

Tételek

5. tétel

Az objektum. A formok és jellemzőik.

A, Az objektum: manapság már a legtöbb programozási nyelv támogatja az objektum-orientált vagy objektumközpontú programozást (OOP). Az OOP nyelvek három pillére ..

1. betokozás (encapsulation) – ezt általában az osztályokkal valósítják meg,
2. az öröklődés (inheritance),
3. és a többalakúság (polymorphism vagy késői kötés, azaz late binding).

Ahhoz, hogy az objektum fogalma tiszta, egyértelmű legyen, előtte meg kell határozni az osztály fogalmát.

Osztály egy, a felhasználó által megadott adattípus, melynek állapota (megjelenése, belső adatai) és műveletei (viselkedése, tagfüggvényei) vannak. Az objektum az osztály egy példánya vagy az osztály által meghatározott adattípusba tartozó változó. A ténylegesen használt egyedek az objektumok, melyek a program futása során memóriát foglalnak le. Az objektum és az osztály ugyanúgy viszonyul egymáshoz, mint a típus és a változó.

A Delphi osztálytípusú változói sem közvetlen objektumtárolók, hanem a memóriában elhelyezett objektumokra mutató hivatkozások (reference), vagy mutatók (pointer) Mielőtt használni kezdenénk az objektumot, memóriát kell foglalnunk a számára: ehhez létre kell hoznunk egy új példányt, vagy egy meglévő példányt hozzá kell rendelnünk egy változóhoz:

Var

Obj1, obj2: TMyClass;

Begin

//hozzárendelés újonnan létrehozott objektumhoz

Obj1 := TMyClass.Create;

//hozzárendelés egy már létező objektumhoz

Obj2 := ExistingObject;

A Create tagfüggvény (metódus) az osztály alapértelmezett létrehozó függvényét (konstruktorát) hívja meg, kivéve, ha az osztály felülbírája azt. A Create metódussal tehát az objektum számára memóriát foglalunk le, ez az objektum konstruktora. Az alapértelmezett Tobject.Create metódus minden adatot nullára állít. Ha nem nulla adatokkal szeretnénk létrehozni a példányt, akkor saját konstruktort kell írni.

Ha például a Create konstruktort kiegészítjük néhány paraméterrel, akkor a konstruktort csak ilyen formában használhatjuk:

Type

TDate = class;

Public

Constructor Create (y, m, d: integer);

Ebben az esetben ezt a konstruktort csak ilyen formában használhatjuk:

Var

ADay : TDate;

Begin

//hiba, a fordítás nem fog sikerülni:

ADay := TDate.Create;

//Helyes megoldás:

ADay:=TDate.Create (1,1,2000);

A destruktork olyan tagfüggvény, amely felszabadítja az adott objektum által lefoglalt erőforrásokat. Csak azon objektumok esetében van szükség destrukorra, melyek

5. tétel

létrehozásukkor vagy életük során erőforrásokat foglalnak le. A destruktor a FREE metódus lesz. Ha kétszer hívjuk meg az objektum destruktorát hibaüzenetet kapunk.

B, A formok és jellemzőik: A formokat a Delphiben a TForm osztály határozza meg. A TForm osztály az ablakos vezérlők hierarchiájába tartozik, mely a TWinControl osztállyal veszi kezdetét. A formok rengeteg osztály elemeit öröklik. Általában tervezési időben készülnek el.

Az ablak egy memóriabeli területet jelent, mely a képernyő egy látható eleméhez kapcsolódik.

Ablakok típusai:

1. Átlapolt (Overlapped) => Klasszikus ablakok.
2. Előugró (Popup) => Főként párbeszéd és üzenet ablakok
3. Gyermekablak (Child) => Nem hagyhatja el a szülőablak területét.

Formstílus: a FormStyle tulajdonság lehetővé teszi, hogy válasszunk a normális formok és az MDI alkalmazásokat felépítő ablakok között. (MDI – többablakos alkalmazás, pld.: a jegyzetomb csak egy dokumentumot nyithat meg, ezért ez nem MDI. A Word-ben több ablakot lehet megnyitni, ezért ez MDI.) Beállítási lehetőségei:

1. fsNormal => alapbeállítás, ekkor a form kerettel, címsorral – ezen a szokásos nyomógombokkal – ellátva jelenik meg. Bezárható, nagyítható, ikonizálható, mozgatható.
2. fsStayOnTop => ekkor a form mindig a többi ablak felett marad, kivéve ha találkozik egy másik ugyanilyen tulajdonságú ablakkal.
3. fsMDIForm => ha az MDI alkalmazásban ő a szülő ablak,
4. fsMDIChild => ha az MDI alkalmazásban ő a gyerek ablak.

Szegélystílus: beállítási lehetőségei:

1. bsSizeable => alapbeállítás, ekkor a windows vastag szegéllyel rendelkezik az ablak, méretezhető
2. bsDialog => vastag szegély, az ablak nem méretezhető,
3. bsSingle => olyan főablak, amely nem átméretezhető
4. bsNone => szegély és címsor nélküli ablak. Ritkán használjuk.
5. bsToolWindow => a címsor betűtípusa és bezárógombjai is kisméretűvé válnak.

A formok helyzete: a formok helyzetének megállapítására több tulajdonságot is használhatunk. Ilyen például a Position tulajdonság, amely a form létrehozáskor érvényes kezdeti helyzetét mutatja meg. Beállítási lehetőségei:

1. poDesigned => alapértelmezett, használatakor a form tervezésének helyén jelenik meg, és saját méretezési (width és height) és elhelyezési (left és top) tulajdonságait használja,
2. poDefault, poDefaultPosOnly, poDefaultSizeOnly => a Delphi a rendszer tulajdonságait is figyelembe veszi, így egy megfelelő jelző használatával a Windows képes az új ablakokat átlapoltan elhelyezni, illetve méretezni. Ilyenkor a tervezéskor beállított helyzet és méret érvényét veszti, de ha a felhasználó kétszer indítja el ugyanazt az alkalmazást, a kapott ablakok nem fedik egymást.
3. poScreenCenter => a form a képernyő közepére kerül, mérete pedig a tervezéskori érték marad.

5. tétel

Formok állapota: tervezéskor a WindowState (ablak állapota) tulajdonság beállításával elérhetjük, hogy a program indításakor az adott ablak teljes (wsMaximized), ikon (wsMinimized) és normál (wsNormal) méretben jelenjen meg.

Formok méretezése: ha elkészítünk egy formot, választhatunk rögzített méretű szegélyt, de lehetővé tehetjük az átméretezést is, automatikus gördítősávokkal biztosítva a form teljes felületének elérhetőségét. (Minderre szükség lehet akkor is, ha a felhasználó képernyőfelbontása jóval kisebb, mint a mienk.)

A form ScaleBy tagfüggvénye lehetővé teszi a form és egyes komponensei méretezését. Amennyiben azt szeretnénk, hogy a form átméretezze az ablakát, mindkét esetben fontos, hogy AutoScale tulajdonságot false-ra állítsuk

1. kézi méretezése: ha a formot komponenseivel együtt szeretnénk átméretezni, a ScaleBy tagfüggvényt kell használni. Két egész típusú paramétert kell megadni, mely a nagyítási arány számlálóját és nevezőjét adja meg. Pld.: ScaleBy (3,4) => ebben az esetben a form az eredeti háromnegyedére csökken. Megadhatjuk százalékban is ScaleBy(75,100)
2. automatikus méretezés=> a Delphi indításkor lekérdezi a rendszer képernyőfelbontását, és az értékeket tárolja a Screen objektum PixelsPerInch tulajdonságában. Ez nem a felbontáshoz kapcsolódik, hanem az aktuális rendszerbetűtípushoz. (A felhasználó általában azért változtatja meg a rendszerbetűtípus méretét, hogy az olvasás könnyebbé váljon, de ekkor jogos igény az, hogy más alkalmazások is kövessék ezt a módosítást. Amennyiben ez nem történik meg, a program megjelenése nagymértékben elüthet a felhasználói felületétől, néha használhatatlanná is válhat.) A PixelsPerInch leggyakoribb értékei a 96 (kis karakterek) és a 120 (nagy karakterek), de más érték is alkalmazható. Azonban ennek ellenére adódhatnak problémák, amelyek elkerülhetők, ha:
 - a form Scaled tulajdonságát tru-ra állítjuk,
 - csak TruType betűtípusokat használunk (Times NR, Arial)
 - a fejlesztéskor kis betűket (96 dpi) használunk,
 - amennyiben nem csak a formban található vezérlőket, hanem magát a formot is méretezni akarjuk, állítsuk az AutoScroll tulajdonságot hamisra.
 - Helyezzük a formot a bal felső sarokba, vagy a képernyő közepére (poScreenCenter érték segítségével)

Formok létrehozása: függetlenül attól, hogy a formot 'kézzel' vagy automatikusan hozzuk létre, az alábbi események következnek be az alábbi sorrendben:

1. az OnCreate esemény jelzi, hogy éppen létrehozuk a formot,
2. Az OnShow tudatja, hogy a form éppen megjelenik a képernyőn,
3. az OnActivate jelzi, hogy a form az alkalmazás aktív formjává válik,
4. más események, mint az OnResize és az OnPaint, olyan műveletekről tudósítanak, melyeket a Delphi indításkor mindig elvégez, de a program futása során újra és újra ismétlődnek.

Formok bezárása:

Az alkalmazás ablak:

- Application objektumhoz kapcsolódik az ablak az alkalmazáshoz tartozó ablakok összefogására szolgál.

3. tétel

- Ez rejtett.

Párbeszédablak keretstílusnál nem figyeli a beállítást a Delphi.

Egyszerű párbeszédablakok

- `MessageDlg`
- `MessageDlgPos`
- `ShowMessage`
- `ShowMessageDlg`
- `Application.MessageBox`
(Win API `MessageBox` fgv.)
- `InputDialog`