





## szűkebbel évalasztása

- CheckBox
  - lehet van pipálva
  - nincs pipálva
  - szürke

type TCheckBoxState = (cbUnchecked, cbChecked, cbGrayed);

property State: TCheckBoxState;

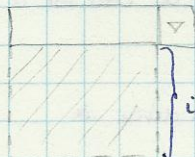
híján dönteni, töltet is lehet pipálni

- RadioButton : - csoportban használjuk
- GroupBox

## listák

- ListBox : - dinamikusan adatkészlet (változhat a mérete)
  - leszűrit a memora
  - string lista, hal ikus kulajdosága van

- ComboBox : - ListBox és az Edit kombinációja



ikusindex : évalasztott ListBox soránna

- CheckBoxList : - az elemek el egy checkbox lent
  - megizsgálható, hogy lehet van-e pipálva

- ListView : - windows használja
  - megjeleníthető íkonként, listaként
  - listikust ell lehet értékeket, beállítások és hozzáadni a listához

- TreeView : - adatokat fa szerűen megjeleníteni tudja
  - sávoként tudunk megjeleníteni
  - ha ⊕-ra kattintunk, megjelenít az alfa, ha ⊖-ra ⇒ összevisszól
  - az elemekhez lehet épreket rendelni (pl.: windows írtó)



## Tartományok:

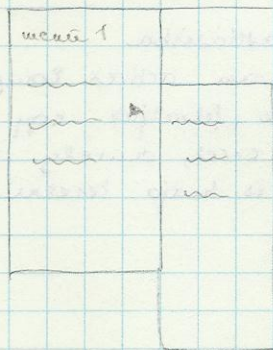
- Scroll Bar
- Track Bar
- UpDown
- Page Scroller
- Scroll Box

UpDown: - fel és lefelé mutató nyílaként  
- delphi-ben van ilyen  
- ListBox-val kombinálva lehet használni pl.: számok között felfelé  
nyitra kattintva

Track Bar: - nagy méretű komponens  
- csúszka, melyet a min és a max között lehet mozgatni

## Menü használata:

- Main Menu: - sorozat egy programban 1 ilyen menüt használunk, általában a főmenü  
- a menüpontok azáltal eseményekhez lehet kötni feladatokat  
- max. 1 mint a submenu-éből.



- minden menüpont a mainmenüben szereplő lista
- lehet hozzá gyorsmenüt rendelni
- van submenu láthatósága, visible (látható v. nem) mező.



aries. ctf. hu / ~ dream

letöltés → delphi → vizsgálat 2

Pen if OpenFileDialog.Execute Then

memo.Lines.LoadFromFile(OpenDialog.FileName);

- SaveDialog:

- FontDialog:

if FontDialog.Execute Then

memo.Font := FontDialog.Font;

- ColorDialog:

- PrintDialog:

- PrinterSetupDialog:

- FindDialog:

- ReplaceDialog:

} RichEditben is van ilyen

### Egyéb fontos komponensek:

- PageControl: - ha a formon több lapot szeretnék megjeleníteni  
- lehet file v. nyomógomb is  
- jól használható → csoportosítható az információk

- ImageList: - kép lista  
- be lehet állítani: 32x32; 64x64  
- lehet használni pl.: Toolbar-nál

- TrackBar:

- ProgressBar: - van min és max értéke  
- állapotjelző → érték változása  
- ha hosszú kivételre érkezik végez a program, akkor  
"alsó" használható

Application.ProcessMessages módszer meghívása → feldolgozza a  
windows üzeneteket → Ne legyen olyan, h. a  
"Program nem válaszol".  
Megszünteti a felületfeszítést is



- DateTimePicker : - olyan, mint a ComboBox, ha rá kattintunk, egy naptár jelenik meg

- MonthCalendar : - kalendárium, vagy helyet foglal

- Animate : - animációt jelenít meg  
(pl.: elmozdítás, másolás... stb.)  
- lehet saját animációt készíteni → lépést órák alatt készítenek  
- .ani kiterjesztésű vannak

- Splitter : - elhelyezve 2 adatablát → a kettőt el lehet helyezni → jól kell összerakni  
- a felhasználó azt látja, h. rugalmas a program

- Image : - lépés tárolásra, megjelenítése alkalmas  
- .bmp, v. .jpg -t tud megjeleníteni

- Shape : - lehet rajolni A-t, 4x-et, köröket... stb.

- ValueListEditor : - többnyelvi programok készítéséhez listákat lehet összeállítani.



# Formok, ablakok

Ez a windows-os programok alapelvei

## Az ablak:

Kenttel, címsorral, rendszerűvel ellátott (val felső sors, a formában alaptól be van. Mozgatható, Beszárható, nagyítható, kicsizálható

- A Formot a TForm osztályból származtatjuk. Az ablak egy memóriabeli képletet jelent, mely a képernyő egy látható eleméhez kapcsolódik.
- A windows egyedi azonosító számokkal, címekkel (handle) tartja nyilván a formákat. Ezek elöljáróként meg őket, pl. melyik képernyő-képletet kell frissíteni a memóriában.

## Az ablakok típusai:

### I) Átlapoló (Overlapped)

- klasszikus ablak
- nem látóolja ezáltal az a főeset egyik formája.

### II) Előugró (Popup)

- párbeszéd és üzenet ablak

pl.: dialógusablak

### III) Gyermekablak (Child)

- Nem kapcsolja el a szülőablak képletét. Milyen vezérlő is virtuális ablakhoz létre.

Ez az MDI technológiával készített formák vonatkozására.

Az MDI formában MDI child-ok nyílnak meg, melyek a



a szülőablaknak kényszerűen hozzájárulni kell. Milyen viselkedés is  
virtuális ablakot hoz létre, pl.: Memo, Edit, Richedit.

#### 10) Alkalmazás ablak:

- Ez fogja össze az összes formát, amit használunk a program-  
ban. Application objektumhoz kapcsolódik az ablak,  
az alkalmazáséhoz tartozó ablakok összefogására szolgál.
- Ez rejtett.

Új üres alkalmazás projektfile-ja:

Begin

Application Initialize;

Application . CreateForm (TForm1, TForm1);

Application . Run;

End.

// Application . Title := TForm1 . Caption;

#### Alkalmazás és form aktiválása:

##### - Kezdeti esemény:

OnCreate → Ez egyszer fut le a form esetében

##### - Aktiválás esemény:

OnActivate → Amikor fut le, ahányszor a fókus a formra  
kerül

#### Form típusok:

- fsNormal: normal SDI ablak, vagy párbeszédablak lesz
- fsMDIChild: A form MDI gyermekablak lesz.
- fsMDIForm: MDI szülőablak, MDI alkalmazás zintablak



- fs StayOnTOP: legfelül, pl. a Windows Feladatkezelő

A StayOnTOP form főkéül csak egy másik alho StayOnTop értéket

### Értesíthetőség:

- fs Sizeable: méretező vastag éret
- fs Dialog: nem méretezhető vastag éret
- fs Single: nem méretezhető, nem vastag
- fs Tool Window: pl. lebegő eszköztárnál
- fs Size Tool Window: ez már méretezhető

### Formok helyzete és mérete:

A position tulajdonság

- no Designed
- no Default
- no Default TopOnly
- no Default Size Only
- no Screen Center

Tabeszédetével értesíthetőségéről a delphi nem figyel a beállításra.

ablak állapota (Windowstate)

- ws Normal
- ws Minimized

### Formok létrehozása:

Automatikus és dinamikus

Project Options → Formok lap

Sok form esetén lehet állítani, hogy melyik form legyen AutoCreate (azaz létrejönne induláskor), és melyik ne. (Available)

pl.: form 2. show (A menü-kor be kell írni még a unit 2-t.)

Available formoknál:

előbb példányosítani kell a formot.



```
var form : TForm3;
```

```
begin
```

```
form := TForm3.create(self);
```

```
form.show
```

```
end;
```

A TForm3 OnClose eseményébe kell említeni a felszabadítását:

```
Action := ca None
```

```
ca Hide - eltűnkh
```

```
ca Free - a memóriától is felszabadítja
```

```
ca Minimize - lezárja
```

És volt a normál formok automatikus és diszkrét vezérlésére és vezérlésére.

példa: if MessageDlg ('Aranja!', mtConfirmation,

```
[mbYes, mbNo], 0) = mmOk then
```

```
Action := caNone
```

```
else
```

```
Action := caFree;
```

Ez a módosított formát jelenti meg, hogy még  
előtt látni, nem látja a főablakot.

Módosítás: Ha a form megkapja a főablakot, addig látja  
ja, amíg be nem zárja.