

15 Unitok

- unitok
- CRT unit

5.1 Unitok

Mik azok a unitok?

A unit önállóan lefordítható programegység, amely jól definiált kapcsolódási felületen (interface) keresztül kapcsolódik a program más részeihez. Alapvető elv a moduláris programozás során, hogy a modulok belső része (implementation) rejtett marad a külvilág számára. Valójában az egyes unitok újabb eljárásokat, függvényeket, konstansokat és típusokat tartalmaznak, így ezeket a unitokat használva további eljárásokat, függvényeket ill. típusokat használhatunk a programjainkban.

A pascal rendszer sok szabványos modullal rendelkezik, amelyek hasznos típust, konstanst, eljárást és függvényt tartalmaznak. Ezek közül a leggyakrabban használtak:

- **system unit** - ez a unit automatikusan hozzákapcsolódik minden pascal programhoz, ez tartalmazza tartalmazza a szabványos pascal nyelv eljárásait és függvényeit, melyek közül már néhányat megismertedtünk (pl. write, writeln, read, readln, inc, dec, abs, ...),
- **crt unit** - elsősorban a képernyő szöveges üzemmódját támogató eljárásokat és függvényeket tartalmaz. Ezek segítségével lehetőség van a kurzor pozícionálására, a használt színek beállítására. Ezen kívül a modul tartalmaz néhány eljárást, amelyek segítségével a billentyűzet speciális módon érhető el, továbbá a unit segítségével lehetőség adódik hanggenerálásra is.
- **dos unit** - azok az eljárásokat és függvényeket tartalmazza, melyek segítségével hatékonyabban kihasználhatjuk az DOS operációs rendszer lehetőségeit - pl. idő. dátum lekérdezése, mappákkal, állományokkal való munka,
- **graph unit** - a képernyő grafikus üzemmódját támogató típusok, konstansok, eljárások és függvények gazdag készletét tartalmazza,
- **wincrt unit** - window operációs rendszer alatt futó program grafikus ablakában használható crt unit,
- **winmouse unit** - windows operációs rendszer alatt futó program grafikus ablakában használható unit, amely az egér kezelésére szolgál,
- **string unit** - #0 karakterrel végződő dinamikus stringek támogatása.

Ahhoz hogy valamelyik unitot (unitokat) használhassunk a programunkban, meg kell adnunk a **uses** hivatkozás után a unit nevét. Ez alól egyedüli kivétel a system unit, mivel ez automatikusan hozzákapcsolódik minden pascal programhoz.

```
program unitok;
uses crt;
var .... ;
begin
  ....
end.
```

Használhatunk egyszerre több unitot is, ekkor vesszővel választjuk el őket:

```
program unitok;
uses crt, dos, graph;
var .... ;
begin
  ....
end.
```

15.2 CRT unit

A **Crt** unit a billentyőzetről való bevitel és a képernyőre írás valamennyi elemét tartalmazza. Segítségével lekérdezhető a billentyőzet, beállíthatók a képernyő üzemmódjai és színei, ablakok definiálhatók, hangok állíthatók elő stb.

A **Crt** egység néhány fontosabb elemei:

ClrScr; képernyőtörlés
Delay (ms:Word) késleltetés (ms-ban)
GotoXY (X,Y:Byte) kurzor helyzete
KeyPressed : Boolean
ReadKey : Char
TextBackground (szín:Byte) a háttér színe
TextColor (szín:Byte) a betű színe
Window (X1,Y1,X2,Y2:Byte) belső ablakot hoz létre

A **Crt** egységben vannak a színekódok is, ezekkel lehet a képernyőt, az ablakok és az írás színeit beállítani (a **TextBackground** és a **TextColor** eljárások felhasználásával).

A **színekódok** a következők (háttérszín 0..7, betűszín 0..15):

0 Black (fekete)	8 DarkGray (sötétszürke)
1 Blue (kék)	9 LightBlue (világoskék)
2 Green (zöld)	10 LightGreen (világoszöld)
3 Cyan (türkiz)	11 LightCyan (világostürkiz)
4 Red (piros)	12 LightRed (világospiros)
5 Magenta (lila)	13 LightMagenta (világoslila)
6 Brown (barna)	14 Yellow (sárga)
7 LightGray (világosszürke)	15 White (fehér)

Így ha valamilyen színt szeretnénk használni, megadhatjuk a szín számát (pl. 4) vagy helyette a hozzá tartozó konstant, ami nem más mint a szín angol neve (pl. red). A szöveg színét a **textcolor** paranccsal állíthatjuk be. A **textcolor** parancs után addig lesz érvényes a megadott szín, amíg azt nem állítjuk át egy másik színre a **textcolor** paranccsal. Pl.:

```
program Pelda38;
uses crt;
begin
  clrscr;
  textcolor(red);
  writeln('Ez a ket sor');
  writeln('pirossal van irva. ');
  textcolor(10);
  writeln('Ez pedig vilagoszolddel. ');
  textcolor(7);
  writeln('Ez mar az eddig megszokott szurkevel. ');
  readln;
end.
```

A **clrscr** parancs a képernyő letörlésére szolgál. A képernyőtörlés után a kurzor átáll az első sor első oszlopába.

A szöveg (betűk) színéhez hasonló módon megadhatjuk a szöveg háttérszínét is. Ezt a **textbackground** paranccsal tehetjük meg.

```
program Pelda39;
uses crt;
begin
  textbackground(blue);
  clrscr; {fontos, hogy a képernyőtörlés később legyen, mint a háttérszín beáll.}
  textcolor(yellow);
  writeln('Sarga betu kek haterrel. ');
  textbackground(0);
  textcolor(7);
  writeln('Eredeti - szurke betu feketen. ');
  readln;
end.
```

A **gotoxy(oszlop , sor)** parancs segítségével beállíthatjuk a kurzor pozícióját a képernyőn, így segítségével bárhová ki tudunk írni szöveget a képernyőre. Ehhez tudnunk kell, hogy a képernyő felbontása szöveges módban 80x25, tehát 1-től 80-ig állíthatjuk be az oszlopot és 1-től 25-ig a sort. A következő példa letörli a képernyőt, majd kiírja a képernyő közepére a "Hello" szót.

```
program Pelda40;
uses crt;
begin
  clrscr;
  gotoxy(39,13);
  write('Hello');
  readln;
end.
```

Feladat: Készíts egyszerű programot, melyben sárga betűkkel, a 10. sorban a 35. karakterpozíciótól kiíratod a Vezeték és Keresztnevedet, kék háttérszínre!

```
program Pelda41;
uses crt;
begin
  textbackground(1);
  clrscr; {fontos, hogy a képernyőtörlés később legyen, mint a háttérszín beáll.}
  textcolor(14);
  gotoxy(35,10);
  writeln('Vezeteknev Keresztnev');
  readln;
end.
```

A crt unit tartalmaz a billentyűzetről való beolvasáshoz is egy függvényt, ez a **readkey**. Ez a parancs vár egy billentyű megnyomására. A lenyomott billentyűt a **readkey** függvény visszatolja, így azt megjegyezhetjük (egy char típusú változóban).

```
program Pelda42;
uses crt;
var c:char;
begin
  clrscr;
  writeln('Nyomj meg egy betut!');
  c:=readkey;
  writeln('Az ',c,' billentyut nyomtad meg!');
  writeln('Nyomj meg egy tetszoleges billentyut. ');
  readkey;
end.
```