

6. tétel

Privát. Több form használata

A, Privát: az adatrejtés (data hiding) az egységbezárás alproblémája. A célját az alábbi három dologban lehetne összefoglalni:

1. az objektum képes legyen adatokat tárolni
2. azokat a külvilág képes legyen lekérdezni és megváltoztatni,
3. de ez csak ellenőrzött módon történhessen meg.

Az elérhetőség szempontjából megkülönböztetünk:

1. az objektum belvilágát: a saját metódusai tartoznak ide,
2. Külvilág: minden más egyéb metódus.
 - a, távoli külvilág: a példányokat készítő programmodulok (pld.: a főprogram)
 - b, közeli külvilág: az osztály leszármazottjai, s azok metódusai

Az adatrejtés szempontjából különféle hozzáférési lehetőségeket különböztetünk meg:

A, Attribútumokkal kapcsolatos:

1. **Private** (privát)=> a külvilág nem férhet ezen attribútumokhoz hozzá (az osztály magánügye). Ezek általában segédváltozók, segédmezők.
2. **Public** (publikus)=> a program tetszőleges pontján hozzáférhetőek. Olyan attribútumok, amelyek jellegüknél fogva nem igényelnek speciális szabályozást, azot tetszőleges időben és értékre történő megváltoztatása nem okoz, nem okozhat problémát az objektum működésében.
3. **Protected** (védett)=> olyan attribútumok, melyekhez a távoli külvilág nem férhet hozzá (számára private), de a közeli külvilág, leszármazott osztályok metódusai hozzáférhetnek (számukra public). Tehát korlátozott láthatósággal rendelkeznek.

B, Metódusokkal kapcsolatos: a metódusokhoz való hozzáférést is a fentiek szerint osztályozzuk, de itt az adott metódus meghívhatóságát jelöli:

1. **public**=> ezen metódusokat a távoli külvilág is meghívhatja,
2. **private**=> ezen metódusokat a külvilág nem hívhatja meg, csak az adott osztály metódusai hívhatják meg,
3. **protected**=> olyan metódusok, melyeket a távoli külvilág nem hívhat meg, de a közeli külvilág, a leszármazott osztályok metódusaiból meghívhatóak.

Az osztály mezőinek általában privátnak kell lenniük, a tagfüggvények ezzel szemben többnyire nyilvánosak

Pld.:

Type

TDate = class;

Private

Month, Day, Year: integer;

Public

Procedure SetValue (y,m,d: integer);

Procedure SetValue (NewDate:TDateTime);

Function LeapYear: boolean;

Function GetText:string;

Procedure Increase;

End;

B, Több form használata: a Delphi alkalmazások általában több ablakot tartalmaznak. Az ablakok közül egy adja az alkalmazás menüvel ellátott főablakát, míg a továbbiak párbeszéd- vagy gyermekablakok, amelyek menüpontok kiválasztásakor, vagy nyomógombok lenyomásával jelennek meg.

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

Ablakok fajtái:

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

Párbeszédablakok: olyan form, amelyik a logikailag összetartozó vezérlőket gyűjti egybe és a program futása során akár többször is megjeleníthető.

Párbeszédablakok fajtái:

1. Modális és nem modális párbeszédablakok=> a modális párbeszédablakot a felhasználónak be kell zárnia az aktivizáló ablakhoz való visszatérés előtt. A modális párbeszédablakot a ShowModal függvény metódussal jelenítjük meg. Amennyiben egy párbeszédablak modális módon jeleni meg, akkor formjának ModalResult tulajdonságában definiálhatjuk az ablak zárásakor, a ShowModal függvény által visszaadott értéket. Az egész típusú ModalResult tulajdonságot előre definiált konstanssal – mrNone (0 ez az alapértelmezett), mrOK, mrCancel – is megadhatjuk.

Ha a párbeszédablakot úgy jelenítjük meg, hogy bármikor átkapcsolhatunk másik (párbeszéd)ablakra, akkor nem modális párbeszédablakról beszélünk. Ezt a Show metódus jeleníti meg. Ha azt szeretnénk, hogy a megjelenítendő ablak az összes többi ablak fölé jelenjen meg, akkor a form FormStyle tulajdonságát fsStayOnTop-ra kell állítani.

2. Üzenetablak=>

a) a ShowMessage eljárással egyszerű üzenetablakot jeleníthetünk meg: **procedure ShowMessage (const Msg: string).**

b) A MessageDlg eljárás összetettebb üzenetablakot jelenít meg a képernyő közepén. Pld.:

```
function MessageDlg(const Msg: string)
```

```
    Atype: TMsDlgType;
```

```
    AButtons: TMsgDlgButtons;
```

```
    HelpCtx: Longint; Word;
```

(A TMsDlgType paraméter értékei:

- 1, mtWarning: figyelmeztető ablak ikon: felkiáltójel,
- 2, mtError: hibaüzenet ablak ikon: piros körben, fehér x,
- 3, mtInformation: információs ablak, ikon: sárga/fehér körben egy i bötű,
- 4, mtConfirmation: jóváhagyó ablak, ikon: sárga/fehér körben egy ?,
- 5, mtCustom: az alkalmazás neve jelenik meg, a bitkép nélküli ablak fejlécében

3. Egyszerű adatbeviteli párbeszédablakok=> a Delphiben két függvény is lehetővé teszi, hogy párbeszédablak segítségével adatot kérjünk a felhasználótól:

a) InputBox: string típusú függvény visszatérési értéke tartalmazza a kért szöveget.

```
Function InputBox (const ACaption, APrompt, ADefault: string) : string;
```

(Az ACaption az üzenetablak fejlécét határozza meg, míg az APrompt a bekérendő szövegre utal. Az ADefault tartalmazza azt a szöveget, amely az ablak első megjelenésekor a szövegmezőben látható lesz. Az ok. Gomb

5. tétel

választása esetén az InputBox fgv. a szövegmezőben megadott szöveggel tér vissza, egyébként pedig az alapértelmezett szöveggel.

4. Általános párbeszédablakok: a Dialogs palettalapon komponensként érjük el az egyes ablakatípusokat. A párbeszédablak-komponensek használatában közös vonás, hogy tulajdonságok beállításával paraméterezzük az ablak megjelenését. Alapvető szerepe van a komponensek Execute metódusának, amely megjeleníti a szóban forgó párbeszédablakot.

Fajtái:

A, fájlkezelő általános párbeszédablakok:

- OpenDialog: fájlnyitás párbeszédablak felületén kiválaszthatunk egy állományt, melyet aztán a program megnyithat olvasásra,
- SaveDialog: állományok mentésére a 'Mentés másként' fejlécű párbeszédű ablakot jeleníti meg. Használata megegyezik az előzőével.
- OpenPictureDialog és a SavePictureDialog: képfájl megnyitására, mentésére szolgál.

B, Betűtípus beállító párbeszédablak:

- FontDialog: a betűtípus kiválasztására szolgáló komponens. Alkalmazásával a Betűtípus párbeszédablakban a kiválasztott Betűtípus, Betűstílus és Méret adatokat lehet kiválogatni.

C, Színbeállító általános párbeszédablak:

- ColorDialog: megjeleníti a színkiválasztási párbeszédablakot.

D, Nyomtatással kapcsolatos általános párbeszédablak:

- PrintDialog: komponens feladata a Nyomtatás párbeszédablak megjelenítése.
- PrintSetupDialog: komponens közvetlenül a nyomtatóvezérlő program adatait állítja be (Tájolás, Papírméret, Tulajdonságok)

E, Szövegkezelő általános párbeszédablakok

- FindDialog: komponens megjeleníti a Keresés (Find) párbeszédablakot, amellyel szövegkereséshez szükséges adatokat kérhetünk be a felhasználótól.
- ReplaceDialog: komponens hasonló a FindDialog-hoz, célja azonban a szövegek cseréjének támogatása.