Vizuális form öröklés. Nyomtatási lehetőségek.

A, Vizuális form öröklés: ha több hasonló formot kell készítenünk, netalán különböző eseménykezelőkkel, akkor ahelyett, hogy a TForm alapján hoznánk létre egy formot, származtathatjuk azt egyszerűen egy másik formból is, majd kiegészíthetjük új komponensekkel, vagy módosíthatjuk a meglévő komponensek tulajdonságait.
De mi az igazi előnye?
Ez attól függ, milyen alkalmazást fejlesztünk. Ha a program számos formot tartalmaz, és ezek között sok hasonló van, illetve sok használja ugyanazokat a komponenseket, akkor létrehozhatunk egy alapformot a közös komponensekkel, elkészíthetjük a megfelelő eseménykezelő eljárásokat és a többi formot erre 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 a formjainkat a megfelelő programhoz tartozó szülőformból származtatjuk. A vizuális formöröklés legjelentősebb előnye, hogy később bármikor lecserélhetjük az eredeti formot és vele együtt természetesen a származtatott formokat is.

```
type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

→

```
type
  TForm2 = class(TForm1)
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

A vizuális formöröklés szabályai:

1. az alosztálybeli form rendelkezik a szülőform összes komponensével, és ezek mellett új komponenseket is tartalmazhat,
2. az alaposztály komponenseit nem tudjuk eltávolítani, de – mivel vizuális elemekről van szó – láthatatlanná tehetjük azokat,
3. ha megváltoztattuk egy örökölt komponens valamely tulajdonságát, és később ugyanezt a tulajdonságot módosítjuk az eredeti komponensnél, a változás az örökölt komponensben nem jelenik meg. Ha egy másik tulajdonságot módosítunk, a vltoztatás az örökölt tulajdonságokban is tükröződik majd. A két tulajdonságértéket az Object Inspector helyi menüjének Revert to Inherited pontjával egyeztethetjük össze.
4. A komponensek mellett az új form örökli az alapform összes tagfüggvényét, beleértve az eseménykezelőket is.

B, Nyomtatási lehetőségek: A Delphi grafikus objektumaira alapozott nyomtatás egyszerű feladat.. Például a form képének az aktuális nyomtatóra való küldéséhez elegendő meghívnunk a Print metódust.
Ha programozni szeretnénk a nyomtatót, akkor abban a modulban, ahonnan a nyomtatást végezzük, használnunk kell a Printers modult: **uses Printers;**
A Printers modulban definiált Printer függvény egy TPrinter típusú objektummal tér vissza, amely a nyomtatókkal való kapcsolattartást szolgálja.
A TPrinter osztály (TStrings típusú) Printers tulajdonsága tartalmazza az összes – a rendszerben elérhető – nyomtatót, a PrinterIndex pedig meghatározza, hogy ezek közül melyik az alapértelmezett. Ha nincs alapértelmezett nyomtató, akkor a PrinterIndex értéke –1.

A nyomtatási feladat megnyitását a TPrinter BeginDoc metódusával végezzük. Ezt követően a TPrinter Canvas objektumnak felhasználásával végzett rajzolási műveletek eredménye bekerül a nyomtatási feladatba. Ha több oldalas a dokumentum, akkor bármikor új oldal nyithatunk a NewPage metódussal. A nyomtatást az EndDoc metódussal kell befejeznünk, ami lezárja a feladatot és elküldi azt a nyomtatókezelőnek. Abort metódus megszakítja a nyomtatási folyamatot.
A TPrinter.Canvas objektumot ugyanúgy kezelhetjük, mint a képernyőn megjelnő grafikus objektumokat.
A nyomtatás előtt az aktuális papírméretekről nyerhetünk információt a PageHeight és a PageWidth tulajdonságok használatával.
A Copies egész típusú tulajdonsággal beállíthatjuk a másolatok számát, a (TprinterOrientation típusú) Orientation tulajdonság értéke (poPortrait, poLandscape) határozza meg a papír függőleges, ill. vízszintes állását.
Szöveges állományok nyomtatóra küldését az AssignPrn eljárás segíti, melynek egyetlen paramétere egy TextFile típusú fájlváltozó.
Szöveg nyomtatása

```
var
        MyFile: TextFile;
begin
        AssignPrn(MyFile);
        Rewrite(MyFile);
        Writeln(MyFile, 'Print this text');
        System.CloseFile(MyFile);
End;
```

Jelentések: az adatbázisokban tárolt adatokból jelentéseket készíthetünk, amelyeket megjeleníthetünk, illetve ki is n yomtathatunk. A QReport palettalapon található komponensek (QuickRep, QRLabel, QRDBText, stb) felhasználásával.
Elsősorban azadatbázhisban tárolt adatokat megjelenítő QRDBxxx nevű vezérlőkre van szükségünk, amelyek DataSet és DataField tulajdonságait megfelelő értékekre kell beállítani. A statikusan megadható feliratok, képek stb. megjelenítésére a QRxxx nevű vezérlőelemeket használhatjuk.
Jelentés elkészítésének menete:
A jelentés elkészítését a QuickRep vezérlőelem formon való elhelyezésével kezdjük. (Fontosabb jellemzőit szerkesztési időben a vezérlőelem Page tulajdonságán állíthatjuk be – méretek, írásirány, margók, oszlopok száma, egymástól való távolságuk, a vonalzó{Ruler}megjelenítése). Ahhoz, hogy a QuickRep vezérlőelem a beállítottaknak megfelelő méreteket vegye fel, a vezérlő felbukkanó menüjében a Zoom out menüpontot kell kiválasztani.
További teendők:

1. a QuickRep vezérlőelem Bands tulajdonságán keresztül kell bekapcsolni annak az ismétlődő sávnak a megjelenítését, melynek tartalma a kinyomtatott jelentés mindegyik oldalán fejlécként (HasPageHeader) vagy lábjegyzetként (HasPageFooter), az elején (HasTitle), vagy pedig az ismétlődő struktúrájú feljegyzések (HasDetail) előtt (HasColunHeader), illetve után (HasSummary) jelenik meg.
2. A sávokon el kell helyeznünk azokat a vezérlőket (a QReport palettalapról) amelyek az adatok megjelenítését végzik (pld: QRLabel segítségével szöveget, a QRDBText segítségével pedig egy adatmező értékét jeleníthetjük meg)
3. teszteléshez, illetve a kész jelentés kinyomtatásához a QuickRep vezérlő felbukkanó menüjének Preview menüpontját kell kiválasztani. Ennek segítségével megjeleníthetjük a „Print Preview" ablakot. Az ablak eszközsorában található

gombok az oldalak különböző mértékű megtekintését, az oldalak közötti léptetést, az oldalak nyomtatóra való küldését segítik elő.

Kiegészítő komponensek
Free Report
XReport
Christal Report
ReportBuilder