INFORMATIKA SZAK

A „REKURZIÓS" BEADANDÓ FELADAT ÉRTÉKELÉSI SZEMPONTJAI

1. A BEADANDÓ FELADAT CÉLJAI

A beadandó feladat célja annak bizonyítása, hogy a hallgató képes önállóan azokat a módszereket és formalizmust alkalmazni, amelyeket a típusok definiálására és megvalósítására el kellett sajátítania, és azokat a számítógépi eszközöket a feladat megoldása során hatékonyan fölhasználnia, amelyekkel megfelelő minőségű munkát képes leadni.

2. BEADANDÓ

1. A komplett anyagot **e-mail-ben kell elküldeni egy fál**lá összecsomagolva. Az fájl természetesen a következő pontban részletezett struktúrában tartalmazza az anyagot, amelynek **neve**: **feladatsorszám + szerző neve** (Pl. 01EinsteinA.zip).
2. A beadandó részei: a **dokumentáció**(ka)t tartalmazó fájl, a **Lazarus/Delphi forráskód**ot tartalmazó fájlok (a beépített saját unit-, include-oké is), valamint a lefordított program (exe-file), továbbá a fordításhoz és futtatáshoz esetleg szükséges további fájlok (pl. tesztadat-fájlok). A beadandót alkotó **fájlszerkezet**:

```
🖫\          -- a gyökérben a futtatási környezet (EXE + adatfájlok)
  🗀\FORRAS\-- PAS… programállomány + UNIT/INCLUDE forrásállományok
  🗀\DOKU\  -- DOC-állományok
```

3. A **feladat sorszáma**, **szövege** és a **név** a dokumentáció nyitó lapján jól látható helyen szerepeljen! (L. a mintadokumentációt doc-ban, pdf-ben!)
4. A **dokumentáció** kinyomtatott formában. Ez elmaradhat. (Ekkor azonban nem tudunk részletesebb értékelést adni róla).

3. SZEMPONTOK

3.1. Módszerek

A '**Felülről-lefelé programkifejtés**' elvének, valamint a **programozási tételek** fölhasználásának tükröződnie kell az algoritmuson. (Az alkalmazott tételek nevei szerepeljenek megjegyzésként a megfelelő algoritmikus részletek mellett.) E beadandók legfontosabb jellemzője, hogy az **algoritmusok rekurzívak. Azaz tilos ismétlődést ciklusok bármelyikével szervezni!** Az egyes felhasznált saját típusok megvalósításnál törekedni kell annak „újrafelhasz-

nálhatóságára", azaz a **megvalósító kódot különálló fájl**ban kell elhelyezni. (Egyszerűen szólva: ahány új típust kell definiálni a programhoz, annyi fájlban kell elhelyezni a hozzájuk tartozó kódot.) *Célszerű a „manuális" kezelőfelületet (klaviatúra-ablak kettőst) és a fájlt speciális típusként felfogni, s ekként a hozzájuk tartozó operációkat rekurzívan megvalósítani.*

3.2. Formalizmusok

Specifikáció rekurzív. Az esetleg bevezetett új típusok algebrai leírását nem kell mellékelni, de a **moduljai**t igen. A modulokban el lehet tekinteni az egyes operációk elő- és utófeltételeinek részletezésétől.

Az **algoritmus**t (és a típust definiáló **modulok**at) az órákon is használt pszeudó kódos nyelvezetben kell leírni. A kód **Lazarus/Delphi** géptermekben található változatában készüljön.

A **fejlesztői dokumentáció** minden megadott „kérdésre" (pl. kód is!) válaszoljon. A **felhasználói dokumentáció** igényesen (futásokból vett screen-shot-okkal illusztrálva) bemutatja a programhasználat miként jét.

3.3. Eszközök

Lazarus/Delphi 4GL fejlesztőrendszere a program gépreviteléhez (egy fordítási egységből álló program szerkesztése, fordítása és futtatás, elemi hibakeresési szolgáltatások készsége).

Valamilyen igényes számítógépes **szövegszerkesztő** a dokumentáció elkészítéséhez. (Tükröződjön a dokumentáción: szövegszerkesztési rutin; minimális ergonómiai igényesség: címek, fejezetcímek, bekezdések, igazítások alkalmazása).

4. SÚLYOK

Legnagyobb súlyt az **algoritmus helyessége** mellett az **algoritmusok rekurzív** volta kapja. Ezt követi a program minősége: a barátságosság (a felhasználói felület, az eredménymegjelenítés ízléses volta) és a biztonságosság (a bemenő paraméterekkel –pl. a fájlnévvel– szemben támasztott elvárásoknak ellenőrzése).

Szintaktikusan hibás programot érdemben **nem értékelünk**. Szemantikusan hibás (elszálló), nem a megadott adatszerkezet(ek)re épülő, vagy lényegesen leegyszerűsített feladatot megoldó program töredékértékben számítható csak be. (Utóbbi két esetben teljes pontértékkel csak a dokumentáció számítható, minden más csak bizonyos százalékban.)

5. PONTOZÁS

Programkód *futtatással ellenőrizve* **Pr**	Fut, *helyes* eredményeket produkál (használat *kényelmessége*, *egyér-telműsége*,...)	0..30
	Ha nincs legalább 3 –lényegesen eltérő– adat-fájl	-10
	Ha egyáltalán nincs adat-fájl	*összesen: 0*
	Ha *nincs forrás* fájl, de EXE van, akkor a büntető súly (**Ps**)	**Ps**=0.5
	Ha *nem fordítható le és EXE sincs*, akkor összpont (utólag pótolnia kell)	*összesen: 0*
Ps*Pr	**Minimum..Maximum**	**0..30**
Dokumentáció	*Ha nincs, akkor nincs specifikáció, nincs algoritmus és nincs ellen-őrizhető kód. Ezért azok pontjai elvesznek.*	*Összesen: 0..30*
D	Fejlesztői teljessége	0..12
	Felhasználói teljessége	0..11
	Igényesség (a teljes dokumentációé; pl. ábrák vannak-e…)	0..7
D	**Minimum..Maximum**	**0..30**
Specifikáció	A feladatnak nem megfelelő specifikáció büntető súlya (**Ss**)	**Ss**=0..1
S	Rekurzív program-specifikáció	0..10
	Bemeneti, kimeneti fájlok szerkezetének megadása	0..5
	Formalizáltság (hatékony, leírást rövidítő fogalmak; matematizáltság)	0..5
Ss*S	**Minimum..Maximum**	**0..20**
Algoritmus	A specifikációtól (a megadott ábrázolástól) eltérés büntető súlya (**As**) ...	**As**=0.5..1
	Algoritmus nincs, a büntető súly (**As**=0.5, és A=0).....	**As**=0.5,**A**=0
	Ha nem felülről-lefelé tervez, akkor	A=0
A	Tételek alkalmazása tükröződik az algoritmuson	0..5
	A ciklusok rekurzívan megoldottak.	0..10
	Típusok körültekintő választása, jól definiálása és megvalósítása	0..5
	Hibákért (pl. nem megengedett szerkezetek...) egyedi levonások	-10..0
Ss*As*A	**Minimum..Maximum**	**0..20**
Kód *(doku. alapján)*	Az algoritmustól eltérés büntető súlya (**Ks**).....	**Ks**=0..1
	Ha nem ellenőrizhető a kód, akkor e rész összpontja.....	**K**=0
K	A típusok megvalósítása (logikus unitokra… szétszedett kód)	0..10
	Külön ablakos, igényes tartalmú tájékoztató.....	0..8
	Barátságos (beolvasás, visszajelzés, eredménymegjelenítés)	0..7
	Biztonságos (beolvasás, file-megnyitás).....	0..5
	Fájlos hibafigyelés és –jelzés.....	0..5
	Hibákért (GOTO, EXIT...) egyedi levonás	-5..0
	Több hiba-leállási pont.....	-2
	Ha egyetlen fájlba „gyömöszölte”…	-10
Ss*As*Ks*K	**Minimum..Maximum**	**0..35**
Pluszokért: Pl	(kódfájlok, menük, help...)	**0..10**
Beadandó:	**Összpontszám, Minimum..Maximum: (Ps*Pr+D+Ss*S+Ss*As*A+Ss*As*Ks*K+Pl)**	**0..135+10**

6. ÉRTÉKELÉS

Alsóhatár	Felsőhatár	Jegy
115	..	5
100	114	4
80	99	3
60	79	2

..	59	1

Fontos: a beadandókat szóban is meg kell védeni. Ennek időpontját és helyét egyeztetjük.