

A DELPHI PROGRAMRENDSZER ALKALMAZÁSÁNAK LEHETŐSÉGEI A REPÜLŐESZKÖZÖK ÜZEMELTETÉSE SORÁN ALKALMAZOTT ANYAGOK RAKTÁROZÁSÁBAN ÉS TÁROLÁSÁBAN

**Kovács Klára egyetemi szakoktató
Zrínyi Miklós Nemzetvédelmi Egyetem
Vezetés- és Szervezéstudományi Kar
Repülő sárkány-hajtómű tanszék**

A repülőcsapatok gazdaságos és hatékony üzemeltetésének meghatározó eleme a repülőanyag-gazdálkodás és biztosítás. A MH központi raktárbázisa megbízható, számítógéppel támogatott nyilvántartási rendszerrel rendelkezik, aminek a korszerű tervezéshez, készlet felhasználáshoz potenciálisan rendelkezésre álló lehetőségei még nincsenek kellően kihasználva. Az alábbiakban a meglévő rendszer épp általam kidolgozás alatt álló fejlesztési változatát kívánom bemutatni.

A RAKTÁROZÁS-TÁROLÁS SZÁMÍTÁSTECHNIKAI PROBLÉMÁI

A repülőgépek biztonságos üzemeltetésének egyik feltétele, hogy bármely meghibásodás esetén a javításhoz szükséges anyagok a rendelkezésre álljanak. A feladat megoldásához jól ellátott raktárak szükségesek. Nem megoldható azonban, hogy minden repülőtéren, minden anyagból maximális készlet álljon rendelkezésre, mivel ez akkora többlet költséget és többlet anyagmennyiséget eredményezne, aminek gazdaságossága megkérdőjelezhető. Itt nem a mindennapokban felhasznált fogyóeszközökről (pl. anyacsavarok) van szó, hanem a repülések műszaki kiszolgáláshoz szükséges anyagokról. A repülő csapatok isaszegi raktárbázisán hosszú évek folyamán létrehoztak egy jól működő raktározási rendszert. A későbbiekben a raktár áttért a számítógéppel segített adatfeldolgozásra, mely adatbázisában egy-egy anyagról többféle gazdaságossági-, műszaki mutatót tartanak nyilván. Van olyan, aminek közel

száz jellemzője található meg. A nyilvántartásban szerepelnek a különböző műszaki adatok - a méretektől kezdve, az esetleges üzemidőig -, de természetesen a beszerzéshez, felhasználáshoz, javításhoz szükséges mutatókat is nyilvántartja a rendszer.

Néhány évvel ezelőtt a Magyar Hcnvédségben bevezették az egységes Eszköz Analitikai Nyilvántartó Rendszert. Minden alakulat áttért a számviteli törvényben meghatározottaknak megfelelő program használatára. Ez azonban a speciális feladatoknak nem felel meg, mivel nem tud nyilvántartani egy anyagról olyan sokféle információt, mint ami az isaszegi raktárbázisban rendelkezésre áll, illetve a folyamatos, gyors logisztikai biztosításhoz szükséges. Ez a probléma nagyon egyszerűen megoldható a Delphi rendszer segítségével. A rendelkezésre álló adatbázisok tartalmazzák az Eszköz rendszerhez szükséges információkat. Nem kell mást csinálni, mint egy lekérdezést készíteni, ami egy olyan adattáblát generál, amit az Eszköz rendszer fel tud dolgozni, mert lehetőség van e rendszeren belül, a lemezzről, illetve file-ból származó adatokat feldolgozásának. A Delphi biztosítja ezt, mert az Eszköz rendszer olyan nyelven készült, ami a dBase adatbázis-kezelőt használja és ezt a Delphi is alkalmazni is képes. Mivel a rendszer felépítése titkos, ezt természetesen az Eszköz rendszert készítő programozók tudják csak végrehajtani, mert ők ismerik a bemeneti táblák felépítését. Fontos lenne biztosítani, hogy az idők folyamán felgyűlt adathalmaz ne vesszen el, mert az itt tárolt információk felhasználhatók a repülőtechnika jobb kiszolgálására. Ezek között különösen fontosak a különböző statisztikai számítások, amelyek az anyagok, eszközök és berendezések beszerzésének tervezését számottevően meggyorsítják.

A DELPHI ALAPJAI

A Delphi programrendszer alkalmazhatóságának megértését megkönnyítheti a programnyelvek fejlődésének vázlatos áttekintése.

Az első nagygépes programozási módszer az úgynevezett "gépi kód" volt, ami tulajdonképpen még nem is igazán mondható a mostani értelemben vett nyelvnek. Ez lehetővé tette, hogy a memóriát közvetlenül programozzuk. Tulajdonképpen két bitsorozatot tartalmazott, ami "megmondta" a gépnek, hogy melyik memóriacím milyen értéket tartalmazzon.

A következő állomás az Assembly nyelv, ami már nem olyan közvetlen, mint elődje, de ez is a memóriacímek átprogramozását tette lehetővé. A legnagyobb különbség, hogy itt már vannak parancsszavak. Az ezen a nyelven

A Delphi programrendszer alkalmazásának lehetőségei a repülőeszközök üzemeltetése során alkalmazott anyagok raktározásában és tárolásában

készült programokhoz fordító programok szükségesek, ami a gép számára érthető bitek sorozatává alakítja át a programsorokat.

Az előző két nyelv hátránya, hogy gépenként különbözők, tehát előfordulhat, hogy ugyanazt a problémát az egyik processzorral másképpen lehet megoldani, mint a másikkal.

A személyi számítógépek elterjedése és egyre szélesebb körű felhasználása során biztosítani kellett, hogy a felhasználók is tudjanak saját, igényeiknek megfelelő programokat készíteni. Biztonsági célok elkerülhetetlenné tették, hogy a memória közvetlen programozásának lehetőségét kivegyék a programkészítő kezéből. Olyan nyelvre volt szükség, ami az emberi gondolkodáshoz közelebb áll, minden gépen futtatható programot készít - a fordító segítségével -, könnyű javítani stb. A különböző munkaterületeken dolgozók között azonban felmerült az igény is, hogy a saját szakmájuknak jobban megfelelő programnyelveket hozzanak létre. Ilyen például a Fortran, Algol, C, Pascal, Clipper stb. A programok írásakor az egyik legfontosabb feladat a megfelelő nyelv kiválasztása. Ez nem is olyan egyszerű feladat, mert mint a különböző nyelvek más-más jellegű feladatok megoldására készültek. Úgy kell ezt érteni, hogy az egyik az adatbázisok kezelésében erős, a másikkal könnyebb megoldani a gazdaságossági problémákat, míg a harmadik inkább a mérnöki gyakorlatban előforduló feladatok megoldására alkalmasabb. Ezek közül a Pascal és a C terjedt el a legjobban, mert könnyen kezelhetők, jól használhatók a műszaki és gazdasági életben egyaránt, ugyanakkor "szép" programok készíthetők velük.

A Delphi programrendszer az objektumorientált Pascal nyelven alapul. A Pascal olyan programozási nyelv, ami a legáltalánosabban elterjedt, objektum orientáltsága pedig lehetővé teszi, hogy hasonló adatszerkezetek, illetve feladatok készítésekor ne kelljen mindent újra programozni, hanem felhasználhatók legyenek az előzőekben elkészített szerkezetek úgy, hogy "örököltetjük" a szerkezetüket és a bennük definiált eljárásokat, függvényeket. Azokat az objektumokat, amikből örököltetünk szülő objektumnak hívjuk, míg az öröklő objektumok a leszármazott, vagy gyerek objektumok. Az az előnyük is megvan, hogy a szülő objektumok eljárásai, függvényei átdefiniálhatók a követelményeinknek megfelelően. Mindez tulajdonképpen érvényes a C nyelvre is, bár annak elemei, szerkezete kicsit különbözik a Pascal-tól.

Mi is az a Delphi? A gyártó hivatalos megfogalmazása szerint:

"A Delphi olyan Windows-os alkalmazásfejlesztő eszköz, amelyet az alábbi fontosabb tulajdonságok kombinációja jellemez:

- A vizuális, komponensalapú felépítéséből következő gyors alkalmazásfejlesztés (RAD-Rapid Application Development).

- Az optimalizálásra is képes forrásnyelvi fordítóprogram.

- A testre szabható adatbázis-hozzáférés." [1]

Az objektumorientált programozás alapelemeit, az egységbefoglalást (encapsulation), az öröklődést (inheritance) és a polimorfizmust (polymorphism) az Turbo Pascal-hoz hasonlóan tartalmazza. Tovább lépést jelent, hogy lehetőségünk van az objektumok hozzáférési jogainak szélesebb körű meghatározására, melyek már a program tervezésekor beállíthatók. Ezek a következők:

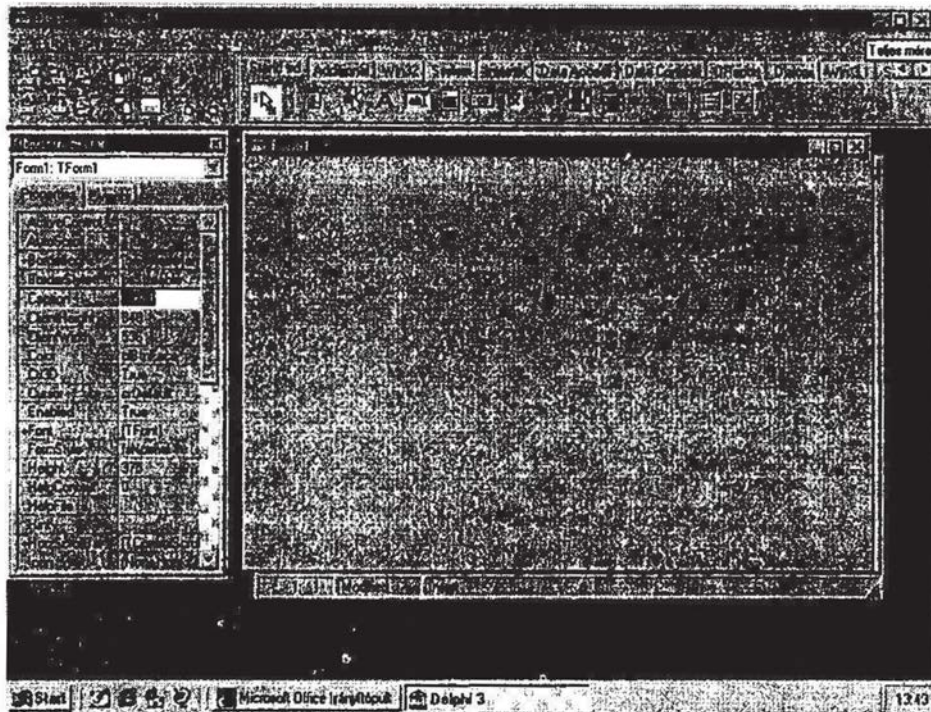
- nyilvános (public);
- saját (private);
- védett (protected);
- publikált (published).

Azon események kezelését is lehetővé teszi, amelyek eddig oly sok fejtörést okoztak programkészítőnek és felhasználónak egyaránt, például a különböző hibák felmerülése, amik után a program egyszerűen "lemerevedett", a nem mentett adatok elvesztek, és még sorolhatnám a különböző problémákat. Most lehetőségünk van kikényszeríteni a "rendes befejezést" pl. bezárhatjuk a nyitott adatbázisokat, a menthető adatokat menthetjük akár más néven is stb. Támogatja az úgynevezett nyitott tömbök létrehozását, ami azt jelenti, hogy a megszokottól eltérően nem kell megadni a méretét, hanem a program futása során derül ki, hogy tulajdonképpen mekkora is lesz.

Szólnom kell még egy nagyon fontos dolgról. A gyártó megfogalmazásában szó van a vizuális és komponens alapú alkalmazásfejlesztésről. A 1. ábrán látható, hogy a Delphi 3.0 verziója hogyan jelentkezik be.

Megfigyelhető, hogy az ablak öt részre van osztva. Az első rész a menüsor, ami tulajdonképpen egy szabvány Windows menü. Alatta balra található az eszközzaletta, ami szintén a szabványos elemeket tartalmazza, a megszokottól kicsit eltérő képekkel. Az eszközzaletta mellett a komponenspaletta helyezkedik el, ami a Windows-ban megszokott elemeket tartalmazza megszokott jelekkel, funkcióiknak megfelelő csoportosításban. A legalsó sorban pedig az Object Inspector és egy üres Form kapott helyet. A Form egy Windows alkalmazás ablaka. Ide helyezzük el a komponenspalettáról a kiválasztott elemeket. A rajta lévő rácpontokhoz igazíthatjuk őket - ott fognak megjelenni futtatáskor -, de a helyük számmal is megadható az Object Inspector Properties (tulajdonságok) oldalán, a megfelelő értékek beírásával. Ez kicsit nehézkes, mert pixelben kell dolgozni. Vannak látható és nem látható komponensek. Ez a különbség onnan adódik, hogy egy menü komponens felhelyezése csak a tervezési időben látható, futtatáskor már természetesen csak a beleírt menüpontok. A látható komponensek viszont már tervezéskor is a helyükön vannak.

A Delphi programrendszer alkalmazásának lehetőségei a repülőeszközök üzemeltetése során alkalmazott anyagok raktározásában és tárolásában



1. ábra

Nagy segítség az Object Inspector. Az itt megjelenő tulajdonságok tervezési időben beállíthatók, és ezek egy része futtatáskor, másik része már tervezéskor hatályba lép. Az Events (események) oldalon kezdődik el a tulajdonképpeni programozás. Itt kell megadni, hogy pl. egy gombra való kattintáskor mi történjen. Megnyitja a programszerkesztőt és megírhatjuk a megfelelő programrészletet.

Nagy segítség a programozásban a Delphi rendszer, mert leveszi a programozó válláról az egyik legnagyobb gondot, hogy a program szép is legyen ne csak hasznos. Nem kell hosszú napokat eltölteni a szép programok kialakításával és azzal foglalkozni, hogy átszámolgassuk a koordinátákat, ha egy másik gépre kerül át a program. A jól bevált kattintásos módszerrel egyszerűen felhelyezhetők az elemek a Form-ra. Húzással áttehetők másik helyre, ha nem tetszik, kiválaszthatók a színek, megadható a megfelelő betűtípus, az egér formája, lehetőség van a standard Windows-os értékek megadására - ami egyforma minden gépen - és még sorolhatnám a programozást kiváltó lehetőségeket.

A DELPHI ADATBÁZIS-KEZELÉSE

A Delphi nem kimondottan adatbázis-kezelő nyelv, de igen jól felszerelt komponensekkel rendelkezik a felmerülő problémák megoldásához. "Az adatbázis-kezelő rendszerek elsődleges célja az információ-szolgáltatás, amihez négy dologra van szükség:

- adatok;
- hardver;
- szoftverek (adatbázis-kezelő rendszer);
- felhasználók.

E négy alkotóelem összességét Információs rendszernek nevezzük." [2] Természetesen az információ és az adat nem azonos. A számítógépen tárolt adatokból a felhasználó számára fontos információk nyerhetők ki. Talán nem haszontalan tisztázni mi is az az adatbázis.

"Az adatbázis tartalmaz:

- adatokat;
- az adatok közötti összefüggéseket." [3]

Az adatbázis-kezelőkkel szemben támasztott követelmények közül az alábbiakat fontos kiemelni:

- adatfüggetlenség: az információkérés legyen független az adatok tárolási módjától;
- adatok közötti kapcsolatok ábrázolása: ha ez nincs meg, akkor egy használhatatlan adathalmazunk van;
- adatvédelem: illetéktelenektől való védelem, valamint a felhasználók ne okozhassanak véletlenül hibákat;
- adatsérülések javíthatósága: az üzemzavarokból, vagy géphibákból adódó sérülések kezelése;
- adatok megosztása: többfelhasználós adatbázisoknál szükséges.

Az adatbázis-kezelők felépítésüket tekintve igen különbözők, de egyes alapvető funkcióikban egyformák. Mindegyiknek van:

- adatleíró nyelve: az adatbázis formai leírására szolgáló elem;
- adatkezelő nyelve: ami az adatok manipulálását teszi lehetővé;
- képernyő tervezője: a képernyőn történő műveletek megjelenésére;
- jelentés készítője: a tárolt adatokból kimutatások, jelentések elkészítésére.

A Delphi rendszerben sikerült megoldani a különböző adatbázis-kezelőkkel már elkészített adathalmazok egyszerű felhasználását, ezáltal nincs szükség a

A Delphi programrendszer alkalmazásának lehetőségei a repülőeszközök üzemeltetése során alkalmazott anyagok raktározásában és tárolásában

meglévő adatok újbóli gépre vitelére. Ezek lehetnek a saját gépünkön elhelyezkedő lokális-, illetve távoli, hálózati adatbázisokban. A rendszer lehetővé teszi, hogy saját magunk is készítsünk adattáblát, amit úgy állíthatunk elő, hogy a már meglévő adatbázisunkhoz alkalmazkodjon, vagy átkonvertálhatjuk egy másik fajta adatbázis-kezelőbe.

A Delphi komponenspalettáján két oldalon helyezkednek el azok az elemek, amelyek segítenek megvalósítani az adatbázis-kezelést. A Data Access oldalon a nem vizuális komponensek érhetők el. Ezek segítségével a különböző adatbázisok helyei, egymáshoz való kapcsolódási módjai adhatók meg. A Data Controls oldalon pedig a felhasználói felület elkészítéséhez szükséges komponenseket helyezték el. Az elemek segítségével bonyolult, mégis könnyen kezelhető programok készíthetők.

A Delphi alkalmazásának van még egy további előnye is. Minden adatbázis-kezelőnek van egy adatkezelő nyelve. Az egyik az úgynevezett SQL. Ennek az a hátránya, hogy a felhasználónak olyan ismeretekkel kell rendelkeznie egy adatbázisról, ami nem várható el tőle. A Delphi segítséget nyújt abban, hogy a felhasználó ezt használja holott nem is tud róla semmit. Könnyen megoldható a Data Access és Data Control komponensek segítségével, hogy saját kereső kérdéseket fogalmazzunk meg, és ezek eredményét szabadon konvertálhassuk át egy külön adattáblába. Ez a rendszer - holott nem kimondottan adatbázis kezelésre készült - minden lehetőséget megad, amire szükségünk van.

Szólnom még egy nagyon fontos dologról. A különböző kimutatások, jelentések készítéséről. Ezek jelentősége mindenki előtt ismert, hiszen ezek alapján mondható meg, hogy mi, mennyi, hol és milyen mennyiségi és/vagy minőségi mutatók szerint található meg a raktárban. Vannak olyan jelentések, amik előírt időközönként szükségesek. Ilyenek például a leltározások után készítenedők, vagy éppen a következő évre tervezett és a felhasználásból adódó beszerzések tervezéséhez szükségesek. Ezen kívül mindig előfordulnak olyan esetek, hogy valamilyen kimutatást kell készíteni. A Delphi-nek erre egy egyszerű komponense van, az úgynevezett Report. A Report felhasználásával az ismétlődő jelentések, kimutatások rendszere egyszerűen feldolgozható, mert egy üres nyomtatvány elkészítése után csak a megfelelő gombra kattintva a gép önmaga elkészíti a megfelelő formai követelményekkel és tartalommal ellátott jelentést, ami nyomtatás és aláírás után továbbítható. Az előzőekben említett saját keresőkérdés megfogalmazása után már bonyolultabb a dolog, mert komolyabb programozásra van szükség, a nyomtatott jelentés elkészítéséhez.

KOVÁCS KLÁRA

VÉGSŐ KÖVETKEZTETÉS

Eddigi tapasztalataim azt mutatják, hogy a Delphi programrendszerrel készített programok felhasználóbarátok, könnyen elkészíthetők és módosíthatók. A vele alkotott raktári nyilvántartó rendszer könnyen konvertálható adatbázist eredményez és az Eszköz rendszer fel tudja használni. Bevezetésének fontosságát még az is indokolja, hogy a számítógépeken manapság szinte kizárólag a Windows '95-ös, vagy magasabb verziójú "operációs rendszer" fut. Így egységes felületű, gyors, más Windows-os programokhoz konvertálható, megszokott kezelésű és szélesebb körben felhasználható programokat tudnánk készíteni, felhasználni.

IRODALOMJEGYZÉK

[1] Juhász Mihály-Kiss Zoltán-Kuzmina Jekatyerina-Sölétormos Károly-Dr. Tamás Péter-Tóth Bertalan: DELPHI ÚT A JÖVŐBE, ComputerBooks Budapest, 1998. 3. oldal

[2] Ua. 242. oldal

[3] Ua 242. oldal

Organizaton and maintenance of a reliable supply system is a significant requirement for economical and effective management of the Air Force units. The store centre of the Hungarian Armed Forces has got an up-to-date, computer-controlled registration system; however, its potencialities have not been fully exploited yet in the field of planning and optimum utilization of stocks. In what follows, I wish to demonstrate a version of improvement of the present system, which has been being elaborated by me recently.