Public. A BDE

A, Public: az osztályok bármennyi adatot és tagfüggvényt tartalmazhatnak. Az adatokat el kell rejtenünk, be kell tokoznunk az azokat használó osztály belsejébe. A betokozás nagy előnye, hogy az osztály létrehozója bármikor módosíthatja az osztály belső ábrázolását.
A betokozás lényege egyszerű: úgy kell elképzelni egy osztályt, mint egy 'fekete dobozt', amelynek csak kis része látható. A doboz belsejét nem látjuk, kizárólag a rendelkezésünkre álló felületen, az interfészen, keresztül férhetünk hozzá. Az objektumok használata közben a kód nagyobb része rejtve marad. Általában nem tudjuk, milyen belső adatokat tárol az objektum, és nem is férhetünk hozzá közvetlenül. Az adatokat csak tagfüggvényeken keresztül érhetjük el, amelyek megvédik az osztályt az illetéktelen hozzáféréstől. Ezt az adatok elrejtésének (data hidding, information hidding) nevezzük. A Delphi osztály alapú betokozást valósít meg. Az egységek felületíró (interface) részében meghatározott azonosítók a program többi egységéből is elérhetők, feltéve, hogy a 'uses' utasításban felsoroljuk az azonosítót tartalmazó egység nevét.

Hozzáférési szintek:

1. *private:* utasítással megjelölt tagfüggvények vagy mezők az osztályt bevezető egységen kívül nem érhetők el.
2. *protected:* az adott mező vagy tagfüggvény korlátozott láthatósággal rendelkezik. Kizárólag az adott osztályban, illetve annak leszármazottaiban férhetünk hozzá. Tehát a mező tartalmához csak az osztály, az alosztályok és az egységen belüli kódrészletek férhetnek hozzá.
3. *public:* a program tetszőleges pontján, szabadon hozzáférhetők.

Az osztályok mezőinek általában privátnak kell lenniük, a tagfüggvények többnyire nyilvánosak.
A hozzáférést szabályozó kulcsszavak kizárólag az adott egységen kívüli kóddal szemben védik az osztály felületleíró részben bevezetett tagjait. Ez azt jelenti, hogy ha ugyanabban az egységben határozunk meg két osztályt, nem tudjuk megvédeni azok privát mezőit.

B, A BDE (Borland Database Engine): a BDE lehetővé teszi a helyi adatbázis-formátumokhoz és az SQL kiszolgálókhoz való hozzáférést egyaránt, valamint minden olyan adatbázishoz is képes hozzáférni, amely ODBC (Open Database Connectivity) illesztőprogramok segítségével érhető el. (A Borland azonban a BDE motort elavultnak tartja, helyette az InterBase Express-t ajánlja megoldásként.

A BDE mint programozási felület sok olyan alapfeladat elvégzése alól mentesít, mint az adattáblák és bejegyzések lezárása, bejegyzések frissítése, alapvető I/O műveletek, stb. A BDE-használat általában a Delphi adatelérési komponensein keresztül történik (az eszközpaletta DataAccess lapja).
A BDE AZ ALKALMAZÁSOK ÉS AZ ADATFORRÁSOK KÖZÖTT ELHELYEZKEDŐ SZOFTVERRÉTEG.
A Delphi a relációs adatbázisok kezelését támogatja.
Logikai (alias) nevek létrehozása a BDE Administrátor segítségével: a BDE Administrator (Start-beállítások-vezérlőpult-BDE Administrátor) segítségével beállíthatjuk, hogy a programunk az elérni kívánt adatokat hol találja meg.
A BDE Administrátor ablakban bal oldalt, a Databases lapon az ún. logikai nevek(alias), a Configuration lapon pedig a meghajtók és a rendszerbeállítások felsorolása látható. A jobb oldali panelben a kijelölt elemre vonatkozó paraméterek láthatók.

A logikai vagy adatbázisnév, ill. az általa takart paraméterkészlet, az adathalmaz elérhetőségét és fizikai elhelyezését írja le. (Az SQL adatbázisokra val hivatkozást csak logikai névvel lehetséges.) Az adatok fizikai áthelyezése is csak a logikai név beállításainak, programon kívüli megváltoztatását igényli.

DataSet komponens: számos olyan rekordhoz enged hozzáférést, amelyet valamilyen adatforrásból olvas ki a rendszer, majd ezeket eltárolja egy ideiglenes tároló helyre. Ezeket a felhasználó módosíthatja, majd a módosításokat elmentheti az állandó tárolóba.

A logikai nevet a formra helyezett DataSet komponens DatabaseName tulajdonsága kapja értékül:

```
Query1.DatabaseName :='Adatbazis_pelda';  //logikai név
Query1.DatabaseName :='C:\AdatBazis\pelda3'; //könyvtárnév
```

Az adatforrásunkhoz új logikai nevet többféleképpen is létrehozhatunk·

1. használhatjuk a BDE Administrator-Object-New,
2. vagy a felbukkanó Databases panel New menüpontja,

Ugyanitt módosíthatjuk,átnevezhetjük vagy törölhetjük is azokat.
A logikai neveket az SQL Explorer (Database-Sql Explorer menüpont) alkalmazás segítségével is létre lehet hozni.
A Delphi vezérlőelemei adatbázisok használatához: az egyszerűbb adatbázis-kezelési feladatok a komponenspaletta DataAccess (nem vizuális komponensek) és a DataControls (vizuális komponensek) lapján található vezérlőelemek felhasználásával oldható meg.

1. *DataAccess*=> az itt található vezérlőelemek (Sesseion Database, Table, Query, stb.) feladata az adatbázisban tárolt adatok és az alkalmazás közötti kapcsolat megteremtése.
   A Sesseion és a Database vezérlőelemeket nem kell külön elhelyezni a formon, mivel ez önműködően létrejön.
   A DataAccess lapon lévő Table, Query, StoredProc komponenseket adatkészletező vezérlőelemeknek nevezzük, mivel az adatbázishoz intézett lekérdezések eredményeként az adatrekordok csoportját kapják vissza.
   A visszakapott rekordokat a DataSource vezérlőelemen keresztül átadava a DataControls palettalap vizuális komponenseinek segítségével jelenítjük meg.
2. *DataSource*=>vezérlőelem, akkor kapja meg az adatelérési jogot, amikor a DataSet tulajdonságában megadjuk egy adatkészletező vezérlőelem nevét, például a formon elhelyezett Table komponensét. Az adatkészletező vezérlőelemek 'hangolása' pedig abból áll, hogy a DatabaseName tulajdonság értékét beállítjuk az alkalmazásban használt adatbázis nevére (logikai név vagy elérési út). A TableName tulajdonságon keresztül megadhatjuk az adatbázis elérni kívánt táblájának nevét.
3. Az adatkészletező vezérlőelemek két további objektumon keresztül jutnak hozzá az adatbázisban tárolt adatokhoz. Az egyikük a *Tdatabase*, amely lehetővé teszi az adatbázis-kapcsolat vezérelhetőségét, a másik a *Tsession* típusú objektum, amely egyszerre több adatbázis-kapcsolat menedzselésére képes.
4. *Adatkészletező (DataSet) vezérlőelemek*: a BDE adatkészletező vezérlőelemei a komponenspaletta DataAccess lapján találhatók (Table, Query, StoredProc). Ezeknek az adatkészletező objektumoknak közös őse a TdataSet osztály. A TdataSet-ből származtatott osztályok fontosabb tulajdonságainak és metódusainak használata:

5. Session globális objektum AddStandardAlias vagy AddAlias metódusai
   Kell: SaveConfigFile hívás a rögzítéshez

6. *Az adathalmaz állapota:* amikor adathalmazzal dolgozunk, különböző állapotokat használhatunk. Az állapotokat a State tulajdonság jelzi, amely több értéket vehet fel:
   A, dsBrowse: jelzi, hogy az adathalmaz normál böngészési üzemmódban van, az adatok megtekintésére és a rekordok pásztázására használható,
   B, dsEdit: jelzi, hogy az adathalmaz szerkesztési üzemmódban van, melybe akkor kerül, amikor a program az Edit tagfüggvényt hívja meg. A megváltozott rekord adatbázisba küldése után az adathalmaz kilép a dsEdit állapotból.
   C, dsInsert: azt jelzi, hogy új rekordot veszünk fel az adathalmazba. Ezt úgy tehetjük meg, hogy az Insert vagy az Append tagfüggvényt hívjuk meg.
   D, dsInactive: azt jelzi, hogy zárva van az adathalmaz.
   E, dsSetKey: ekkor keresést készítünk elő.
   F, dsCalcField: azt jelzi, hogy mezőszámítás van folyamatban. (amikor az OnCalcFields eseménykezelőt hívtuk meg.
   G, dsNewValue, dsOldValue, dsCurValue: azt jelzik, hogy a gyorstár frissítése zajlik,
   H, dsFilter: azt jelzi, hogy az adathalmazra szűrő van beállítva – amikor az OnFilterRecord eseménykezelőt hívtuk meg.

7. *Adatfelismerő vezérlők használata:* miután beállítottuk a megfelelő adathozzáférési komponenseket, létrehozhatunk egy felhasználói felületet, hogy lehetővé tegyük a felhasználók számára az adatok megtekintését, módosítását. A komponenseknek a megfelelő tulajdonság, a DataSource segítségével kapcsolódnak egy adatforráshoz. Sok olyan komponens van, amely a szokásos vezérlőkre emlékeztet, de felismeri az adatokat:
   A, dbEdit: hasonlít az Edit komponenshez, egy érték megváltoztatására alkalmas,
   B, dbCheckBox: szintén megfelel a CheckBox komponensnek.
   C, dbGrid: adatrács, egy olyan táblázat, amely képes egyszerre megjeleníteni egy teljes adattáblát, lehetőséget ad a görgetésre és a mozgásra, valamint módosítható a tábla tartalma.
   D, dbNavigator: olyan gombok gyűjteménye, amelyeket az adatbázisban való mozgásra és különféle adatbázis-műveletek végrehajtására használhatunk.

8. *Szöveg alapú adatfelismerő vezérlők:*
   A, dbText: olyan mező tartalmát jeleníti meg, amit a felhasználó nem módosíthat.
   B, dbMemo: azt teszi lehetővé, hogy a felhasználó megtekinthessen és módosíthasson egy nagy szövegmezőt, amely lehet egy jegyzet vagy BLOB (Bnari Large Object, nagy bináris objektum).

9. *Grafikus adatfelismerő vezérlők:* a Delphi egy grafikus adatfelismerő vezérlőt is tartalmaz:
   dbImage: ez az Image komponens kiterjesztése, és egy BLOB mezőben tárolt jelenít meg.

10. *Lista alapú vezérlőelemek:* akkor használjuk, ha azt szeretnénk engedélyezni a felhasználók számára, hogy egy előre meghatározott listáról választhassanak értéket.

A, dbListBox: előre meghatározott elemek közüli választás (zárt kijelölés) telsz lehetővé, de nincs szövegbevitel,

B, dbComboBox: használható zárt kijelöléshez és felhasználó általi adatbevitelhez is Lehetővé teszi, hogy a felhasználó új értéket is megadhasson, de választhasson a listán szereplők közül.

C, dbRadioGroup: választógombokat jelenít meg, ezek közül csak egyet lehet kiválasztani. Kizárólag zárt kijelölést tesz lehetővé és csak korlátozott számú lehetőség közüli választásra ad módot.

D, dbCheckBox: egy beállítás ki- és bekapcsolására használható, a logikai mező értékétől függően.

11. *Számított mező felvétele:* az adathalmazban egyszerre csak egy rekord lehet aktív. Ezt a rekordot egy átmeneti tárban (pufferben) tárolja a program. Az adatfelismerő vezérlők közvetlenül kapcsolódnak ezekhez a mezőobjektumokhoz.

A Delphi futásidőben, automatikusan hozza létre a TField komponenseket, minden alkalommal, amikor a program megnyit egy adathalmaz-komponenst. Ez azután történik meg, hogy a program elolvasta az azzal a táblával vagy lekérdezéssel kapcsolatos metaadatokat, amelyre az adathalmaz hivatkozik. A mezőkomponensek az adathalmaz Fields tömbtulajdonságában tárolódnak. Ezeket az értékeket szám szerint vagy a nevük alapján lehet elérni.

Mivel néhány mezőkomponenst tervezési időben hozunk létre a mezőszerkesztővel, nyilvánvaló, hogy azokhoz a mezőkhöz, amelyeket átugrunk(???), nem tartozik majd objektum. Az átugrott mezők futásidőben sem lesznek elérhetők. Amikor egy program futásidőben megnyitja valamelyik táblát és nincsenek tervezési időben kialakított mezőkomponensek, a Delphi a tábla meghatározása alapján hoz létre mezőobjektumokat, de ha van néhány tervezési időben létrehozott mező, a Delphi azokat használja és nem vesz fel további mezőobjektumokat.

Az új mező tartalma kiszámításának módját meg kell adnunk. Ezt a számítást a ClientDataSet komponens OnCalcFields eseményével hajthatjuk végre. A számított mező tartalmát minden rekord esetében kiszámolja és minden alkalommal újraszámolja azt, amikor a rekordot betöltjük.

•