3. tétel
Öröklés, a VCL alapvető elemei

A, Öröklés: Az OOP nyelvek három pillére a betokozhás (encapsulation – ezt általában osztályokkal valsóítják meg!), az öröklődés (inheritance) és a többalakúság (polymorphism).

Gyakran előfordul, hogy egy korábban megírt osztály némileg módosított változatát szeretnénk használni. Ha lemásoljuk az eredeti osztályt, majd a másolatban végezzük el a szükséges módosításokat (rossz programozói gyakorlat!), akkor megkétszerezzük a kódot. Ehelyett kell használni az OOP programozás egyik legfontosabb szolgáltatását, az öröklést. (inheritance).
Ha örökléssel szeretnénk egy osztályt létrehozni, csak annyi a dolgunk, hogy az új osztály bevezetésekor feltüntessük az eredeti osztály nevét. Például amikor új formot hozunk létre:

```
Type
        TForm1 = class(TForm);
        End;
```

Ez azt jelzi, hogy a Tform1 osztály örökli a Tform osztály összes tagfüggvényét, mezőjét, tulajdonságát és eseményét. Így a TForm osztály összes nyilvános tagfüggvényét alkalmazhatjuk a TForm1 típusú objektumokra is, míg a TForm néhány tagfüggvényét saját őseitől örökli, és ez így megy le egészen a TObject alaposztályig.
Például származtassunk egy osztálya a TDate osztályból, és változtassuk meg annak GetText függvényét:

```
Type
        TNewDate = class (TDate)
        Public
                Function GetText:string;
        End;

Function TnewDate.GetText :string;
Begin
        GetText := FormatDateTime ('dddd',fDate);
End;
```

Öröklődés és típusmegfelelőség: a Pascal szigorúan típusos nyelv, ami azt jelenti, hogy nem adhatunk például egész értéket egy boolean változónak, legalábbis típusátalakítás nélkül nem. A szabály az, hogy két érték csak akkor feleltethető meg egymásnak, ha azonos az adattípusuk.
Ha ugyanazt a típust két egységben is meghatározzuk, a típusok annak ellenére sem lesznek egymásnak megfeleltethetők, hogy a nevük megegyezik.
Kivételt képeznek ez alól az osztálytípusok. Ha létrhozunk egy osztályt (például TAnimal), és származtatunk belőle egy másik osztályt (például TDog), akkor egy TDog típusú objektumot értékül adhatunk egy TAnimal típusú változónak. Az általános szabály az, hogy minden olyan helyen használhatjuk a leszármazott osztály objektumait, ahol az ősosztály objektumait használhatnánk. A dolog azonban fordítva nem működik, tehát az ősosztály objektumait nem használhatjuk a leszármazott osztály objektumai helyett.
Például:

```
Var
        MyAnimal: TAnimal;
        MyDog: TDog;
Begin
        MyAnimal := MyDog;  // Ez jó
```

MyDog := MyAnimal // Ez hibás!

Vizuális formöröklés: ha például több hasonló formot készítünk, különböző eseménykezelőkkel, azonban sok, apróbb változás van a különböző formokon, akkor ahelyett, hogy a TForm alapján hoznánk létre egy formot, származtathatjuk azt egyszerűen egy másik formból is, majd kiegészíthetjük új komponensekkel, illetve módosíthatjuk a meglévő komponensek tulajdonságait.

De mi a vizuális formöröklés igazi előnye?

Az attól függ, milyen alkalmazást fejlesztünk. Ha tehát, a program számos formot tartalmaz, és ezek között sok hasonló van, illetve sok hasznlja ugyanazt a komponenseket, akkor létrehozhatunk egy alapformot, és a többi formot errel az alapformra építhetjük.

A vizuális formöröklés alkalmas arra is, hogy az alkalmazás felületét egyes kódrészletek újbóli beírása nélkül alakíthassuk a különböző programváltozatok igényeihez. Mindössze annyit kell tennünk, hogy formjainkat a megfelelő programhoz tartozó szhülőformból származtatjuk.

A vizuális formöröklés legjelentősebb előnye, hogy később bármikor lecserélhetjük az eredeti formot, és vele együtt természetesen a származtatott formokat is.

A vizuális formöröklés szabályai nagyon egyszerűek. Az alosztálybeli form rendlekezik a szülőform összes komponensével, és ezek mellett új komponenseket is tartalmazhat. Az alaposztály komponenseit nem tudjuk eltávolítani, de –mivel vizuális vezérlőelemről van szó – láthatatlanná tehetjük azokat. A lényeg az, hogy könnyedén megváltoztathatjuk az örökölt komponensek tulajdonságait.

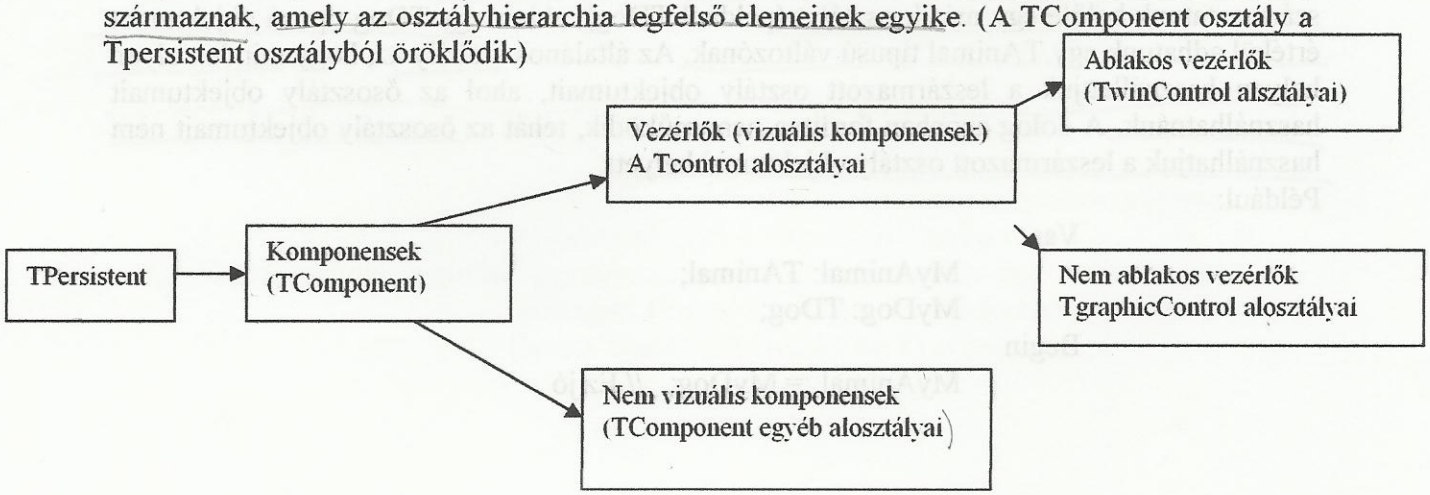
B, a VCL (Visual Component Library) alapvető elemei: a Delphi két vizuális osztálykönyvtárral rendelkezik:

1. a hagyományos Windows könyvtár (VCL).
2. egy rendszerfüggetlen könyvtár (CLX)

A VCL vizuális része tulajdonképpen a Windows alkalmazás-programozási felületének (API) 'beburkolására' szolgál. Megtalálhatjuk itt a Windows vezérlők (Például gombok és szövegmezők) burkolóit, a közös vezérlőket (TreeView és ListView) és számos olyan Delphi vezérlőt is, amely szorosan kapcsolódik a Windows ablakfogalmázhoz. A Tcanvas osztály az alapvető grafikus hívásokat burkolja be, így egyszerűen rajzolhatunk egy ablak felületére.

A D. számtalan függvényt és eljárást tartalmaz, a vizuális Delphi programozás valódi ereje azonban a programnyelvvel kapott hatalmas méretű osztálykönyvtárban rejlik. A Delphi szabványos oszálykönyvtára több száz osztályt tartalmaz, melyek ezernyi tagfüggvényt rejtenek.

Az idők folyamán a Delphi programozók a Borland dokumentációjában javasolt nevekkel hivatkoztak a VCL különböző részeire, így ezek a nevek lassan a különféle komponenscsoportok azonosítóivá váltak. A komponensek a TComponent osztályból származnak, amely az osztályhierarchia legfelső elemeinek egyike. (A TComponent osztály a Tpersistent osztályból öröklődik)



Vezérlők: a Tcontrol osztály összes leszármazottja. Van helyzetük és méretük a képernyőn; tervezés közben ugyanott láthatjuk ezeket az elemeket a formon, ahol futásidőben megjelennek majd. Két altípusa létezik:

1, ablakos vezérlők: az operációs rendszer ablakain alapuló vezérlők. A VCL TwinControl osztálya egy ablakleírót (handle) is tartalmaz, ez a szám a belső ablakszerkezet egy elemére mutat. A felhasználó szempontjából nézve az ablakos vezérlők megkaphatják a bemeneti fókuszt, néhányuk pedig további vezérlőket is tartalmaz. Az ablakos vezérlőket további két csoportra bontjuk:

a, szabványos Windows vezérlők burkolóira,

b, egyedi vezérlőkre

2, grafikus vezérlők. (nem ablakos vezérlők) olyan vizuális komponensek, amelyek nem kapcsolódnak az op.rendszer ablakaihoz. Ezek a komponensek nem tartalmaznak ablakleírót, nem kaphatják meg a fókuszt és nem tartalmazhatnak további vezérlőket sem. Ilyen vezérlő például a Label és a SpeedButton.

A vezérlő méretéhez és helyzetéhez kapcsolódó tulajdonságok=> A Tcontrol méretre és helyzetre vonatkozó tulajdonságait az összes vezérlőnél megtalálhatjuk. A vezérlők helyzetét a Left és a Top tulajdonságok értéke határozza meg. Méretét pedig a Height és a Width. (A komponensek helyzetének megadásakor ügyelni kell arra, hogy mindig a szülőkomponens (azaz a Parent tulajdonságban megadott komponens) területéhez kell viszonyítanunk.

Engedélyezés és láthatóság=> a felhasználó két alaptulajdonság segítségével engedélyezheti, ill. rejtheti el a komponenseket. Ha a komponens nem engedélyezett (Enabled=false), akkor erről a felhasználó valamilyen 'képi tájékoztatást' kap, pld. Futáskos szürke (inaktív).

Betűtípusok=> a komponensek felhasználói felületének kialakításakor gyakran használjuk a Color és Font tulajdonságokat. A Color tulajdonság általában a komponens háttérszínére vonatkozik, de a betűtípusok és más grafikus elemek is rendelkeznek Color tulajdonsággal. Sok komponensnél megtalálhatjuk a ParentColor és ParentFont tulajdonságot, ami azt jelzi, hogy örökli e a komponens a szülő színeit, betűtípusait.

TwinControl osztály (VCL)=> a felhasználói felületet felépítő elemek többsége ablak. A felhasználó nézőpontjából az ablak olyan terület a képernyőn, amelyet keret vesz körül, van neve, és általában egy rendszermenüt is tartalmaz.
Technikai oldalról azonban az ablak egy bejeghyzés a belső rendszertáblában, amely általában egy, a képernyőn látható elemhez kapcsolódik, és kód is tartozik hozzá.
Az összes ablakra igaz, hogy a Windows tud róluk, és minden alkalommal, amikor valami történik a rendszerben, a megfelelő ablak üzenetet kap, és ennek hatására valamilyen kódot futtat. Minden ablakhoz tartozik egy függvény (*ablakbejárás*, window procedure), ami az ablak számára érdekes üzenetekez kezeli.

Komponenspaletta: a teljesség igénye nélkül néhány:
1. szövegbeviteli komponensek:

a, edit komp.: egy sornyi szöveg beírása,

b, memo komp.: több sor bevitelére alkalmas, de csak egy betűtípust használhatunk. Szerkeszthetjük soronként (a Lines segítségével) vagy akár egészben is (Text tulajdonság használatával)