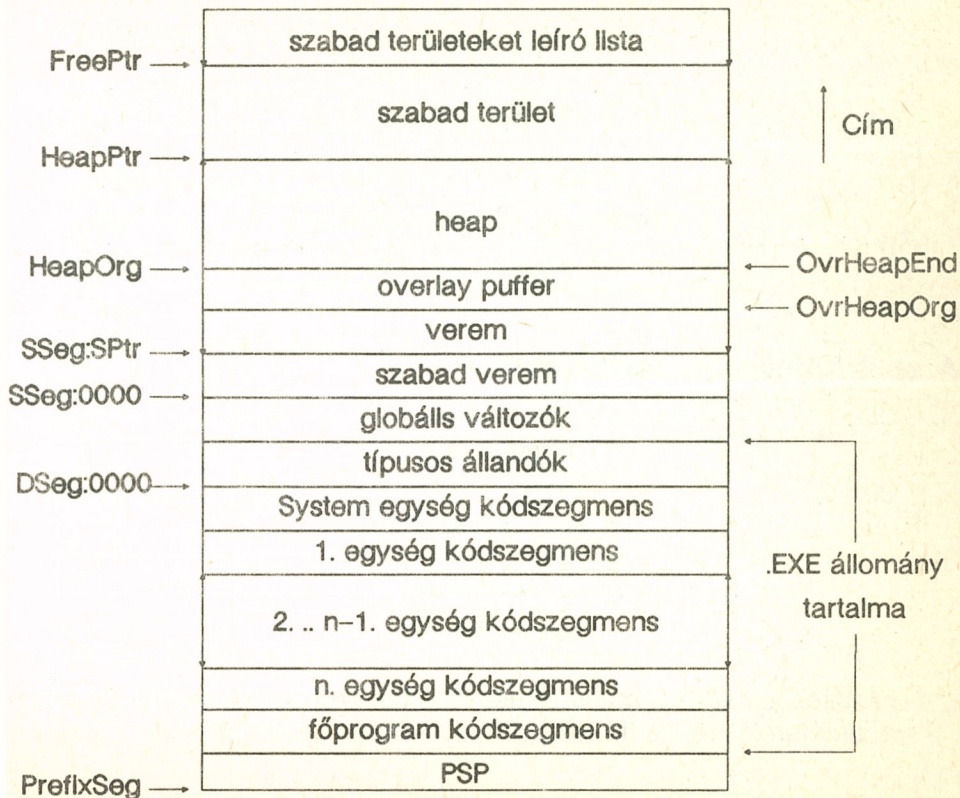
Kód	Char	Kód	Char	Kód	Char	Kód	Char
128	Ç	160	á	192	ˆ	224	α
129	ü	161	í	193	ˆ	225	β
130	é	162	ó	194	ˆ	226	Γ
131	â	163	ú	195	ˆ	227	Π
132	ä	164	ñ	196	ˆ	228	Σ
133	à	165	Ñ	197	ˆ	229	σ
134	â	166	•	198	ˆ	230	μ
135	ç	167	•	199	ˆ	231	τ
136	ê	168	ˆ	200	ˆ	232	Φ
137	ë	169	ˆ	201	ˆ	233	Θ
138	è	170	ˆ	202	ˆ	234	Ω
139	ï	171	ˆ	203	ˆ	235	δ
140	ì	172	ˆ	204	ˆ	236	ø
141	ì	173	ˆ	205	ˆ	237	Ø
142	Ä	174	«	206	ˆ	238	ε
143	Ä	175	»	207	ˆ	239	∩
144	É	176	•	208	ˆ	240	≡
145	æ	177	•	209	ˆ	241	±
146	Æ	178	•	210	ˆ	242	≠
147	ô	179	ˆ	211	ˆ	243	≡
148	ö	180	ˆ	212	ˆ	244	∫
149	ò	181	ˆ	213	ˆ	245	∫
150	ù	182	ˆ	214	ˆ	246	∫
151	ù	183	ˆ	215	ˆ	247	∫
152	ÿ	184	ˆ	216	ˆ	248	∫
153	Ö	185	ˆ	217	ˆ	249	∫
154	Ü	186	ˆ	218	ˆ	250	∫
155	ø	187	ˆ	219	■	251	∫
156	£	188	ˆ	220	■	252	∫
157	¥	189	ˆ	221	■	253	∫
158	¤	190	ˆ	222	■	254	∫
159	f	191	ˆ	223	■	255	∫

# GRAFIKUS ÜZEMMÓDOK JELLEMZŐI

Üzem mód	Felbontás	Lap	Színek
CGAC0	320x200	1	LightGreen,LightRed,Yellow (palette 0)
CGAC1	320x200	1	LightCyan,LightMagenta,White (palette 1)
CGAC2	320x200	1	Green,Red,Brown (palette 2)
CGAC3	320x200	1	Cyan,Magenta,LightGray (palette 3)
CGAHI	640x200	1	2
MCGAC0	320x200	1	LightGreen,LightRed,Yellow (palette 0)
MCGAC1	320x200	1	LightCyan,LightMagenta,White (palette 1)
MCGAC2	320x200	1	Green,Red,Brown (palette 2)
MCGAC3	320x200	1	Cyan,Magenta,LightGray (palette 3)
MCGAMed	640x200	1	2
MCGAHI	640x480	1	2
EGALo	640x200	4	16
EGAHI	640x350	2	16
EGA64Lo	640x200	1	16
EGA64HI	640x350	1	4
EGAMonoHI	640x350	1 v. 2	2
HercMonoHI	720x348	2	2
ATT400C0	320x200	1	LightGreen,LightRed,Yellow (palette 0)
ATT400C1	320x200	1	LightCyan,LightMagenta,White (palette 1)
ATT400C2	320x200	1	Green,Red,Brown (palette 2)
ATT400C3	320x200	1	Cyan,Magenta,LightGray (palette 3)
ATT400Med	640x200	1	2
ATT400HI	640x400	1	2
VGALo	640x200	4	16
VGAMed	640x350	2	16
VGAHI	640x480	1	16
PC3270HI	720x350	1	2
IBM8514Lo	640x480		256
IBM8514HI	1024x768		256

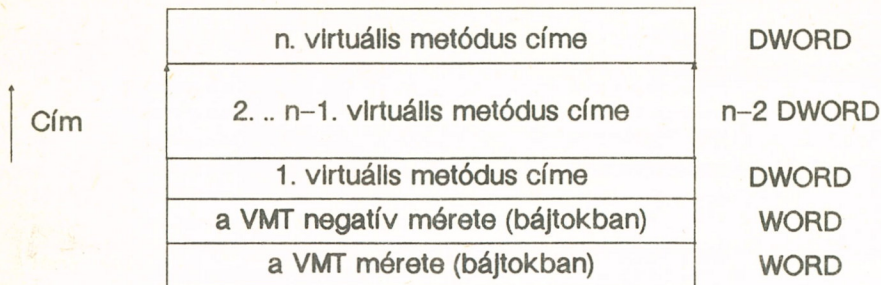


## A program memóriatérképe

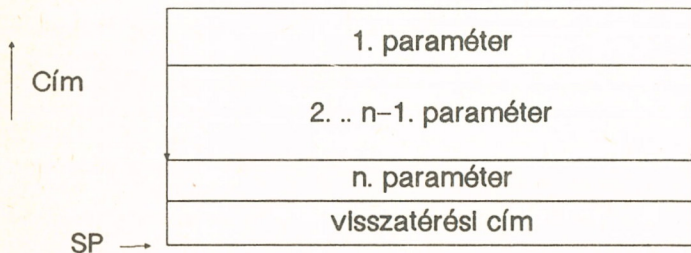


# MEMÓRIATÉRKÉPEK

## A VMT (Virtual Method Table) felépítése<sup>5.5</sup>



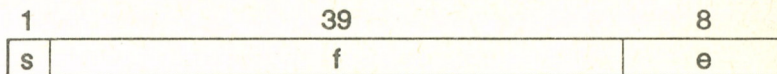
## A szubrutinverem felépítése



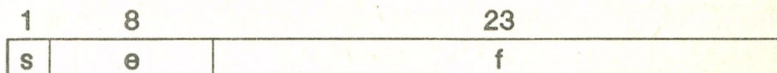
Visszatérés címe NEAR model esetén WORD, FAR model esetén DWORD.  
A paraméterek mérete típusfüggő, de minimum 2 bájtt (WORD).

## Valós számábrázolás

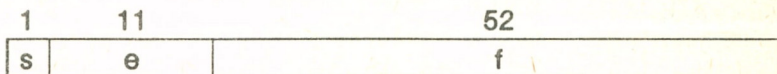
Real - 6 bájt (48 bit)



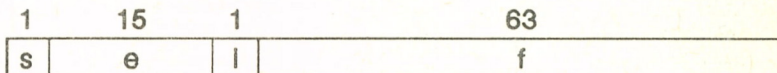
Single - 4 bájt (32 bit)



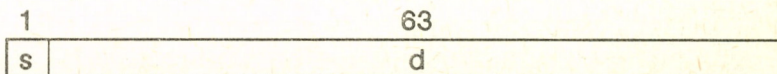
Double - 8 bájt (64 bit)



Extended - 10 bájt (80 bit)



Comp - 8 bájt (64 bit)



Cím

# NYOMTATÓ VEZÉRLŐKÓDOK

<i>FX-1000</i>	<i>saját kód</i>	<i>funkció</i>
ESC @	-----	nyomtató inicializálása
ESC x <i>n</i>	-----	levélminőségű ( <i>n</i> =1) / durva ( <i>n</i> =0) nyomtatás
ESC p <i>n</i>	-----	proporcionális ( <i>n</i> =1) / normál ( <i>n</i> =0) nyomtatás
SI (^O)	-----	sűrített nyomtatás be
DC2 (^R)	-----	sűrített nyomtatás ki
ESC W <i>n</i>	-----	duplaszéles ( <i>n</i> =1) / normál ( <i>n</i> =0) nyomtatás
ESC w <i>n</i>	-----	duplamagas ( <i>n</i> =1) / normál ( <i>n</i> =0) nyomtatás
ESC E	-----	klemelt nyomtatás be
ESC F	-----	klemelt nyomtatás ki
ESC G	-----	duplánütés be
ESC H	-----	duplánütés ki
ESC S <i>n</i>	-----	felső ( <i>n</i> =0) / alsó ( <i>n</i> =1) Index
ESC T	-----	felső vagy alsó Index ki
ESC - <i>n</i>	-----	aláhúzás be ( <i>n</i> =1) / ki ( <i>n</i> =0)
ESC _ <i>n</i>	-----	áthúzás be ( <i>n</i> =1) / ki ( <i>n</i> =0)
ESC 4	-----	dőltbetű be
ESC 5	-----	dőltbetű ki
FF (^L)	-----	puffer ürítése, lapdobás
CR (^M)	-----	puffer ürítése, kocsvissza
LF (^J)	-----	puffer ürítése, soremelés
-----	-----	-----
-----	-----	-----
-----	-----	-----

## I

# 33, 43, 45  
 \$ 32, 44, 45  
 \$A+/- 17, 59  
 \$B+/- 17, 55, 59  
 \$D+/- 13, 14, 15, 17, 60,  
 61, 146  
 \$DEFINE 18, 63  
 \$E+/- 17, 44, 60, 62, 68  
 \$ELSE 64  
 \$ENDIF 64, 146  
 \$F+/- 16, 49, 60, 147  
 \$I 20, 61  
 \$I+/- 16, 60, 69, 77, 150  
 \$IFDEF 64  
 \$IFNDEF 64  
 \$IFOPT 64  
 \$L 20, 61, 66  
 \$L+/- 17, 61  
 \$M 18, 61, 101, 152  
 \$N+/- 17, 44, 54, 60, 62,  
 68, 69, 79, 145  
 \$O 62  
 \$O+/- 16, 62, 147  
 \$R+/- 16, 57, 62, 86, 151  
 \$S+/- 16, 63, 68, 152  
 \$UNDEF 64  
 \$V+/- 17, 63  
 ' 45  
 (\*\$ 32  
 (\*\*) 32  
 ° 30, 52, 54, 56

+ 52, 54, 55, 56, 72  
 - 52, 53, 54, 56  
 .. 44  
 .BAK 3, 11, 19  
 .BGI 3, 122, 123  
 .CFG 3, 4  
 .CHR 3, 122, 123  
 .DOC 3  
 .EXE 3, 15, 20, 22, 132,  
 159  
 .MAP 3  
 .OBJ 3, 20, 61, 66  
 .OVR 62  
 .PAS 3, 11, 12, 19, 61  
 .PCK 3  
 .TP 3, 4  
 .TPL 3  
 .TPU 3, 15, 20, 143  
 / 52, 56  
 @ 34, 35, 48, 52, 54, 56,  
 70, 144, 147  
 /B 4, 5, 6, 132  
 /C 4, 18, 101  
 /D 4, 5, 14, 22  
 /M 4, 5  
 80287 69  
 80387 69  
 8087 69  
 80x86 59, 65  
 80x87 17, 44, 60, 62,  
 65, 68, 69, 145, 152  
 80x88 59, 65  
 := 48, 144

< 53, 66, 145  
 <= 53  
 = 53  
 > 53, 66  
 >= 53  
 ^ 48, 49  
 ^Z 48, 73, 74, 81, 90  
 {\$ 32  
 {} 32

## 5.5

3, 4, 21, 23, 31, 32, 33,  
 35, 36, 37, 39, 41, 43, 46,  
 47, 59, 61, 62, 65, 66, 73,  
 74, 78, 84, 86, 131, 132,  
 133, 139, 140, 146, 147,  
 152, 160

## A

Abs 23, 35, 70  
 Absolute 31, 38, 135  
 Add watch 21, 22, 24, 25  
 Addr 23, 70  
 AH 98  
 AL 98  
 Align data 6, 17, 59  
 ALT 92  
 ALT-B 26  
 ALT-C 26  
 ALT-D 26

# Tárgymutató

ALT-E 13, 26  
ALT-F 26  
ALT-F1 25  
ALT-F3 11, 25  
ALT-F5 14, 25  
ALT-F6 25  
ALT-F9 14, 25  
ALT-O 26  
ALT-R 26  
ALT-X 12, 26  
Always 8, 22  
And 31, 52, 54, 56, 125  
ANDPut 114  
AnyFile 99  
Append 70, 75, 84, 149,  
150  
Arc 115, 137  
ArcCoordsType 110, 117  
Archive 99  
ArcTan 70  
Array 31, 43, 45  
ASCII 43, 45, 55, 72  
Assign 70, 84, 150  
AssignCrt 91  
ATT400 111  
ATT400x 112, 158  
Attr 97  
Autoindent mode 7  
AUX 71  
AX 65, 98, 104, 105

## B

Back 137  
Backspace unindents 7  
backup 3  
Backup files 7, 11, 19  
Bar 115, 122, 127  
Bar3D 115, 122, 127  
Begin 31, 36, 39, 40, 61,  
139, 141  
BH 98  
BINOBJ.EXE 4  
BIOS 90  
BkSlashFill 113  
BL 98  
Black 89, 112  
Black and White 8  
Blink 89  
BlockRead 71, 149, 150  
BlockWrite 71, 150  
Blue 89, 112  
Boolean 43, 52, 53, 141  
Boolean evaluation 6, 17,  
59  
BottomText 113, 121  
BP 65, 98, 104, 105  
Break/watch 22, 26  
Brown 89, 112, 158  
BufEnd 97  
Buffer 97  
BufPos 97  
BufSize 97  
Build 5, 15, 16, 143

BW40 89  
BW80 89  
BX 65, 98, 104, 105  
Byte 6, 44, 49, 53, 54

## C

C40 89  
C80 89  
Call stack 21, 25  
CAPS LOCK 92  
Case 31, 36, 46, 143,  
145  
CBreak 135  
CD 71  
CenterLn 113  
CenterText 113  
CGA 90, 111  
CGAx 111, 158  
CH 98  
Change dir 12  
Char 43, 55  
CharSize 109  
ChDir 71, 149  
CheckBreak 26, 89, 90,  
93, 135  
CheckEOF 81, 89, 90  
CheckSnow 89, 90, 94  
Chr 23, 35, 47, 72  
Circle 115, 137  
CL 98  
Clear all breakpoints 24

ClearDevice 115  
 ClearScreen 137  
 ClearViewPort 115, 122  
 Clip 110  
 ClipOff 113  
 ClipOn 113  
 Close 72, 150  
 CloseDotFill 114  
 CloseFunc 97  
 CloseGraph 115, 122  
 ClrEol 91  
 ClrScr 91  
 CO40 89  
 CO80 89  
 Color 8, 109  
 Colors 109  
 ColorTable 137  
 COM1 71  
 COM2 71  
 COMMAND.COM 101  
 Comp 44, 45, 161  
 Compile 14, 25, 26  
 Complete 6, 17  
 ComStr 98  
 CON 70, 71, 84  
 Concat 72  
 Conditional defines 6, 18,  
 63, 146  
 Config auto save 7, 18  
 CONFIG.SYS 139, 149  
 Const 31, 38, 45, 46, 47  
 Constructor 31, 39  
 Copy 72

CopyPut 114  
 Cos 72  
 CPU86 65  
 CPU87 65  
 CR/LF 48, 74, 80, 81, 84,  
 87  
 CRC 151  
 CRT 70, 91  
 CS 65, 72  
 CSeg 23, 72  
 CTRL 92  
 Ctrl-Break 26, 89, 93,  
 103, 105  
 CTRL-F1 25  
 CTRL-F2 13, 25  
 CTRL-F3 21, 25  
 CTRL-F4 21, 25  
 CTRL-F7 22, 25  
 CTRL-F8 24, 25  
 CTRL-F9 13, 25  
 CurrentDriver 111  
 Current pick file 20  
 Customize colors 9  
 CX 65, 98, 104, 105  
 Cyan 89, 112, 158

## D

DarkGray 89, 112  
 DashedLn 113  
 DateTime 98, 105, 107  
 Day 98

Debug 21, 26  
 Debug Information 6, 13,  
 14, 15, 17, 22, 60  
 Dec 72  
 Default 8  
 Default color set 9  
 DefaultFont 113, 121  
 DEL 74  
 Delay 91  
 Delete 73  
 Delete watch 24  
 DelLine 91  
 Destination 6, 14, 15, 22,  
 139  
 Destructor 31, 39  
 Detailed 7, 18  
 Detect 111  
 DetectGraph 116, 122  
 DH 98  
 DI 65, 98, 104, 105  
 Direction 109, 125  
 Directory 12, 99  
 DirectVideo 89, 90, 94  
 DirStr 98, 102  
 Disk 6, 7, 139  
 DiskFree 100  
 DiskSize 100  
 Display swapping 8, 22  
 Dispose 73, 152  
 Div 31, 52, 54, 56  
 DL 98  
 Do 31, 36, 37, 142  
 DOSError 99, 101, 102

# Tárgymutató

DosExitCode 76, 100,  
101, 104  
DosVersion 100  
DottedLn 113  
Double 44, 45, 161  
DownTo 31, 37, 142  
Draw 137  
DrawPoly 116, 122, 130  
DS 65, 73, 98, 104, 105  
DSeg 23, 73, 159  
DX 65, 98, 104, 105

## E

East 137  
Edit 13, 26  
Edit auto save 7, 19  
Edit watch 24  
Editor buffer size 7  
Editor commands 8  
EGA 19, 111, 126, 128  
EGA64 111  
EGABlack 112  
EGABlue 112  
EGABrown 112  
EGACyan 112  
EGADarkGray 112  
EGAGreen 112  
EGALightBlue 112  
EGALightCyan 112  
EGALightGray 112  
EGALightGreen 112

EGALightMagenta 112  
EGALightRed 112  
EGAMagenta 112  
EGAMono 111  
EGARed 112  
EGAWhite 112  
EGAx 111, 158  
EGAYellow 112  
Ellipse 116  
Else 31, 36  
EmptyFill 113  
EMS 7, 131, 133  
Emulation 6, 17, 60  
End 31, 36, 39, 40, 46,  
61, 74, 139, 141  
EnvCount 101  
EnvStr 101  
Eof 73, 75, 86, 150  
Eoln 74, 150  
Erase 74, 149, 150  
ErrorAddr 67, 69  
ERRORLEVEL 76, 104  
ES 65, 98, 104, 105  
Esc 26  
Evaluate 21, 25  
EXE & TPU directory 8,  
15, 20  
Exec 100, 101, 106, 145  
Exit 12, 26, 74  
ExitCode 68, 69, 76  
ExitProc 68, 69  
Exp 74  
Extended 44, 45, 54, 161

External 31, 38, 61, 66,  
141  
Extrn 142  
ExtStr 98, 102

## F

F1 25  
F2 11, 25, 29  
F3 11, 25  
F4 13, 25  
F5 19, 25  
F6 25  
F7 13, 25  
F8 14, 25  
F9 14, 25  
F10 25, 29  
Fall 74, 147  
False 43  
Far 60, 160  
fAuxiliary 99  
fCarry 99  
FExpand 101  
File 11, 26, 31, 47, 48  
FileMode 68, 149, 150  
FilePos 74, 82, 150  
FileRec 48, 97  
FILES 149  
FileSize 75, 82, 83, 150  
Fill 30, 97  
FillChar 75  
FillEllipse 116



FillPattern 137  
 FillPatternType 110, 118,  
 127  
 FillPoly 116, 122, 127  
 FillScreen 137  
 FillSettingsType 109, 118  
 FillShape 137  
 Find error 15  
 Find procedure 21  
 FindFirst 97, 101  
 FindNext 97, 102  
 Flags 65, 98, 104, 105  
 FloodFill 117, 122, 127  
 Flush 75, 150  
 FlushFunc 97  
 fmClosed 99  
 fmlnOut 99  
 fmlnInput 99  
 fmOutput 99  
 Font 109  
 Font8x8 89  
 For 31, 37, 145  
 Force far calls 6, 16, 60  
 Forward 31, 38, 66, 142,  
 146  
 Forwd 137  
 fOverflow 99  
 fParity 99  
 Frac 75, 77  
 FreeMem 75, 152  
 FreeMin 67  
 FreePtr 67, 159  
 FSearch 102

fSign 99  
 FSplit 102  
 Full graphics save 7  
 Function 31, 38, 39, 43,  
 49  
 fZero 99

## G

GarphDefaults 128  
 Get Info 16  
 GetArcCoords 115, 116,  
 117  
 GetAspectRatio 117, 127  
 GetBkColor 117  
 GetCBreak 103  
 GetColor 117  
 GetDate 103  
 GetDefaultPalette 117  
 GetDir 76, 150  
 GetDotColor 137  
 GetDriverName 118  
 GetEnv 103  
 GetFAttr 103  
 GetFillPattern 118  
 GetFillSettings 118  
 GetFTime 103  
 GetGraphMode 118, 122  
 GetImage 118, 123, 125  
 GetIntVec 103  
 GetLineSettings 119  
 GetMaxColor 119, 121

GetMaxMode 119  
 GetMaxX 119, 121  
 GetMaxY 119, 121  
 GetMem 48, 55, 76, 152  
 GetModeName 119  
 GetModeRange 119  
 GetPalette 120  
 GetPaletteSize 120  
 GetPic 137  
 GetPixel 120  
 GetTextSettings 120  
 GetTime 104  
 GetVerify 104  
 GetViewSettings 120  
 GetX 121  
 GetY 121  
 Go to cursor 13, 25  
 GothicFont 113  
 GoTo 31, 35, 74, 144  
 GoToXY 91  
 GraphBackground 137  
 GraphColorMode 137  
 GraphDefaults 114, 121  
 GraphErrorMsg 121  
 GraphFreeMemPtr 114  
 GraphGetMemPtr 114  
 GraphMode 137  
 GraphResult 114, 116,  
 122  
 GraphWindow 137  
 Green 89, 112, 158  
 GREP.COM 4  
 grError 111

# Tárgymutató

grFileNotFound 111  
grFontNotFound 111  
grInvalidDriver 111  
grInvalidFont 111  
grInvalidFontNum 111  
grInvalidMode 111  
grIOError 111  
grNoFloodMem 111  
grNoFontMem 111  
grNoInnitGraph 111  
grNoLoadMem 111  
grNoScanMem 111  
grNotDetected 111, 116  
grOk 111

## H

Halt 69, 76, 100  
handle 72, 97, 150  
Hardware 6, 17  
HatchFill 113  
Heading 137  
HeapError 67  
HeapOrg 67, 159  
HeapPtr 67, 77, 81, 159  
help 3, 25  
HercMono 111  
HercMonoHI 112, 158  
Hi 23, 35, 76  
Hidden 99  
HideTurtle 137  
High heap limit 18

HighVideo 92, 135  
HiRes 137  
HiResColor 137  
Home 138  
Horiz 109, 125  
HorizDir 113, 121, 125  
Hour 98  
Hz 93

## I

I/O checking 6, 16, 60  
IBM8514 111, 117, 129  
IBM8514x 112, 158  
If 31, 36  
ImageSize 118, 122, 123  
Implementation 31, 40  
In 31, 53, 56  
Inc 76  
Include directories 8, 20, 61  
Indent 30  
InitGraph 118, 122, 123, 126, 128  
InLine 31, 35, 38, 49, 65, 145, 147  
InOutFunc 97  
InOutRes 60, 68, 69, 77  
Input 67, 70, 91  
InstallUserDriver 122  
Insert 77  
Insert mode 7

InLine 92  
InstallUserDriver 123  
InstallUserFont 122, 124  
Int 75, 77  
Integer 44, 53, 54  
Integrated debugging 8, 13, 14, 15, 22  
Interface 31, 40, 142, 146  
InterleaveFill 113  
Interrupt 31, 38, 49, 65, 145, 147  
Intr 98, 104  
IOResult 23, 60, 69, 77, 135, 150  
IP 65

## K

Kbd 135  
Keep 100, 104  
KeyPressed 92, 93

## L

Label 31, 38  
LastMode 89, 90, 94  
LCD 8  
LeftText 113, 121, 125  
Length 23, 35, 77  
LightBlue 89, 112  
LightCyan 89, 112, 158  
LightGray 89, 112, 158

LightGreen 89, 112, 158  
 LightMagenta 89, 112, 158  
 LightRed 89, 112, 158  
 Line 124, 130  
 LineFill 113  
 LineRel 124, 130  
 LineSettingsType 109, 119  
 LineStyle 109  
 LineTo 124, 130  
 Link buffer 7, 18, 139  
 linker 145  
 Ln 77, 152  
 Lo 23, 35, 77  
 Load 11, 25  
 Local symbols 6, 17, 61  
 LongFilePos 135  
 LongFileSize 135  
 LongInt 44, 46, 49, 53, 54  
 LongSeek 135  
 Low heap limit 18  
 LowVideo 92, 135  
 LPT1 71  
 LPT2 71  
 LPT3 71  
 Lst 95  
 LtBkSlashFill 113  
 LtSlashFill 113

## M

Magenta 89, 112, 158  
 Make 5, 13, 14, 15, 16, 25, 143  
 Make use of EMS memory 7  
 MAKE.EXE 4  
 map 3  
 Map file 7, 18  
 Mark 77  
 MaxAvail 23, 78, 135  
 MaxColors 109, 114  
 MaxInt 67  
 MaxLongInt 67  
 MCGA 111  
 MCGAx 111, 158  
 MD 78  
 Mem 23, 49  
 MemAvail 23, 78, 133, 135  
 MemL 23, 49  
 Memory 6, 7  
 Memory sizes 6, 18, 61  
 MemW 23, 49  
 Min 98  
 MkDir 78, 149  
 Mod 31, 52, 54, 56  
 Mode 97  
 Mode for display 8  
 Mono 89  
 Monochrome 8  
 Month 98

MOV 125, 130  
 Move 78  
 MoveRel 124  
 MoveTo 124  
 MS-DOS 65  
 MSDOS 65, 98, 105

## N

Name 97  
 NameStr 98, 102  
 Near 60, 160  
 New 4, 11, 48, 55, 78, 152  
 New value 21  
 Nil 31, 38, 48, 69, 123, 146, 152  
 NONAME.PAS 4, 11  
 None 8, 22  
 NormVideo 92, 135  
 NormWidth 113, 121  
 North 137  
 NoSound 93  
 Not 31, 52, 54, 56, 125  
 NOTPut 114  
 NoWrap 138  
 NUL 71  
 NUM LOCK 92  
 Numeric processing 6, 17, 62

# Tárgymutató

## O

object 3, 31, 43, 46, 61, 66, 141  
Object directories 8, 20, 61  
Odd 23, 35, 78  
Of 31, 36, 45, 46, 47, 55, 142  
offset 79  
Ofs 23, 79  
OpenFunc 97  
Optimal fill 7  
Options 16, 26  
Options for editor 7  
Or 31, 53, 54, 56, 125  
Ord 23, 35, 43, 44, 75, 79  
ORPut 114  
OS shell 12  
Output 67, 70, 87, 91  
OutText 124, 125  
OutTextXY 125  
Overlay 60, 147  
Overlays allowed 6, 16, 62  
OvrClearBuf 132  
OVRCodeList 67  
OVRDebugPtr 67  
OVRDOSHandle 67  
OVREMSHandle 67  
ovrError 131  
OvrGetBuf 132, 133  
OvrGetRetry 132

OVRHeapEnd 67, 159  
OVRHeapOrg 67, 159  
OVRHeapPtr 67  
OVRHeapSize 67  
OvrInit 132, 152  
OvrInitEMS 133  
ovrIOError 131  
ovrLoadCount 131  
OVRLoadList 67  
ovrNoEMSDriver 131  
ovrNoEMSMemory 131  
ovrNoMemory 131  
ovrNotFound 131  
ovrOk 131  
ovrReadBuf 131  
ovrReadFunc 131  
ovrResult 131, 132  
OvrSetBuf 133  
OvrSetRetry 132, 133  
ovrTrapCount 131

## P

Packed 31, 43  
PackTime 105, 106  
Palette 138  
PaletteType 109, 117, 120, 126  
ParamCount 79  
Parameters 8, 13, 20  
ParamStr 79  
PATH 102, 132

PathStr 98, 101, 102  
Pattern 109, 138  
PC3270 111  
PC3270HI 112, 158  
PenDown 138  
PenUp 138  
PI 79  
pick 3, 11, 25  
Pick file name 8, 20  
PieSlice 122, 125, 127  
Plot 138  
Pointer 43, 48, 49, 140, 141  
PointType 109, 116  
Port 49  
PortW 49  
Pos 79  
Pred 23, 35, 73, 79  
PrefixSeg 68, 159  
Primary file 6, 13, 14, 15, 16  
Private 97  
PRN 71  
Procedure 31, 38, 39, 43, 49  
Program 31, 38  
Program reset 13, 25  
PSP 68, 159  
Ptr 23, 35, 48, 80  
Public 61, 66, 141, 142  
Publics 7, 18  
PufPtr 97  
PutImage 125

PutPic 138  
PutPixel 125

## Q

Quit 5, 12  
Quit/save 9

## R

Random 80  
Randomize 68, 80  
RandSeed 68, 80  
Range checking 6, 16, 62  
RD 82  
Read 80, 81, 143, 149,  
150, 151  
ReadKey 93  
ReadLn 81, 150, 151  
README 4  
README.COM 4  
ReadOnly 99  
Real 44, 45, 62, 161  
Record 31, 43, 46  
RecSize 97  
Rectangle 125, 130  
Red 89, 112, 158  
Refresh display 22  
RegisterBGIDriver 122,  
123, 125  
RegisterBGIFont 122, 126  
Registers 98

Relaxed 6, 17  
Release 77, 81  
Remove all watches 24  
REN 82  
Rename 81, 149, 150  
Repeat 31, 36  
Reset 82, 84, 149, 150  
Resize windows 9  
RestoreCrtMode 126  
Result 21  
Retrieve options 21  
Rewrite 75, 82, 84, 149,  
150  
RGB 129  
RightText 113  
Rmdir 82, 149, 150  
Round 23, 35, 83, 152  
Run 13, 25, 26  
RunError 83

## S

SansSerifFont 113  
Save 11, 18, 25  
Save options 21  
SaveIntXX 67, 69, 106  
Screen size 7, 19  
SearchRec 97, 102  
Sec 98  
Sector 126  
Seek 83, 150  
SeekEof 83, 150

SeekEof 83, 150  
Seg 23, 84  
Segments 7, 18  
Self 39, 160  
Set 31, 43, 47, 55  
Set colors 9  
SetActivePage 126, 130  
SetAllPalette 122, 126  
SetAspectRatio 127  
SetBkColor 114, 127  
SetCBreak 105  
SetColor 114, 127  
SetDate 105  
SetFAttr 105  
SetFillPattern 114, 118,  
122, 127  
SetFillStyle 114, 118,  
122, 127  
SetFTime 106  
SetGraphBufSize 122, 128  
SetGraphMode 118, 122,  
126, 128  
SetHeading 138  
SetIntVec 106  
SetLineStyle 114, 119,  
122, 128  
SetPalette 122, 128  
SetPenColor 138  
SetPosition 138  
SetRGBPalette 129  
SetTextBuf 84  
SetTextJustify 114, 120,  
122, 129

# Tárgymutató

SetTextStyle 114, 120,  
122, 129  
SetTime 106  
SetUserCharSize 120, 129  
SetVerify 106  
SetViewPort 114, 120,  
129  
SetVisualPage 126, 130  
SetWriteMode 114, 130  
SHIFT 92  
SHIFT-F10 25  
Shl 31, 52, 56  
Short circuit 6, 17  
ShortInt 44, 54  
ShowTurtle 138  
Shr 31, 52, 56  
Sl 65, 98, 104, 105  
Sin 84  
Single 44, 45, 161  
Size 97, 109  
SizeOf 23, 35, 84  
SlashFill 113  
SmallFont 113  
Smart 8, 22  
Software 6, 17  
SolidFill 113, 121  
SolidLn 113, 121  
Sound 93  
South 137  
SP 68, 85, 104, 105  
SPtr 23, 85, 159  
Sqr 85  
Sqrt 85, 152

SS 85, 104, 105  
SSeg 23, 85, 159  
Stack checking 6, 16, 63  
Stack size 18  
StackLimit 68  
Standalone debugging 8,  
22, 133  
Step over 14, 25  
Str 85, 87  
Strict 6, 17  
String 31, 43, 45, 49, 52,  
140, 141  
Succ 23, 35, 76, 85  
Swap 23, 25, 35, 85  
SwapVectors 101, 106  
switch 25  
SysFile 99  
System 60, 106, 159

## T

Tab 30, 80, 83, 86  
Tab size 7, 19, 30  
Test8087 68, 69  
Text 48  
TextAttr 89, 90, 92, 93  
TextBackground 93  
TextBuf 97  
TextColor 93  
TextHeight 130  
TextMode 90, 94  
TextRec 48, 97

TextSettingsType 109,  
120  
TextWidth 130  
Then 31, 36, 142  
Thickness 109  
ThickWidth 113  
Time 97  
TINST 5  
TINST.EXE 3  
TINSTXFR.EXE 3  
To 31, 37, 142  
Toggle breakpoint 24, 25  
TopOff 113  
TopOn 113  
TopText 113  
TOUCH.COM 4  
TPC.EXE 3  
TPCONFIG.EXE 4  
TPUMOVER.EXE 3  
Trace Into 13, 14, 25  
TriplexFont 113  
True 43  
Trunc 23, 35, 86, 152  
Truncate 86  
TURBO 4  
Turbo Debugger 22  
Turbo directory 4, 8, 19  
TURBO.EXE 3, 5, 6, 8  
TURBO.HLP 3, 19  
TURBO.PCK 8, 20  
TURBO.TP 4, 19  
TURBO.TPL 3  
TurnLeft 138

TurnRight 138  
 Turquoise color set 9  
 TurtleDelay 138  
 TurtleThere 138  
 TurtleWindow 138  
 Type 31, 38  
 TypeOf 23, 86

## U

Unindent 30  
 Unit 31, 40, 144  
 Unit directories 8, 20  
 UnpackTime 103, 107  
 Until 31, 36  
 UpCase 86  
 UPGRADE.DTA 4  
 UPGRADE.EXE 4  
 Use tabs 7  
 User screen 5, 14, 25  
 UserBitLn 113, 119  
 UserCharSize 113  
 UserData 97  
 UserFill 114, 118  
 Uses 31, 38, 40, 143

## V

Val 86  
 Var 31, 33, 38, 63  
 Var-string checking 6,  
 17, 63

VER50 65  
 VER55 65  
 VERIFY 104, 106  
 Version 4.0 color set 9  
 Vert 109  
 VertDir 113  
 VGA 19, 111, 129  
 VGAX 112, 158  
 View next breakpoint 24  
 ViewPortType 110, 120  
 Virtual 31, 39, 66, 147  
 VMT 23, 39, 84, 86, 160  
 VolumID 99

## W

West 137  
 WhereX 94  
 WhereY 94  
 While 31, 36  
 White 89, 112, 158  
 WideDotFill 114  
 WindMax 89, 90  
 WindMin 89, 90  
 Window 94  
 WITH 23, 31, 37, 46  
 Word 6, 44, 49, 53, 54  
 Wrap 138  
 Write 87, 143, 150, 151  
 Write to 12  
 WriteLn 87, 150, 151

## X

X 109, 110  
 X1 110  
 X2 110  
 XCor 138  
 XEnd 110  
 XHatchFill 113  
 Xor 31, 53, 54, 56, 125,  
 130  
 XORPut 114  
 XStart 110

## Y

Y 109, 110  
 Y1 110  
 Y2 110  
 YCor 138  
 Year 98  
 Yellow 89, 112, 158  
 YEnd 110  
 YStart 110

## Z

Zoom windows 7, 19, 25





260 Ft.