

DEBRECENI EGYETEM  
TANÁRKÉPZÉSI KÖZPONT

CSERNOCH MÁRIA

**Algoritmusok és sémák  
az informatika oktatásában II.**

Algoritmusalapú szöveg- és táblázatkezelés



Debreceni Egyetemi Kiadó  
Debrecen University Press  
2015

## Szaktárnet-könyvek 18/2.

Sorozatszerkesztő:

**Maticsák Sándor**

Készült

a SZAKTÁRNET (TÁMOP-4.1.2.B.2-13/1-2013-0009)  
pályázat keretében

Technikai szerkesztő:

**Buzgó Anita**

Borítóterv:

**Nagy Tünde**

© Csernoch Mária

© Debreceni Egyetemi Kiadó – Debrecen University Press,  
beleértve az egyetemi hálózaton belüli elektronikus terjesztés jogát is.

Kiadta a Debreceni Egyetemi Kiadó, az 1795-ben alapított  
Magyar Könyvkiadók és Könyvterjesztők Egyesülésének tagja.  
[www.dupress.hu](http://www.dupress.hu)

Felelős kiadó: Karácsony Gyöngyi  
Készült a Kapitális Nyomdában, 2015-ben.



## Tartalomjegyzék

1. Alkalmazások közötti átjárhatóság (13–16. óra) .....	5
1.1. Dokumentum fájlok kezelése (13. óra).....	5
1.1.1. Bevezetés.....	5
1.1.2. Jellemzők.....	5
1.1.3. Alapértelmezés szerinti használat.....	5
1.1.4. Tévhitek.....	6
1.1.5. MS hibaüzenet.....	7
1.1.6. Megjegyzések az üzenethez .....	7
1.2. Konvertálási feladatok előkészítése (14. óra) .....	9
1.2.1. Adat lekérdezés: hibás szöveges dokumentumok.....	9
1.2.2. Mi indokolja a konvertálást? .....	10
1.3. Konvertálási feladatok (15. óra).....	11
1.3.1. Ötöslottó nyerőszámai.....	11
1.3.2. Legnépszerűbb mozifilmek .....	13
1.4. Weblapok konvertálása táblázatkezelő munkalappá (16. óra).....	14
1.4.1. Weblapok közvetlen konvertálásának lépései .....	16
1.5. Gyakorlatok I: Weblapok többlépcsős konvertálása (27. óra).....	22
1.5.1. IMDb.....	23
1.5.2. Menetdíj.....	25
1.5.3. Megyék.....	28
1.5.4. Könyvek – Best Books Ever.....	32
2. Szöveges dokumentumok hibatípusai (17–20. óra).....	43
2.1. Szövegkezelés (17. óra) .....	43
2.1.1. Mit csináltunk rosszul?.....	43
2.1.2. Szöveges e-dokumentumok.....	45
2.1.3. Helyesen formázott dokumentum.....	45
2.1.4. Definíció hiányának következményei.....	45
2.1.5. Definíció .....	48
2.2. Szöveges e-dokumentumok hibái, hibafelismerés (18. óra) .....	48
2.2.1. Lehetséges hibaforrások.....	49
2.2.2. Hibakategóriák megnevezése .....	64
2.2.3. Hibafelismerés .....	64
2.3. Szövegszerkesztés lépései (19. óra) .....	66
2.3.1. Szövegbevitel.....	66

2.3.2. Hibajavítás.....	69
2.3.3. Formázás .....	70
2.3.4. Tévhitek.....	71
2.4. Hibafelismerés, hibajavítás (20. óra).....	75
2.4.1. Gyógyszerek.....	76
2.4.2. NLP felhívás.....	80
2.4.3. Mozirajongók.....	91
3. Táblázatkezelés mélyszerkezetű metakognitív megközelítése (21–24. óra).....	99
3.1. Sprego programozás (21. óra).....	100
3.1.1. Sprego eszközei.....	101
3.1.2. Sprego függvények.....	102
3.1.3. Összetett függvények.....	103
3.1.4. Tömbképletek.....	103
3.1.5. Tesztelés.....	106
3.1.6. Autentikus táblázatok.....	109
3.2. Sprego feladatok (22–24. óra).....	110
3.2.1. US states (22–23. óra).....	110
3.2.2. Sprego feladatok: Ötöslottó (24. óra).....	124
3.3. Weblapok megnyitása és kezelése táblázatkezelőben (28. óra). 128	
3.3.1. Megyék.....	128
3.3.2. Best Books Ever.....	133
Irodalomjegyzék.....	143
Mellékletek.....	153

## **1. Alkalmazások közötti átjárhatóság (13–16. óra)**

### **1.1. Dokumentum fájlok kezelése (13. óra)**

#### **1.1.1. Bevezetés**

A konvertálások elsődleges célja fájlok átalakítása további adatfeldolgozásra. Jellemzője, hogy a végső felhasználás az eredetitől eltérő fájl formátumban vagy az eredetitől eltérő szerkezetben történik. A végső felhasználáson túl előnye még a konvertálási feladatoknak, hogy a konvertálás során akár többször is indokolt lehet a fájl formátumának megváltoztatása, valamint az, hogy a megnyitásánál rendszerint nem a dokumentumfájloknál szokásos duplakattintást használjuk, hanem a hagyományos *Megnyitás* parancsot.

A dokumentumok megnyitására leggyakrabban használt a duplakattintás, amely feltételez egy grafikus felületet, amelyen ezt a műveletet el lehet végezni, valamint a dokumentumfájl létezését. Sajnálatos módon azonban, a felhasználók többsége nem ismeri a dokumentumfájl fogalmát.

Definíció: A dokumentumfájl egy olyan adatfájl, amelyhez egy program van társítva.

#### **1.1.2. Jellemzők**

- A társítás a kiterjesztés alapján történik.
- A dokumentumfájl nem egy abszolút tulajdonsága a fájlnek, hanem a szoftveres környezet függvénye. Ennek következtében előfordulhat, hogy egy adatfájl az egyik számítógépen dokumentumfájl, míg egy másikon nem, annak függvényében, hogy a számítógépen milyen programok vannak installálva, illetve, hogy megtörtént-e a társítás, és milyen programhoz lett az adott kiterjesztés társítva.
- A dokumentumfájl, mint fogalom a GUI (Graphical user interface) megjelenéséhez, és azon belül a dokumentumfájlok alapértelmezés szerinti használatához köthető.

#### **1.1.3. Alapértelmezés szerinti használat**

A dokumentumfájltra duplán kattintva elindul a hozzá társított program, majd azon belül megnyitásra kerül a fájl. Mindez az operációs rendszer felügyelete alatt történik.

#### 1.1.4. Tévhitek

Gyakran hallott tévhit, hogy a dokumentumfájlhoz az a program van társítva, amellyel azt létrehozták. Ez az állítás esetenként igaz is lehet, de nem jellemzően. Ennek az állításnak a cáfolására az egyik példa a weblapok társítása. A weblapokhoz társított programok az esetek nagy részében a böngészők valamelyike, míg egészen biztos, hogy a weblapok nem böngészőkkel készültek. A felhasználói igénynek megfelelően, a weblapokhoz természetesen egy weblapszerkesztő program is hozzátársítható, de akár egy szövegszerkesztő vagy táblázatkezelő program is. Hasonló a helyzet a különböző olvasó és lejátszó programokkal. A .pdf fájlokhoz a leggyakoribb, hogy egy olvasó program van hozzátársítva, és csak speciális esetekben fordul elő, hogy egy .pdf szerkesztő program. A video, hang, kép és animációs fájlokhoz alapértelmezésben nem ezek szerkesztője, hanem a lejátszó programja van hozzátársítva, de ezekben az esetekben is a felhasználói igények döntenek el, hogy melyik program kerül társításra egy adott konfigurációban.

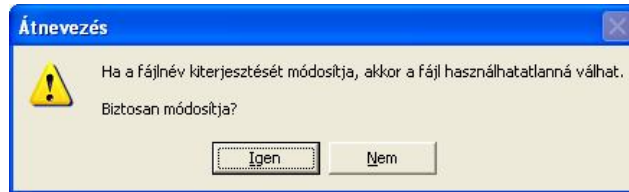
További gyakori tévhit, hogy ha a dupla kattintás nem működik, és annak hatására nem sikerül az adatfájlt megnyitni, akkor a fájl használhatatlan, illetve csak társítással tehető használhatóvá. Túlértékeljük a dupla kattintás és a társítás jelentőségét. Mindkettő csak egy kényelmi szolgáltatásnak a GUI-nak, és nincs kapcsolata a fájl megnyithatóságához. Egyrészt nem-grafikus felületeken is meg tudunk nyitni tetszőleges adatfájlt, ha van arra alkalmas programunk, másrészt a GUI-ban is megnyithatóak az adatfájlok anélkül, hogy kihasználnánk, hogy azok dokumentumfájlok. Mivel a dokumentumfájlok létezéséhez szükséges, hogy a fájlnek kiterjesztése legyen, ezért a kiterjesztés nélküli fájl esetében is csak a hagyományos *Megnyitás* paranccsal tudjuk a fájlokat megnyitni.

Kérdés természetesen, hogy honnan lehetnek kiterjesztés nélküli fájljaink. Bármennyire is meglepő, de a kiterjesztés nélküli fájlknak az egyik leggyakoribb forrása a Microsoft. A Microsoft igyekszik azt a látszatot kelteni, hogy fájlnevek bárki használhatja a saját anyanyelvének bármelyik karakterét, valamint tetszőleges egyéb karaktereket is. Ennek azonban ára van, melyet olyan nyelvterület kerül a fájl, amely alapértelmezésben nem támogatja a Standard ASCII karaktereken túliakat. A hotmail például azt csinálja, hogy ha talál a fájlnevében az angol betűkön kívül egyéb karaktereket – tovább szűkítve a Standard ASCII karakter-

készletet, akkor attól a karaktertől jobbra mindent levág és eldob, így a fájl kiterjesztését is.

### 1.1.5. MS hibaüzenet

A valaha látott legértelmetlenebb és félrevezetőbb hibaüzenetet a Microsoft produkálja, és ez éppen a dokumentumfájlokhoz köthető.



**1. ábra**

*Fájl kiterjesztésének módosításakor megjelenő MS üzenet*

### 1.1.6. Megjegyzések az üzenethez

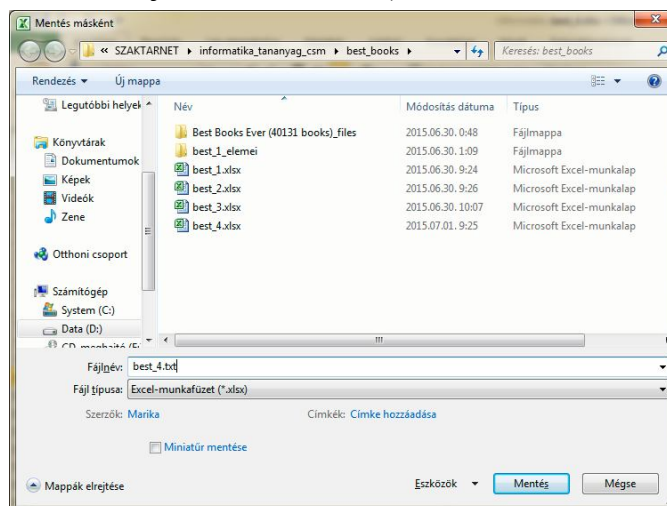
Az üzenet értelmezéséhez néhány további gondolatot az alábbiakban foglalok össze.

- A felhasználók többsége azt hiszi, hogy a kiterjesztés és a fájl típusa között egyértelmű megfeleltetés van, ez azonban nem így van. Ennek megfelelően, ha a fájl típusát meg akarjuk változtatni, akkor azt nem a kiterjesztés átírásával kell elvégezni, hanem konvertálással.
- A fájl típus módosítását az adatfájl megnyitása után a Mentés másként paranccsal végezzük, majd kiválasztjuk a felkínált listából az új fájl típust. Erre a műveletsorra mutatnak példákat jelen tananyag további fejezetei.
- Ha valamilyen okból mégis indokolt a fájl kiterjesztésének módosítása, akkor az biztosan igaz, hogy ezzel a módosítással a fájl nem sérül, és biztosan nem lesz használhatatlan. Egyetlen dolog történik, egy másik program lesz a fájlhoz társítva, vagy esetleg nem lesz társítás. Ez nem egy nagy veszteség. Az egyik leggyakoribb indok a fájl kiterjesztésének átírására, eltávolítására, hogy a levelező programok bizonyos típusú fájlok csatolását nem támogatják.
- A kiterjesztés átírását követően a fájl továbbra is használható lesz, az egyetlen különbség a korábbiakhoz, hogy a dupla kattintás kényelméről lemondunk. Dokumentumfájlok esetén a programot mi ma-

gunk indítjuk el, és azon belül mi magunk adjuk ki a Megnyitás parancsot.

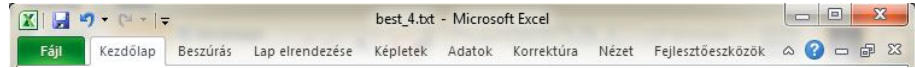
- Ha valaki egy futtatható állomány kiterjesztését írja át, akkor tényleg adódhatnak problémák a program indításával, de ebben az esetben sem sérül a fájl, és nem lesz használhatatlan. Az a legegyszerűbb megoldás, ha futtatható fájlok esetén visszaírjuk az eredeti kiterjesztést.
- A Microsoft egyik óriási módszertani tévedése, hogy a felhasználókat tudatlanságban akarja tartani. Számos esetben megtapasztalhattuk már, hogy a problémás esetekben nem arra törekszik, hogy tudatos felhasználókat neveljen, hanem éppen ezzel ellenkezőleg, a problémás eseteket, helyzeteket igyekszik elkerülni, elrejteni. Ebben az esetben is ez történt. Azt tapasztalta a Microsoft, hogy a felhasználók megváltoztatták a fájlok kiterjesztését, és utána nem tudták a dupla kattintással megnyitni. Erre a Microsoft úgy döntött, hogy elrejtje a kiterjesztést. Óriási hiba volt (2. ábra és 3. ábra).
- A Microsoft ezzel csak a saját fájlkezelési problémáit próbálja palástolni.

Mindezeket figyelembe véve a Microsoft a saját hibáit próbálja a felhasználóra hárítani, és ez a megfelelő szintű digitális műveltség hiányának következtében, sajnos működik is. :(



**2. ábra**

*Gépeltek .txt karaktersorozat a fájl neve után nem változtatja meg a fájl típusát*



### 3. ábra

*A fájl kiterjesztésének elrejtésével a tényleges fájltypus nem jelenik meg a fájlnevben*

## 1.2. Konvertálási feladatok előkészítése (14. óra)

### 1.2.1. Adat lekérdezés: hibás szöveges dokumentumok

Szövegszerkesztés tanítása során hibakereséshez, -javításhoz mindenképpen szükség van autentikus hibás szövegekre. A hibás szövegek elsődleges forrása az internet, ahol bőségesen találni ilyen dokumentumokat. A megfelelő dokumentumok kiválasztása azonban, éppen a hibás dokumentumok és a hibák nagy száma miatt, rendkívül időigényes. Nem egyszerű megtalálni azt a dokumentumot, amelyik tökéletesen megfelel az elképzeléseinknek, módszertani elvárásoknak. Ennek nagyon egyszerű a magyarázata; a hibás dokumentumok többsége halmozottan hibás, ami annyit jelent, hogy több különböző forrásból származó hibát is tartalmaznak.

#### 1.2.1.1. Válogatási szempontok

A hibás dokumentumok kereséséhez, az ilyen dokumentumok nagy száma mindenképpen indokoltá teszi a válogatási szempont előzetes megfogalmazását. Amennyiben szövegszerkesztés-oktatáshoz keresünk hibás dokumentumokat a következő válogatási szempontokat érdemes szem előtt tartani:

- az oktatási célok, amelyek egyértelműen megfogalmazzák, hogy mi az új ismeret,
- az óra helye a tanítási folyamatban, amelyből egyértelmű, hogy melyek a korábban szerzett ismeretek, és az, hogy a későbbiekben miért lesz erre az anyagra szükség.
- nevelési célok, amelyek nagyban befolyásolják a kiválasztott szöveg tartalmát.

Mindezen szempontoknak eleget tenni nem egy egyszerű feladat. Érdemes tehát egy szöveggyűjteményt készíteni, kategorizálva, amelyben a megtalált szövegeket gyűjtögethetjük.

#### **1.2.1.2. Tapasztalatok**

Érdemes megfigyelni a tanulók reakcióit egy olyan házi feladat esetén, amely azt kéri, hogy keressenek adott szempontnak megfelelő hibás Word dokumentumokat az interneten. Az alábbiakban röviden összefoglalom a tapasztalataimat, amelyek összességében azt mutatják, hogy a tanulók milyen szintű digitális műveltséggel rendelkeznek ebben a témában.

- Nagyon kevés helyes dokumentum érkezik.
  - Nem autentikus Word dokumentumok.
  - Nem felelnek meg a feltételeknek, tehát olyan hibákat is tartalmaznak, amelyek nem szerepeltek az eredeti feladatban.
  - A tanulók magyarázatai
    - Nem találtunk az interneten Word dokumentumokat, ezért weblapból konvertáltunk vagy másoltunk.
    - Nem találtunk az interneten hibás Word dokumentumokat, hiába gépeltük be a kereső mezőbe.
  - Következtetések
    - Nem tudják a tanulók, hogy hogyan lehet megadott típusú fájlokra rákeresni, nem ismerik a speciális keresést.
    - Nem tudják, hogy milyen kulcsszavakat érdemes begépelni ahhoz, hogy értelmezhető dokumentumokhoz jussanak. A legtöbben a kereső mezőbe azt gépelték, hogy „hibás word dokumentum”.
- Ezek a példák is azt mutatják, hogy a tanulók keresési hatékonysága rendkívül alacsony.

#### **1.2.2. Mi indokolja a konvertálást?**

A következő alfejezetekben olyan példákat mutatunk be, amelyek indokoltá teszik szöveges fájlok konvertálását. A fájlok konvertálása módszertani szempontból kiemelt jelentőségű. Ez egyik legfontosabb, hogy a konvertálások során végrehajtott műveletek megkövetelik a különböző típusú adatfájlok kezelését, ami a grafikus környezet előretörésével háttérbe szorult (1.1.1. fejezet). A felhasználók többsége nem tud különbséget tenni program- és adatfájlok között, és nem ismeri az adatfájlokon



végrehajtható különböző műveleteket. A másik magyarázat az, hogy a konvertálás során egyik formátumból a másikba alakított fájlok lehetővé teszik, hogy adatvesztés nélkül felhasználjunk és feldolgozzunk adatokat a számunkra alkalmas környezetben. Különösen nagy jelentősége van a weblapokon elérhető adatoknak, mivel ezeket a fájlformátumból adódóan csak olvasásra készítették, de gyakran adódnak olyan esetek, amikor ezen adatok további feldolgozása válik indokolttá.

A további fejezetekben olyan weblapok letöltését és konvertálását mutatjuk be, amelyek alkalmasak arra, hogy irodai szoftverkörnyezetben (röviden: birotikai programok) fejlesszük a tanulók algoritmikus készségét. Megmutatva egyrészt, hogy az algoritmikus készség fejlesztése nem a programozás-oktatás privilégiuma, másrészt, hogy a birotikai szoftverek sokkal hatékonyabban tanulhatók és használhatók az algoritmikus megközelítéssel, mint a hagyományos felületi megközelítésekkel.

Mindenképpen fontosnak tartom azonban megjegyezni, hogy a bemutatott módszerek csak egy-egy lehetőség, ötlet, és további sok-sok algoritmusalapú megoldás létezik, az olvasóra bízva azok kitalálását, kipróbálását.

### 1.3. Konvertálási feladatok (15. óra)

#### 1.3.1. Ötöslottó nyerőszámai

Az ötöslottó nyerőszámai, nyereményei heti lebontásban letölthetők a Szerencsejáték Zrt. weblapjáról. A weblap három különböző formátumot ajánl fel (xls<sup>1</sup>, csv<sup>2</sup>, html<sup>3</sup>), amelyek közül bármelyik megnyitható MS Excellel, OpenOffice, LibreOffice Calc-kal.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	2012	30	2012.07.28	0	0 Ft	35	1 614 650 Ft	2996	19 970 Ft	90394	1 285 Ft	7	33	35	83	89
2	2012	29	2012.07.21	0	0 Ft	26	2 207 860 Ft	3658	16 615 Ft	101268	1 165 Ft	19	43	46	47	52
3	2012	28	2012.07.14	0	0 Ft	22	2 938 155 Ft	2661	25 720 Ft	81847	1 625 Ft	20	44	52	65	83
4	2012	27	2012.07.07	0	0 Ft	24	2 274 725 Ft	2379	24 300 Ft	72903	1 540 Ft	17	29	74	76	82
5	2012	26	2012.06.30	0	0 Ft	26	2 116 750 Ft	2339	24 915 Ft	82221	1 380 Ft	11	12	31	49	58
6	2012	25	2012.06.23	0	0 Ft	37	1 523 825 Ft	3393	17 595 Ft	90960	1 275 Ft	19	24	47	60	88
7	2012	24	2012.06.16	2	2 057 602 470 Ft	67	1 587 555 Ft	5615	20 060 Ft	157973	1 385 Ft	19	26	53	60	89
8	2012	23	2012.06.09	0	0 Ft	39	2 752 115 Ft	5498	20 670 Ft	168644	1 310 Ft	14	21	22	58	64
9	2012	22	2012.06.02	0	0 Ft	45	2 186 345 Ft	4376	23 805 Ft	129733	1 560 Ft	18	20	30	42	51

<sup>1</sup> Az Ötöslottó eddigi nyerőszámai. <http://www.szerencsejatek.hu/xls/otos.xls>. Letöltve 2012. augusztus 3.

<sup>2</sup> Az Ötöslottó eddigi nyerőszámai. <http://www.szerencsejatek.hu/xls/otos.csv>. Letöltve 2012. augusztus 3.

<sup>3</sup> Az Ötöslottó eddigi nyerőszámai. <http://www.szerencsejatek.hu/xls/otos.html>. Letöltve 2012. augusztus 3.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Az ötöslottó eddigi nyerőszámai															
2																
3	Év	Hét	Húzásdátum	5 találat (db)	5 találat (Ft)	4 találat (db)	4 találat (Ft)	3 találat (db)	3 találat (Ft)	2 találat (db)	2 találat (Ft)	Számok				
4	2012	30	2012.07.28	0	0 Ft	35	1 614 650 Ft	2996	19 970 Ft	90394	1 285 Ft	7	33	35	83	89
5	2012	29	2012.07.21	0	0 Ft	26	2 207 860 Ft	3658	16 615 Ft	101268	1 165 Ft	19	43	46	47	52
6	2012	28	2012.07.14	0	0 Ft	22	2 938 155 Ft	2661	25 720 Ft	81847	1 625 Ft	20	44	52	65	83
7	2012	27	2012.07.07	0	0 Ft	24	2 274 725 Ft	2379	24 300 Ft	72903	1 540 Ft	17	29	74	76	82
8	2012	26	2012.06.30	0	0 Ft	26	2 116 750 Ft	2339	24 915 Ft	82221	1 380 Ft	11	12	31	49	58
9	2012	25	2012.06.23	0	0 Ft	37	1 523 825 Ft	3393	17 595 Ft	90960	1 275 Ft	19	24	47	60	88
10	2012	24	2012.06.16	2	2 057 602 470 Ft	67	1 587 555 Ft	5615	20 060 Ft	157973	1 385 Ft	19	26	53	60	89
11	2012	23	2012.06.09	0	0 Ft	39	2 752 115 Ft	5498	20 670 Ft	168644	1 310 Ft	14	21	22	58	64
12	2012	22	2012.06.02	0	0 Ft	45	2 186 345 Ft	4376	23 805 Ft	129733	1 560 Ft	18	20	30	42	51

#### 4. ábra

*Az ötöslottó eddigi nyerőszámai táblázat .csv (felső) és .html (alsó) formátumú verziói MS Excelben megnyitva*

Az ötöslottó nyerőszámai táblázat .csv (4. ábra felső) és .html (4. ábra alsó) formátuma minden konverzió nélkül használható a táblázatkezelő programokban<sup>4</sup>, míg, nem túl szerencsés módon, éppen az .xls formátum az, amely konverziót igényel (5. ábra felső).

Az .xls formátumú táblázatot megnyitva a dátumok és a számok is szöveggént jelennek meg (5. ábra felső). A számok átalakítását el lehet végezni, viszonylag gyorsan a helyi menü *Átalakítás számmá* parancsával. A dátum konverziójához a – (kötőjel) karaktereket kell lecserélni . (pont) karakterre. Talán a leggyorsabb megoldás, ha a karakterek cseréjét a *Csere* paranccsal végezzük el (5. ábra alsó). Annak eldöntése, hogy parancsokkal vagy függvényekkel végezzük el a konverziót, minden esetben az oktatási és a nevelési célok függvénye.

<sup>4</sup> A fájlok megnyitását követően azonnal érdemes azokat a táblázatkezelő program alapértelmezett formátumában lementeni, hogy a későbbiekben a képletek biztonságosan menthetőek legyenek.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	év	hét	húszadátum	5 találat (db)	5 találat (Ft)	4 találat (db)	4 találat (Ft)	3 találat (db)	3 találat (Ft)	2 találat (db)	2 találat (Ft)	számok				
2	2012	30	2012-07-28	0	0 Ft	35	16146 Ft	2996	19 970 Ft	90394	1 285 Ft	7	33	35	83	89
3	2012	29	2012-07-21	0	0 Ft	26	2 207 860 Ft	3658	16 615 Ft	101268	1 165 Ft	19	43	46	47	52
4	2012	28	2012-07-14	0	0 Ft	22	2 938 155 Ft	2661	25 720 Ft	81847	1 625 Ft	20	44	52	65	83
5	2012	27	2012-07-07	0	0 Ft	24	2 274 725 Ft	2379	24 300 Ft	72903	1 540 Ft	17	29	74	76	82
6	2012	26	2012-06-30	0	0 Ft	26	2 116 750 Ft	2339	24 915 Ft	82221	1 380 Ft	11	12	31	49	58
7	2012	25	2012-06-23	0	0 Ft	45	1 523 825 Ft	3393	17 595 Ft	90960	1 275 Ft	19	24	47	60	88
8	2012	24	2012-06-16	2	2 057 602 470 Ft	27	1 587 555 Ft	5615	20 060 Ft	157973	1 985 Ft	19	26	53	60	89
9	2012	23	2012-06-09	0	0 Ft	39	2 752 115 Ft	5498	20 670 Ft	168644	1 310 Ft	14	21	22	58	64
10	2012	22	2012-06-02	0	0 Ft	45	2 186 345 Ft	4376	23 805 Ft	129733	1 560 Ft	18	20	30	42	51
11	2012	21	2012-05-26	0	0 Ft	55	1 792 311 Ft	6589	15 270 Ft	161858	1 205 Ft	11	36	43	63	66
12	2012	20	2012-05-19	0	0 Ft	84	1 127 870 Ft	6589	15 270 Ft	161858	1 205 Ft	11	36	43	63	66

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	év	hét	húszadátum	5 találat (db)	5 találat (Ft)	4 találat (db)	4 találat (Ft)	3 találat (db)	3 találat (Ft)	2 találat (db)	2 találat (Ft)	számok				
2	2012	30	2012-07-28	0	0 Ft	35	1 614 650 Ft	2996	19 970 Ft	90394	1 285 Ft	7	33	35	83	89
3	2012	29	2012-07-21	0	0 Ft	26	2 207 860 Ft	3658	16 615 Ft	101268	1 165 Ft	19	43	46	47	52
4	2012	28	2012-07-14	0	0 Ft	22	2 938 155 Ft	2661	25 720 Ft	81847	1 625 Ft	20	44	52	65	83
5	2012	27	2012-07-07	0	0 Ft	24	2 274 725 Ft	2379	24 300 Ft	72903	1 540 Ft	17	29	74	76	82
6	2012	26	2012-06-30	0	0 Ft	26	2 116 750 Ft	2339	24 915 Ft	82221	1 380 Ft	11	12	31	49	58
7	2012	25	2012-06-23	0	0 Ft	45	1 523 825 Ft	3393	17 595 Ft	90960	1 275 Ft	19	24	47	60	88
8	2012	24	2012-06-16	2	2 057 602 470 Ft	27	1 587 555 Ft	5615	20 060 Ft	157973	1 985 Ft	19	26	53	60	89
9	2012	23	2012-06-09	0	0 Ft	39	2 752 115 Ft	5498	20 670 Ft	168644	1 310 Ft	14	21	22	58	64
10	2012	22	2012-06-02	0	0 Ft	45	2 186 345 Ft	4376	23 805 Ft	129733	1 560 Ft	18	20	30	42	51

5. ábra

Az Ötöslottó eddigi nyerőszámai táblázat .xls formátumú verziói MS Excelben megnyitva (felső), majd a konvertálást követően (alsó)

1.3.2. A legnépszerűbb mozifilmek

Az IMDb a 250 legnépszerűbb mozifilm listáját tartalmazza.<sup>5</sup> Rendkívül érdekessé teszi a táblázatot, hogy egyrészt folyamatosan változik a tartalma a szavazatok számától függően – 6. ábra ugyanazon napi két letöltést mutatja –, másrészt pedig, hogy a böngésző nyelvének beállításától függően változik a filmcímek nyelve (6. ábra). Ez annyit jelent, hogy ha a böngésző nyelve magyarra van állítva, akkor a filmcímek magyarul és minden egyéb angolul jelenik meg a lapon. Ez egy rendkívül érdekes kombinációja az angol és a magyar nyelvnek, amelynek következménye, hogy az írásjelek eltérő használata miatt megnehezíti a tartalom értelmezését. A legnagyobb problémát a szavazatok számában található vessző jelenti. A vesszővel az a gond, hogy míg az angol nyelvben ezreselválasztó karakter, addig a magyarban tizedes vessző<sup>6</sup>. Ennek következménye, hogy a weblapot táblázatkezelőben megnyitva elveszítjük a legkisebb he-

<sup>5</sup> Top 250 movies as voted by our users. <http://www.imdb.com/chart/top>. Letöltve: 2012. július 22. Időközben sajnos megváltozott a weblap tartalma, aminek következtében a táblázat már nem tartalmazza a szavazatok számát (2014. május 1.).

<sup>6</sup> Az alapértelmezés szerinti tizedes és az ezreselválasztó karakterek módosíthatóak (Fájl → Beállítások → Speciális → Szerkesztés → Tizedesjel és Ezreselválasztó), ebben az esetben viszont számolnunk kell azzal, hogy a munkafüzetünk nem lesznek kompatibilisek.

lyiértéken található nullákat (6. ábra és 7. ábra, bal oldali panel, 5. és 6. film). Tehát ha az eredeti érték legkisebb helyiértéke 0 alaki értékű, akkor az eredeti szám tizede jelenik meg a cellában a konvertálás után. Számítálan megoldása létezik a problémának. Az érdeklődő olvasók számára a legegésőbb, talán nem a legrövidebb, de ugyanakkor a legbiztonságosabb konvertálást könyvünk **0. fejezete** mutatja be. A konvertálás utáni helyes táblázatot, a *top\_movies.xlsx* állomány *top\_movies* munkalapjának egy részletét, a 7. ábra jobb oldali mintája mutatja.

Rank	Rating	Title	Votes
1.	9.2	<a href="#">The Shawshank Redemption</a> (1994)	790,425
2.	9.2	<a href="#">The Godfather</a> (1972)	585,045
3.	9.0	<a href="#">The Godfather: Part II</a> (1974)	371,343
4.	8.9	<a href="#">Pulp Fiction</a> (1994)	621,174
5.	8.9	<a href="#">The Good, the Bad and the Ugly</a> (1966)	244,920
6.	8.9	<a href="#">12 Angry Men</a> (1957)	194,680
7.	8.9	<a href="#">Schindler's List</a> (1993)	414,133
8.	8.8	<a href="#">The Dark Knight</a> (2008)	736,027

Rank	Rating	Title	Votes
1.	9.2	<a href="#">A remény rabjai</a> (1994)	790,965
2.	9.2	<a href="#">A keresztapa</a> (1972)	585,385
3.	9.0	<a href="#">A keresztapa 2.</a> (1974)	371,607
4.	8.9	<a href="#">Ponyvaregény</a> (1994)	621,546
5.	8.9	<a href="#">A jó, a rossz és a csúf</a> (1966)	245,072
6.	8.9	<a href="#">Tizenkét dühös ember</a> (1957)	194,820
7.	8.9	<a href="#">Schindler listája</a> (1993)	414,375
8.	8.8	<a href="#">A sötét lovag</a> (2008)	738,327

### 6. ábra

*Az IMDb 250 legnépszerűbb filmjének listája. A filmcímek attól függően jelennek meg angolul (bal) vagy magyarul (jobb), hogy mi a böngésző beállított nyelve. A böngésző nyelvbéállításaitól függetlenül a számok az angol írásmóddal jelennek meg, mely szerint az ezres elválasztó karakter a vessző*

	A	B	C	D		A	B	C	D
1	Rank	Rating	Title	Votes	1	Rank	Rating	Title	Votes
2	1	9,2	The Shawshank Redemption (1994)	790,425	2	1	9,2	The Shawshank Redemption (1994)	790425
3	2	9,2	The Godfather (1972)	585,045	3	2	9,2	The Godfather (1972)	585045
4	3	9	The Godfather: Part II (1974)	371,343	4	3	9	The Godfather: Part II (1974)	371343
5	4	8,9	Pulp Fiction (1994)	621,174	5	4	8,9	Pulp Fiction (1994)	621174
6	5	8,9	The Good, the Bad and the Ugly (1966)	244,92	6	5	8,9	The Good, the Bad and the Ugly (1966)	244920
7	6	8,9	12 Angry Men (1957)	194,68	7	6	8,9	12 Angry Men (1957)	194680
8	7	8,9	Schindler's List (1993)	414,133	8	7	8,9	Schindler's List (1993)	414133
9	8	8,8	The Dark Knight (2008)	736,027	9	8	8,8	The Dark Knight (2008)	736027

### 7. ábra

*Az MS Excel automatikusan elvégzi az adatok konvertálását, amelynek következtében elveszítjük a legkisebb helyiértéken található nullákat. A hibás konvertálás eredményét a bal oldali, míg a helyes adatokat a jobb oldali minta mutatja*

## 1.4. Weblapok konvertálása táblázatkezelő munkalappá (16. óra)

Weblapok munkalapokká konvertálására a különböző táblázatkezelő programok különböző verziói újabb és újabb lehetőségeket, rendszerint parancsokat, kínálnak fel. Ezek a módszerek is működnek, de egyrészt

verziófüggők, másrészt parancsok sorozatának memorizálása továbbra sem egy szerencsés dolog, úgy is elfeleltjük. A szakirodalom a problémamegoldások ilyen típusú megközelítését a felületi megközelítések csoportjába sorolja, és mint ilyenek kevésbé hatékonyak, mint a mély megközelítések (Case–Gunstone–Lewis 2001: 331–355, Case–Gunstone 2002: 459–470, Case–Gunstone 2003: 801–819, Case: 2000, Csernoch–Bíró 2013).

Ebben a fejezetben egy olyan módszer kerül bemutatásra, amely a táblázatkezelő program verziójától és típusától független működik, egy teljesen általánosan használható megoldás. A módszer a különböző fájlípusok tulajdonságain és azok konvertálhatóságán alapszik.

A weblapok konvertálásához elengedhetetlen annak ismerete, hogy a táblázatkezelő programok minden gond nélkül meg tudják nyitni a weblapokat. Ez a tény azért lényeges, mert így a konvertálást érdemes teljes weblapok mentésével kezdeni. Nincs szükség arra, hogy a weblapon elhelyezkedő táblázatot hosszas bűvészkedéssel kijelöljük, majd másolgassuk. A másolásnak az egyik nagy hátránya, hogy táblázatok weblapon történő kijelölése rendkívül időigényes is lehet, míg másik hátránya az esetleges adatvesztés. Ezzel szemben teljes weblapok mentése egyrészt gyors, másrészt nem fordulhat elő adatvesztés.

A weblapok konvertálásának sikeressége azonban nagyban függ a weblap tartalmától. A tartalomtól függően az is előfordulhat, hogy a weblapból közvetlenül nem tudjuk a munkalapot létrehozni, hanem egyéb köztes konvertálási lépéseket kell elvégezni. Erre láttunk példát az IMDb weblap konvertálásakor, amikor a számok írására használt karakterek megakadályozták a közvetlen konvertálást. Ennek részleteit az 1.5. alfejezetben mutatjuk be.

Azon táblázatok konvertálásánál, amelyek nem tartalmaznak számok közötti elválasztó karaktereket nincs különösebb probléma. A tizedes vessző és pont, a dátum és idő formátumok, valamint az ezreselválasztó szóközők – normál és nem-törhető – és vessző, az eltérő magyar és angol írásmódból következően, rendkívüli módon meg tudja nehezíteni a fájlok közötti átjárhatóságot (A magyar helyesírás szabályai 2005). Természetesen a számok különböző írásmódjából származó konvertálási problémák nem csak a magyar és az angol nyelv sajátjai. Ezekkel a problémákkal kell számolni minden olyan esetben, amikor az adathalmaz és a feldolgozás nyelve nem egyezik meg (Keszler 2003: 2–36).

	magyar (európai)	angol
egész számok	78 262 264	78,262,264
tizedes törtek	456,92	456.92
dátum	1983. 11. 28.	11/28/1983, 28/11/1983
idő	13.46	13:46

### 1. táblázat

*Számok, dátum és idő formátumok a magyar és az angol nyelvben*

#### 1.4.1. Weblapok közvetlen konvertálásának lépései:

- weblap mentése
- weblap megnyitása a táblázatkezelő programban
- felesleges sorok és oszlopok törlése (további módszereket az 1.5. fejezetben mutatunk be)
- fájl mentése szövegfájlként
- szövegfájl bezárása,
- szövegfájl megnyitása
- fájl mentése a táblázatkezelő program saját formátumában.

A következőkben nézzük a fenti algoritmus részleteit.

##### 1.4.1.1. Weblap mentése

Weblapok mentése bármelyik böngészőben elvégezhető. Arra kell figyelni, hogy teljes mentést végezzünk, hogy a mentés során ne történjen adatvesztés. A 8. ábra és 9. ábra mintái mentett weblapok táblázatait és azok környezetét mutatják.

- Éremtáblázat<sup>7</sup>
- 2012 Formula 1 Drivers Championship Table<sup>8</sup>
- Várlexikon Vármegyék szerint<sup>9</sup>
- IMDb Charts: IMDb Top 250<sup>10</sup>
- Magyarország megyéi<sup>11</sup>

<sup>7</sup> Éremtáblázat. <http://olimpia2012.org/eremtablazat/>

<sup>8</sup> 2012 Formula 1 Drivers Championship Table. <http://www.f1-fansite.com/f1-results/2012-formula-1-drivers-championship-table/>

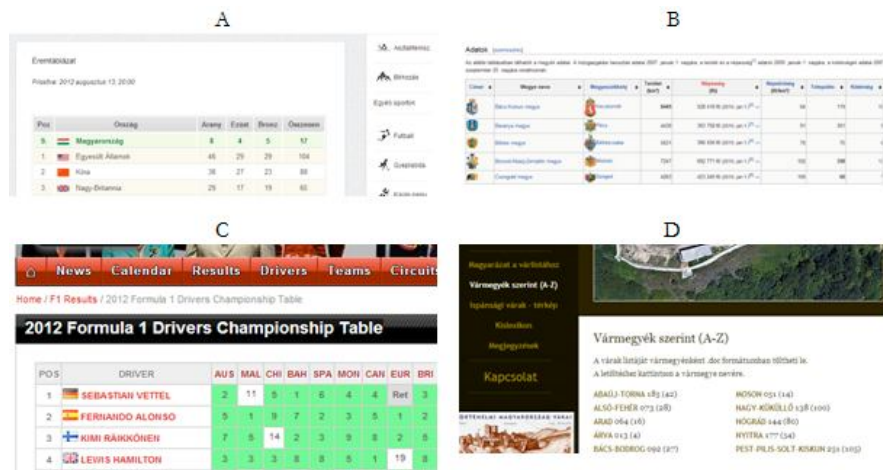
<sup>9</sup> Várlexikon Vármegyék szerint. [http://www.oroksegvedelem.hu/varmegyek\\_szerint\\_a\\_z/](http://www.oroksegvedelem.hu/varmegyek_szerint_a_z/)

<sup>10</sup> IMDb Charts: IMDb Top 250. <http://www.imdb.com/chart/top>

<sup>11</sup> Magyarország megyéi. [http://hu.wikipedia.org/wiki/Magyarorsz%C3%A1g\\_megy%C3%A9i](http://hu.wikipedia.org/wiki/Magyarorsz%C3%A1g_megy%C3%A9i)

Az **xxxxxxx** olyan weblapokat mutat, amelyek munkalappá konvertálása közvetlenül megtörténhet a környezettől függetlenül. Az *Éremtáblázat* és a *Formula1 helyezések* táblázata 1, 2 és 3-jegyű egész számokat tartalmaznak, ezek konvertálása problémamentes. Magyarország megyei weblap Népszerűség oszlopa ijesztőnek tűnhet, mert egyrészt túlságosan sok adatot tartalmaz, másrészt a számok 6-, illetve 7-jegyűek, amelyekben az ezreselválasztó karakter a szóköz. A konvertálás eredményeként azonban ez az oszlop szövegtípusú lesz, mivel a cellák a számok mellett szöveg is tartalmaznak, és így biztosan nem történnek nem kívánt automatikus típus felismerés és átalakítás. A szövegtípusú adatok számokká konvertálása több különböző módon is elvégezhető. Ezek leírását az 1.5. alfejezet tartalmazza. A Vármegyék weblapról csak a vármegyék nevére és az azokhoz tartozó számokra van szükségünk, különösebben nincs gond a konvertálásukkal.

A szövegfájl típusának a megválasztásánál érdemes odafigyelni, hogy magyar nyelvű szövegeknél a *Unicode-os szöveg (.txt)* típust válasszuk, hogy az ékezetes karakterek konvertálása is helyesen történjen meg.



8. ábra

Táblázatkezelő programokba közvetlenül konvertálható weblapokból részletek

Problémás viszont az IMDb lap konvertálása. A közvetlen konvertálás eredményét az 7. ábra bal oldali és a 9. ábra jobb oldali mintái mutatják.



Az eredeti weblap tizedes törtjeit, amelyekben az angol írásmódnak megfelelően tizedes pont szerepel, dátummá konvertálja a táblázatkezelő – Rating oszlop –, kivéve a 9.0 értéket (9. ábra, 3. film). A Votes oszlop számaiban szereplő ezreselválasztó vessző karakter teljesen összezavarja a programot. Három különböző eredményt kaphatunk.

- Két esetben egész,tört alakú számmá konvertálja az eredetileg egész számokat, amelyeknek további kétféle kimenete lehetséges:
- Ha a legkisebb helyiértéken nullától különböző szám található, akkor az eredeti érték ezred részét kapjuk. Ez a probléma a táblázatkezelőn belüli típus átalakításokkal és szorzással kezelhető.
- Az igazi problémát az jelenti, amikor számjegyesztés következik be. Ez akkor történhet, ha a legkisebb helyiértéken/helyiértékeken nulla található. Ebben az esetben az ezerrel való osztás után a program úgy ítéli meg, hogy a törtrészt záró nullák teljesen feleslegesek és eltávolítja, törli ezeket. A törlés következtében ez a folyamat visszafordíthatatlan, mivel nem tudjuk, hogy az eredeti szám legkisebb helyiértékein hány darab nulla volt, vagy volt-e egyáltalán. Az egyszerűbb esetekben az ezerrel történő szorzás segíthet. Jelen tanulmányban a teljesen általános megoldást mutatjuk be. A 9. ábra 2. filmjénél az ezerrel való osztás után levette a nullát a végéről. Így nem tudjuk, hogy az eredeti szám egy 5-jegyű szám volt-e vagy egy olyan 6-jegyű, aminek elvesztettük az utolsó jegyét. Hasonló módon tizedesjegy-vesztés történt, a 9. és 10. ábra 5. és 6. filmjeinél. A 69. film esetében a legkisebb helyiértékeken 3 darab 0 is található. A konvertálás következtében valamennyi törlésre kerül (9. ábra).
- Ha az eredeti szám kettő vagy attól több ezreselválasztó vesszőt tartalmaz, akkor a számokat szöveggé konvertálja a program (9. ábra 1. és 6. film).

Mivel a konvertálás eredménye a számok eredeti értékétől és formátumától függ, így az adatvesztés miatt a táblázatkezelő programon belül utólag már nem tudjuk az eredeti értéket visszaállítani. Az adatvesztést kell megakadályozni. Ahhoz, hogy az adatvesztést elkerüljük, szükség van egy köztes lépésre, amelyik előállítja a szövegfájlt. Ehhez néhány ötletet az **xxx fejezet** tartalmaz.



**Top 250 movies as voted by our users**  
 For this top 250, only votes from regular voters are considered.  
 Track which films you've seen from the Top 250 right here!  
 Take the IMDb Top 250 and Top 100 Quiz.

Rank	Rating	Title	Votes
1.	9.2	The Shawshank Redemption (1994)	1,030,432
2.	9.2	The Godfather (1972)	725,270
3.	9.0	The Godfather: Part II (1974)	469,752
4.	8.9	Pulp Fiction (1994)	797,814
5.	8.9	The Good, the Bad and the Ugly (1966)	309,808
6.	8.9	The Dark Knight (2008)	1,005,380
69.	8.4	Lawrence of Arabia (1962)	125,000

#	A	B	C	D
1	Rank	Rating	Title	Votes
2	1.	02.szept	The Shawshank Redemption (1994)	1,030,432
3	2.	02.szept	The Godfather (1972)	725,270
4	3.	9.0	The Godfather: Part II (1974)	469,752
5	4.	09.aug	Pulp Fiction (1994)	797,814
6	5.	09.aug	The Good, the Bad and the Ugly (1966)	309,808
7	6.	09.aug	The Dark Knight (2008)	1,005,380
8	7.	09.aug	12 Angry Men (1957)	252,769
9	8.	09.aug	Schindler's List (1993)	523,527
10	9.	08.aug	The Lord of the Rings: The Return of the King (2003)	731,072
11	10.	08.aug	Fight Club (1999)	782,181
12	11.	08.aug	Star Wars: Episode V - The Empire Strikes Back (1980)	498,931
67	69.	04.aug	The Prestige (2006)	477,440
68	67.	04.aug	A Clockwork Orange (1971)	343,947
69	68.	04.aug	Amélie (2001)	337,467
70	69.	04.aug	Lawrence of Arabia (1962)	125,000
71	70.	04.aug	To Kill a Mockingbird (1962)	127,100

**9. ábra**

*Az IMDb weblap (bal) és a konvertálás eredménye (jobb). A magyar nyelvű táblázatkezelő programok helytelenül konvertálják az olyan típusú adatokat, amelyekben a számok az angol írásmód szerinti tizedes pontot és ezreselválasztó vesszőt használnak (jobb oldali panel B és D oszlopa)*

**1.4.1.2. Weblap megnyitása a táblázatkezelő programban**

A weblapok táblázatkezelő programokban történő megnyitására a leggyorsabb és legbiztonságosabb módszer, ha azt a táblázatkezelő programon belül a Megnyitás paranccsal végezzük el.

Ennél a lépésnél a teljes weblap megnyitásra kerül. Minden, a weblapon található objektum bekerül a munkalapra, és a szöveges elemek átöröklik a weblapon kapott formátumukat. Első ránézésre egy teljes káosznak tűnik az egész, de ez azért van, mert a táblázatkezelő programok nem tudják olyan módon értelmezni a weblapokat, ahogy azt a böngészők teszik. Sok esetben úgy tűnik, hogy a táblázatot is sikerült elveszíteni. Ez azonban nem így van. A táblázat megjelenik a munkalapon, de nem feltétlenül a lelegején. Ilyenkor lentebb kell görgetni, hogy megtaláljuk a táblázatot.

**1.4.1.3. Felesleges sorok és oszlopok törlése**

A konvertálás következő lépése a táblázat előtti és utáni fölösleges sorok, valamint a táblázattól balra és jobbra elhelyezkedő fölösleges oszlopok

pok törlése. Sorok és oszlopok törlésénél figyelni kell, hogy ne csak azok tartalmát töröljük ki, hanem valóban teljes sorokat és oszlopokat.<sup>12</sup>

Ha lehetséges az eddigi káosz fokozása, akkor az most megtörtént. A weblap összes objektuma a táblázatra halmozódott. Ilyenkor egy dolgot nem szabad csinálni, kétségbeesetten kattintgatni, mivel az objektumokon rajta vannak a linkek, és minden egyes linkre kattintva egy weblap töltődik be.

Nem szabad azonban elkezdni egyesével törölni ezeket az objektumokat, leszedni a linkeket, mert mindkettő rendkívül időigényes folyamat. A szövegfájlá alakítással minden ilyen jellegű, a táblázat adatai szempontjából fölösleges objektumok és formázások eltávolíthatók (0. fejezet).

#### **1.4.1.4. Fájl mentése szövegfájlként**

A fölösleges sorok és oszlopok törlése után a fájlt szövegfájlként érdemes menteni, amelynek következtében csak a formázatlan karakter típusú adatok maradnak meg.

A szövegfájl létrehozása a *Mentés másként* paranccsal történik. A konverzió szempontjából a legfontosabb a fájl típusának a megválasztása, amit a *Fájl típusa* mezőhöz tartozó listából választhatunk ki. Választhatjuk a *Szöveg (tabulátorral tagolt) (.txt)*, és a *Unicode-os szöveg (\*.txt)* típusokat.<sup>13</sup> Magyar nyelvű szövegeknél érdemes az utóbbit választani, azonban a név megtévesztő lehet, mert nem derül ki egyértelműen, hogy ez a típus is tabulátorral tagolt szöveget hoz létre. A szokásos módon szükség van még az elérési útvonal és a fájlnev megadására. A fájlnevnél csak az első felét, a tényleges nevet kell megadni, mert a kiterjesztés a kiválasztott típus alapján automatikusan hozzáadódik a fájlnevhez. Ez az automatizmus vezet azonban sok esetben ahhoz a tévhithez, hogy a kiterjesztés és a fájltypus szinonimák.

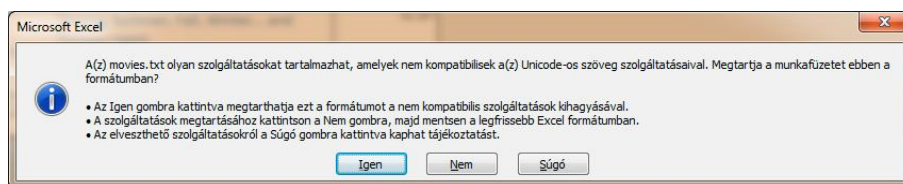
A mentés megkezdésekor megjelenik egy információs ablak, amely arra figyelmeztet, hogy a szöveges formátumban történő mentéskor információvesztés történhet, és rákérdez, hogy valóban ez volt-e az eredeti szándékunk. Igen, éppen azt akartuk, hogy a fölösleges, nem karakteres

---

<sup>12</sup> Sorok és oszlopok törléséhez jelöljük ki a sor és az oszlop fejek segítségével a teljes sorokat és oszlopokat. Ezt követően a helyi menü Törlés gombjával végezzük el a törlést. A Delete gomb csak tartalmat töröl és nem sorokat, oszlopokat.

<sup>13</sup> MS operációs rendszereket használva.

információk ne kerüljenek mentésre, tehát nyugodtan rá lehet kattintani az *Igen* gombra (10. ábra).

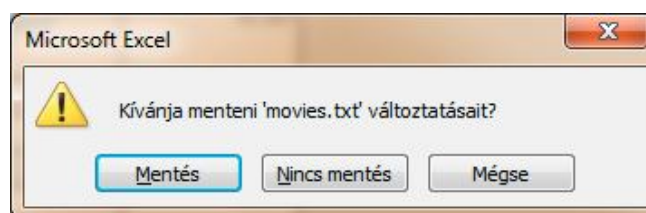


**10. ábra**

*A szövegfájlként történő mentéskor megjelenő információs ablak*

#### 1.4.1.5. Szövegfájl bezárása, majd újra megnyitása

A mentést követően bezárjuk a munkafüzetet,<sup>14</sup> majd ezután újra megnyitjuk. A dokumentum bezárásakor újra rákérdez a táblázatkezelő, hogy akarjuk-e menteni a változtatásokat (11. ábra). Ezt a kérdést akkor is felteszi a program, ha semmiféle felhasználói beavatkozás nem történt. Ha biztosan tudjuk, hogy nem módosítottuk a fájlt, akkor a *Nincs mentés* is választható, de egy újabb mentés is elvégezhető a *Mentés* paranccsal.



**11. ábra**

*A szövegfájl bezárásakor megjelenő figyelmeztető üzenet. Az üzenet akkor is megjelenik, ha semmiféle módosítás nem történt a mentés óta*

A fájlt azonban újra meg kell nyitni, mert még nem értük el a konverzió végcélját, a táblázatkezelő saját formátumával rendelkező formátumot. dokumentumot. A szövegfájlok megnyitása egy három lépésös fo-

<sup>14</sup> Ctrl+F4 MS operációs rendszerekben.

lyamat, amelyben beállíthatjuk a tagolás és a kódolás módját, az oszlopok közötti elhatároló jeleket, valamint az adatok típusát oszloponként.<sup>15</sup>

#### **1.4.1.6. Fájl mentése a táblázatkezelő program saját formátumában**

A következő lépés a megnyitott szövegfájl mentése a táblázatkezelő saját formátumában. Ezzel a mentéssel lesz a konverzió teljes. Az új fájlformátum megadását ismételten a *Mentés másként* paranccsal végezzük, kiválasztjuk a fájl típusát, ami most a táblázatkezelő program saját formátuma, rendszerint az első a listán, majd ezt követően megadjuk az elérési útvonalat és a fájlnevet. A kiterjesztést ebben az esetben sem szabad gépelni, az automatikusan hozzáadódik a fájl nevéhez.

#### **1.5. Gyakorlatok I: Weblapok többlépcsős konvertálása (27. óra)**

Ha a weblap-szövegfájl-munkafüzet hármas nem működik az adatok sajátos írásmódja és elrendezése miatt, akkor a szövegfájl előállítására egy köztes lépést kell beiktatni. Ez annyit jelent, hogy szükség van egy olyan programra, amelyik meg tudja nyitni a weblapot, de nem végzi el a karakteres adatok automatikus típus beállításait, ezzel egyidejűleg azok automatikus típus konverzióját. Annak eldöntésére, hogy táblázatkezelőben vagy szövegszerkesztőben nyitjuk meg a weblapot további segítséget jelent, hogy megnézzük, hogy a táblázatkezelő hány oszlopba és sorba rendezi az adatokat (89. ábra). Mindezeket figyelembe véve a szövegszerkesztő programok közbeiktatása egy lehetséges megoldás.

A szövegszerkesztő programon belüli konvertálás lépései a következők:

- weblap megnyitása,
- a táblázat előtt és után elhelyezkedő fölösleges sorok törlése,
- karakterek cseréje a szükséges oszlopokban,
- táblázat tabulátorokkal határolt szöveggé alakítása,
- szövegfájl mentése.

A Microsoft Word vagy az OpenOffice Writer programokkal elvégezhetőek a konvertáláshoz szükséges lépések. A weblap megnyitását a szövegszerkesztő programon belülről kell indítani, majd érdemes a fájlt azonnal menteni a szövegszerkesztő saját formátumában. A fájlok sokszínűsége, és egy-egy konverziós lépés bizonytalan kimenete mindenképpen

---

<sup>15</sup> A szövegfájl létrehozása történhet .csv formátumban is, de ekkor a megnyitásnál nincs lehetőségünk a konvertálás lépéseinek beállítására.

indokolja a többszörös mentést, tehát, hogy minden egyes komolyabb lépés után indexelve, új néven elmentsük a fájlt.

A megnyitást követően célszerű a *Webes elrendezés* nézetet átállítani *Nyomtatási elrendezés* nézetre, így a már megszokott szövegszerkesztő környezetben dolgozhatunk a továbbiakban.

### 1.5.1. IMDb

#### Fölösleges sorok törlése – IMDb

A szövegszerkesztőben megnyitott *IMDb* weblap sokkal kevésbé ijesztő, mint ahogy azt a táblázatkezelő esetében megtapasztaltuk. A táblázat előtt és után helyezkednek el a számunkra felesleges sorok, ezeket kell kitörölni úgy,<sup>16</sup> hogy végeredményként csak a táblázat maradjon, ami az *IMDb* dokumentum esetében egy 250 soros táblázat, minden egyes filmnek egy-egy külön sor, és 4 oszlop (6. ábra és 9. ábra).

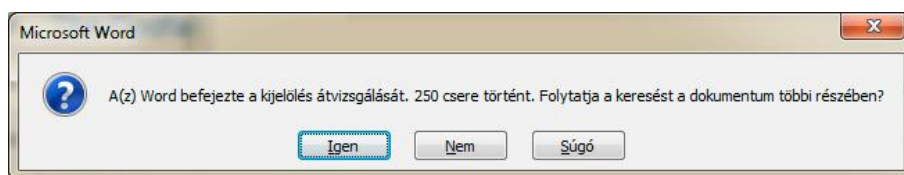
#### Karakterek cseréje a szükséges oszlopokban

Jelöljük ki azt az oszlopot, amelyben a karakterek cseréjét el akarjuk végezni. Adjuk ki a *Csere* parancsot, állítsuk be a régi és az új karaktereket a megfelelő mezőkben, majd indíthatjuk a cserét *Az összes cseréje* paranccsal. A szövegszerkesztő program elvégzi a kijelölt területen a cserét, majd megjelenik az elvégzett és a további cserékre vonatkozó információs tábla. A 12. ábra az *IMDb* lap cseréihez tartozó információs üzenetet mutatja. Az üzenet egy nagyon jó példa arra, hogy mennyire fontos a feldolgozás alatt álló lap adatainak ismerete, mivel ezzel is ellenőrizhető az eredeti lap szerkezete, a cserék elvégezhetősége. Ebben a példában az üzenetből kiolvasható, hogy 250 csere történt, ami azt mutatja, hogy minden film esetén pontosan egy. Tehát jól választottunk a cseréhez szükséges karaktereket és tartományt. Ennek megfelelően, arra a kérdésre, hogy „*Folytatja a keresést a dokumentum többi részében?*” egy

---

<sup>16</sup> A táblázat előtti és utáni szövegrész törlését kijelölésekkel érdemes elvégezni. A táblázat előtti szöveg kijelöléséhez állítsuk a kurzort közvetlenül a táblázat elé, majd végezzük el a kijelölést egyetlen billentyűkombinációval: Ctrl+Shift+Home. A táblázat utáni szövegrész esetében hasonlóan járunk el. Állítsuk a kurzort közvetlenül a táblázat után és jelöljük ki a szöveget a Ctrl+Shift+End billentyűkombinációval.

határozott *Nem* a válasz, mivel nem szabad megengedni, hogy a teljes dokumentumon elvégezze a megjelölt karakterek cseréjét (12. ábra).<sup>17</sup>



**12. ábra**

*Csere parancs információs ablaka*

Az *IMDb* táblázat cserék előtti és utáni állapotát mutatja a 13. ábra.

1.	9.2	The Shawshank Redemption (1994)	1,030,432	1	9.2	The Shawshank Redemption (1994)	1030432
2.	9.2	The Godfather (1972)	725,270	2	9.2	The Godfather (1972)	725270
3.	9.0	The Godfather: Part II (1974)	469,752	3	9.0	The Godfather: Part II (1974)	469752
4.	8.9	Pulp Fiction (1994)	797,814	4	8.9	Pulp Fiction (1994)	797814
5.	8.9	The Good, the Bad and the Ugly (1966)	309,808	5	8.9	The Good, the Bad and the Ugly (1966)	309808
6.	8.9	The Dark Knight (2008)	1,005,380	6	8.9	The Dark Knight (2008)	1005380
7.	8.9	12 Angry Men (1957)	252,769	7	8.9	12 Angry Men (1957)	252769

**13. ábra**

*Szövegszerkesztőben elvégzett karaktercserék, amelyek célja, hogy megakadályozzuk a nem kívánt típus konverziókat a táblázatkezelő programban*

#### A táblázat tabulátorokkal határolt szöveggé alakítása

A szövegszerkesztőben elvégzett konverzió végeredménye egy szövegfájl lesz, ezért a táblázatot át kell alakítanunk tabulátorokkal határolt szöveggé, mivel a szövegfájl nem támogatja a táblázatot. Az átalakítást a *Táblázatkezelők* → *Elrendezés* → *Szöveggé alakítás* paranccsal indíthatjuk el, cellahatárolónak pedig érdemes elfogadni a felkínált tabulátort.

A táblázat tabulátorokkal határolt szöveggé átalakítása ismételtelen egy első ránézésre rendezetlen szöveget eredményezhet. Ez azonban nem így van. Pontosán egy tabulátor választ el minden esetben két oszlopot, és ez a lényeg. A kuszaság a tabulátorhelyek automatikus meghatározásából származhat. A konvertálás menetét azonban nem zavarja a tabulátor elhelyezkedése, tehát nem érdemes azok módosításával foglalkozni.

<sup>17</sup> Az *IMDb* táblázat a *Rating* oszlopában a pontot kell vesszőre cserélni, míg a *Votes* oszlopában a vesszőt semmire. Nem feltétlenül szükséges, de a *Rank* oszlop pont karaktereit is le lehet cserélni semmire.

Egyetlen lépés maradt hátra, a dokumentum szövegfájlként mentése. Ezt a műveletet is a *Mentés másként* paranccsal indítjuk, majd a fájl típusok közül az *Egyszerű szöveg (.txt)* típust választjuk. Az így mentett szövegfájl a táblázatkezelő programokkal gond nélkül megnyitható.

### 1.5.2. Menetdíj

Már az eredeti weblapon böngészőben történő megnyitásánál is lehet sejteni, hogy valami nincs rendben a táblázat adataival, mivel a számok igazítása nem egységes. A weblap megnyitása a táblázatkezelő programban azonban egyértelművé teszi, hogy valami óriási barkácsolás történt. Mindezt figyelembe véve a táblázatot érdemes egy szövegszerkesztő programban megnyitni és megnézni, hogy mi lehet a hiba oka. A szövegszerkesztő programok ugyanis karakter szinten egyértelműen mutatják, hogy milyen barkácsolásokat végeztek a táblázatban. A Menetdíj táblázatban azt is egyértelművé válik, hogy milyen összevisszaságban használják a szerzők a normál és a nem-törhető szöközőket, ami azonnal magyarázatot ad a számok különböző igazítására a táblázatkezelő programban.

Belföldi utazás	<b>Tejjesárú menetdíjak egy útra</b>		Tarifák 2013. június 1-től
Csopontos utazás	A táblázatban megadott díjak az előzetes függvények segítségével számítottak.		Tejjesárú menetdíjak egy útra
<b>Árak és kedvezmények</b>			<a href="#">Tejjesárú menetdíjak menettérti útra</a>
Ráküret			<a href="#">20 %-os mérséklésű menetdíjak egy útra</a>
Kedvezmények			<a href="#">20 %-os mérséklésű menetdíjak menettérti útra</a>
Helyi díjszabás			<a href="#">33 %-os mérséklésű menetdíjak egy útra</a>
Regionális menetdíjak 2014. február 15-től			<a href="#">33 %-os mérséklésű menetdíjak menettérti útra</a>
Online jegyértékesítés			<a href="#">50 %-os mérséklésű menetdíjak egy útra</a>
Jegyértékesítő automaták kezelése			<a href="#">50 %-os mérséklésű menetdíjak menettérti útra</a>
Vasútállomások, jegypénztárak, menetjegytárolók kezelése			<a href="#">90 %-os mérséklésű menetdíjak egy útra</a>
Kerékpárszállítás			<a href="#">90 %-os mérséklésű menetdíjak menettérti útra</a>
Család a vonaton			<a href="#">Kerékpár viteldíj, előállat viteldíj</a>
WiFi a vonaton			
Vasúti menetrend letöltése			
Utazási információk			
Balatonnyvesi kisvasút			

Tejjesárú menetdíjak			
Egy útra			
	Rámely vonat		2-1
Km	1	koninkulállyban	koninkulállyban
	G r i n t b a n		
10	250	315	65
15	310	390	80
20	370	465	95
25	465	580	115
30	560	700	140
35	650	815	165
40	745	930	185
45	840	1 050	210
50	930	1 160	220

### 14. ábra

*Az eredeti Menetdíj táblázat böngészőben megnyitva. A számok össze-vissza igazítása már itt is sejteni, hogy valamiféle barkácsolás történt a lapon*

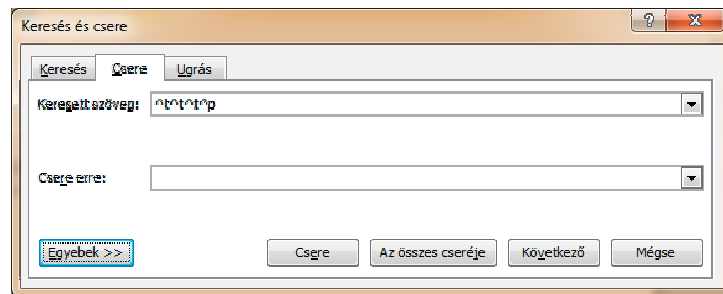




### Karakterek cseréje

A főlöszleges szóközök törlését cserével érdemes elvégezni. A szóközök semmire cseréje kitörli az összes szóközt, függetlenül attól, hogy az normál vagy nem-törhető szóköz volt.

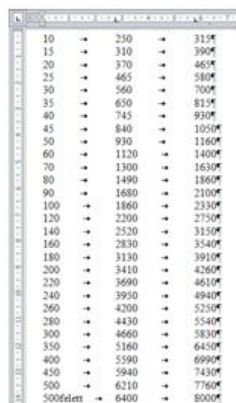
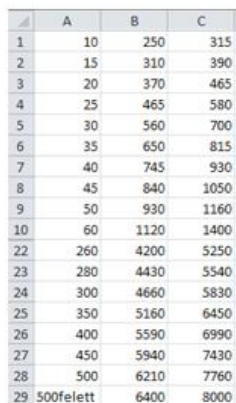
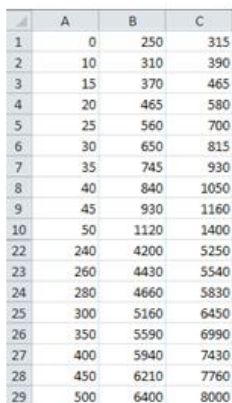
A táblázat üres sorait sem érdemes kézzel törölni. Sokkal gyorsabb és biztonságosabb, ha ezt a lépést is karaktercserével végezzük. Ehhez azonban érdemes a még mindig kusza táblázatot szöveggé alakítani, amelyben az oszlophatárolókat Tab karakterekkel helyettesítjük (*Táblázat* → *Elrendezés* → *Szöveggé alakítás* → *Tabulátor jel* (Táblázatból szöveg párbeszédablak)). Az így kapott szövegből már egyértelműen látszik, hogy az „üres” sorok egy-egy bekezdés három Tab karakterrel. Ezt négy karaktert érdemes a semmire cserélni. A Tab és a ¶ karakterek cseréje a karaktereknek megfelelő escape szekvenciákkal történik, amelyeket vagy gépelünk vagy beszúrunk a *Csere* ablak listájáról (*Kezdőlap* → *Szerkesztés* → *Csere* → *Egyebek* → *Speciális* → *Bekezdésjel és Tabulátorjel*).



**16. ábra**

*A Csere párbeszédablakon nem gépellhető karakterek escape szekvenciákkal helyettesítve – Tabulátor és Bekezdésjel, ^t és ^p*

A cserék után menthetjük a fájlt szövegfájlként, amely megnyitása semmiféle gondot nem fog jelenteni a táblázatkezelő programban. A további feldolgozáshoz érdemes még egy módosítást elvégezni a táblázaton: a menetdíjak úgy értendők, hogy a [0; 10] km-es intervallumon kell 250 Ft, illetve 315 Ft-ot fizetni, tehát a kilométerek és az árak egy sorral el vannak csúsztatva (17. ábra bal oldali és középső minta). Ezt egy egyszerű mozgatóval meg tudjuk oldani (17. ábra jobb oldali minta).

10	→	250	→	315*
15	→	310	→	390*
20	→	370	→	465*
25	→	465	→	580*
30	→	560	→	700*
35	→	650	→	815*
40	→	745	→	930*
45	→	840	→	1050*
50	→	930	→	1160*
60	→	1120	→	1400*
70	→	1300	→	1630*
80	→	1490	→	1860*
90	→	1680	→	2100*
100	→	1860	→	2330*
120	→	2200	→	2750*
140	→	2520	→	3150*
160	→	2830	→	3540*
180	→	3130	→	3910*
200	→	3410	→	4260*
220	→	3690	→	4610*
240	→	3950	→	4940*
260	→	4200	→	5250*
280	→	4430	→	5540*
300	→	4660	→	5830*
350	→	5160	→	6450*
400	→	5590	→	6990*
450	→	5940	→	7430*
500	→	6210	→	7760*
500felett	→	6400	→	8000*

17. ábra

A cserék és a táblázat szöveggé alakítása után létrehozott szövegfájl szövegszerkesztőben (bal) és táblázatkezelőben (középső) megnyitva, majd a módosított A oszloppal (jobb)

### 1.5.3. Megyék

A Magyarország megyéi weblapról a megyék adatait tartalmazó táblázatot fogjuk konvertálni oly módon, hogy az így létrehozott fájl táblázatkezelő programban megnyitható és feldolgozható legyen minden további konvertálás nélkül.

Címer	Megye neve	Megyeszékhely	Terület (km <sup>2</sup> )	Népesség (fő)	Népsűrűség (fő/km <sup>2</sup> )	Település	Kistérség	Járás
	Bács-Kiskun megye	Kecskemét	8445	524 841 fő (2011. jan 1.) <sup>[2]</sup>	64	119	10	11
	Baranya megye	Pécs	4430	391 455 fő (2011. jan 1.) <sup>[3]</sup>	91	301	9	10
	Békés megye	Békéscsaba	5631	361 802 fő (2011. jan 1.) <sup>[4]</sup>	70	75	8	9
	Borsod-Abaúj-Zemplén megye	Miskolc	7247	684 793 fő (2011. jan 1.) <sup>[5]</sup>	102	388	15	16
	Csongrád megye	Szeged	4263	421 827 fő (2011. jan 1.) <sup>[6]</sup>	100	60	7	7
	Fejér megye	Székesfehérvár	4359	426 120 fő (2011. jan 1.) <sup>[7]</sup>	97	108	10	9
	Győr-Moson-Sopron megye	Győr	4208	449 967 fő (2011. jan 1.) <sup>[8]</sup>	105	183	7	7
	Hajdú-Bihar megye	Debrecen	6211	539 674 fő (2011. jan 1.) <sup>[9]</sup>	89	82	9	10
	Heves megye	Eger	3637	307 985 fő (2011. jan 1.) <sup>[10]</sup>	89	121	7	7
	Jász-Nagykun-Szolnok megye	Szolnok	5582	386 752 fő (2011. jan 1.) <sup>[11]</sup>	74	78	7	9
	Komárom-Esztergom megye	Tatabánya	2285	311 411 fő (2011. jan 1.) <sup>[12]</sup>	139	76	7	6
	Nógrád megye	Salgótarján	2546	201 919 fő (2011. jan 1.) <sup>[13]</sup>	86	131	6	6
	Pest megye	Budapest	6393	1 237 561 fő (2011. jan 1.) <sup>[14]</sup>	176	187	16	18

18. ábra

Magyarország megyéi weblap megyékre vonatkozó adatai

Címer	Megye neve	Megyeszékhely	Terrület (km <sup>2</sup> )	Népesség (fő)	Népsűrűség (fő/km <sup>2</sup> )	Település	Kistérség	Járás
	Bács-Kiskun megye	Kecskemét	8445	524'841 fő (2011. Jan. 1.) <sup>18</sup>	64	119	10	11
	Borsod-Abaúj-Zemplén megye	Miskolc	7247	684'793 fő (2011. Jan. 1.) <sup>18</sup>	102	358	15	16
	Csongrád megye	Szeged	4263	421'827 fő (2011. Jan. 1.) <sup>18</sup>	100	60	7	7
	Fejér megye	Székesfehérvár	4359	426'120 fő (2011. Jan. 1.) <sup>18</sup>	97	108	10	9
	Győr-Ménfőcsanak megye	Győr	4208	449'967 fő (2011. Jan. 1.) <sup>18</sup>	105	183	7	7
	Hajdú-Bihar megye	Debrecen	6211	539'674 fő (2011. Jan. 1.) <sup>18</sup>	89	82	9	10
	Heves megye	Egér	3637	307'985 fő (2011. Jan. 1.) <sup>18</sup>	89	121	7	7
	Jász-Nagykun-Szolnok megye	Szolnok	5582	386'752 fő (2011. Jan. 1.) <sup>18</sup>	74	78	7	9
	Komárom-Esztergom megye	Tatabánya	2265	311'111 fő (2011. Jan. 1.) <sup>18</sup>	139	76	7	6
	Nórád megye	Salgótarján	2546	201'919 fő (2011. Jan. 1.) <sup>18</sup>	86	131	6	6
	Pécs megye	Dudapest	6393	1'237'561 fő (2011. Jan. 1.) <sup>18</sup>	176	187	16	18

19. ábra

A megyék adatait tartalmazó táblázat szövegszerkesztőben

A szövegszerkesztőben megnyitott táblázat egyértelműen mutatja, hogy egy jól szerkesztett táblázatot tartalmazott a weblap, tehát várhatóan könnyű lesz algoritmust írni a konvertáláshoz. A táblázat előtti és utáni szövegrészek törlése.

- Címerek oszlop törlése.
- A népesség oszlop adatainak szétválasztása három oszlopra: népesség, dátum, egyéb adatok.
- A népesség és a dátum adatokat megtartjuk.
- A népesség számaiból érdemes törölni a szóközőket.
- A dátumban érdemes a nem-törhető szóközőket normál szóközre cserélni.
- Az egyéb adatok törölhetők.
- A redundáns „megye” szó eltávolítása a Megye neve oszlopból.
- Sortörések eltávolítása.

A táblázat előtti és utáni szöveg törlése egy-egy kijelöléssel<sup>18</sup> és törléssel gyorsan elvégezhető. A táblázat egy teljes oszlopának törléséhez az Oszlop törlése parancsot használjuk (Táblázateszközök → Elrendezés → Törlés → Oszlopok törlése vagy helyi menü → Oszlopok törlése).

<sup>18</sup> A kijelöléshez érdemes a Ctrl + Shift + Home és a Ctrl + Shift + End billentyűkombinációkat használni.

A sortörések eltávolítását érdemes cserével végezni, ahol a sortöréseket szóközre cseréljük.

A népesség oszlop adatait alapján három oszlopot fogunk létrehozni. Ahhoz, hogy új oszlopokat tudjunk létrehozni oszlophatárolókat kell elhelyeznünk az adatok közé a megfelelő helyekre. A három oszlop létrehozásához két oszlophatárolóra lesz szükségünk (2. táblázat). Az oszlophatárolók beillesztésekor érdemes már arra is gondolni, hogy egy úttal a felesleges adatokat is tudjuk törölni. Mindezen szempontokat figyelembe véve érdemes oszlophatárolónak választani a „□fő□(” és a „,)” karaktersorozatokat választani és ezeket Tab karakterre, illetve „.Tab” sztringre cserélni (2. táblázat). Azért célszerű Tab karaktert választani helyettesítő karakterként, mert a táblázat szöveggé konvertálásánál Tab karakterek lesznek az alapértelmezett oszlophatárolók. A második cserénél nem elég a ) karaktert lecserélni, mert a fejléc is tartalmaz ilyen karaktereket. Egy másik lehetséges megoldás a [ sztring cseréje lenne a Tab karakterre, de ez a megoldás sajnos nem működik, mivel a [ egy hiperhivatkozásnak a része.

Keresett szöveg: □fő□(		Keresett szöveg: .)	
Csere erre: ^t		Csere erre: .^t	
Cserék száma: 19		Cserék száma: 19	
Ered- mény:		Ered- mény:	

## 2. táblázat

*Új oszlopok létrehozásához szükséges cserék  
a Magyarország megyei fájlban*

A cserék elvégzése után érdemes az oszlopokat szétválasztani. Ezt a legegyszerűbben táblázat szöveggé alakításával (*Táblázateszközök* → *Elrendezés* → *Szöveggé alakítás*) (20. ábra), majd ezt követően vissza, a szöveg táblázattá alakításával (szöveg kijelölése → *Beszúrás* → *Táblázat* → *Szövegből táblázat*) (21. ábra) végezhetjük el. Az első konvertálási lépés után egy óriási káoszunk tűnik az egész dokumentum, pedig nem az (20. ábra minta), csak összezsúsztak az oszlopok a tabulátor pozíciók automatikus elhelyezéséből és a papírméretből adódóan. Nem szabad azonban elfelejtenünk, hogy a fejlécben a Népességtől jobbra be kell illesztenünk két Tab karaktert, amelyek az újonnan létrehozott oszlopok fejlécét fogják kijelölni (20. ábra alsó minta).

**20. ábra**  
 Táblázat szöveggé alakítása után a Tab karakterekkel oszlopokra tördelt szöveg

Megye neve	Megveszékheh	Terület (km <sup>2</sup> )	Népség (fő)	Népsűrűség (fő/km <sup>2</sup> )	Település	Kistérség	Járás
Bács-Kiskun megye		8445	524841	2011. jan. 1.	64	119	10

**21. ábra**  
 Tab karakterekkel tagolt szöveg visszaalakítása táblázattá

Kitöröljük a fölösleges oszlopot a Népsűrűségtől jobbra, lecseréljük a Népség oszlop nem-törhető szóközeit semmire (20 cseré lesz Pest megye hétjegyű száma miatt), és lecseréljük a Dátum oszlop nem-törhető szóközeit normál szóközeire (38 cseré).

Megye neve	Megveszékheh	Terület (km <sup>2</sup> )	Népség (fő)	Dátum	Népsűrűség (fő/km <sup>2</sup> )	Település	Kistérség	Járás
Bács-Kiskun megye		8445	524841	2011. jan. 1.	64	119	10	11

**22. ábra**  
 Nem-törhető szóközök cseréje a Népség és a Dátum oszlopokban

	A	B	C	D	E	F	G	H	I
1	Megye neve	Megyeszékhely	Terület (km <sup>2</sup> )	Népesség (fő)	Dátum	Népsűrűség (fő/km <sup>2</sup> )	Település	Kistérség	Járás
2	Bács-Kiskun	Kecskemét	8445	524841	2011. jan 1.	64	119	10	11
3	Baranya	Pécs	4430	391455	2011. jan 1.	91	301	9	10
4	Dékes	Dékáncsaba	5631	361802	2011. jan 1.	70	75	8	9
5	Borsod-Abaúj-Zemplén	Miskolc	7247	684793	2011. jan 1.	102	358	15	16
6	Csongrád	Szeged	4263	421827	2011. jan 1.	100	60	7	7
7	Fejér	Székesfehérvár	4359	426120	2011. jan 1.	97	108	10	9
8	Győr-Moson-Sopron	Győr	4208	449967	2011. jan 1.	105	183	7	7
9	Hajdú-Bihar	Debrecen	6211	539674	2011. jan 1.	89	82	9	10
10	Heves	Eger	3637	307985	2011. jan 1.	89	121	7	7
11	Jász-Nagykun-Szolnok	Szolnok	5582	386752	2011. jan 1.	74	78	7	9
12	Komárom-Esztergom	Tatabánya	2265	311411	2011. jan 1.	139	76	7	6
13	Nógrád	Salgótarján	2516	201919	2011. jan 1.	86	131	6	6
14	Pest	Budapest	6393	1237561	2011. jan 1.	176	187	16	18
15	Somogy	Kaposvár	6036	317947	2011. jan 1.	55	245	11	8
16	Szabolcs-Szatmár-Bereg	Nyíregyháza	5936	552000	2012. jan 1.	98	229	12	13
17	Tolna	Szekszárd	3703	231183	2011. jan 1.	67	109	5	6
18	Vas	Szombathely	3336	257688	2011. jan 1.	80	216	9	7
19	Veszprém	Veszprém	4493	353068	2011. nov 1.	82	217	9	10
20	Zala	Zalaegerszeg	3784	287043	2011. jan 1.	78	257	9	6

### 23. ábra

*A táblázatkezelőben megnyitott szövegfájl*

A szövegfájl létrehozásához a táblázatot újra szöveggé kell alakítani, és érdemes Tab karaktert használni oszlophatárolóként. A végeredményt a 23. ábra mutatja.

#### 1.5.4. Könyvek – Best Books Ever

A goodreads<sup>19</sup> weblapról elérhető könyvlisták konvertálása olyan szöveg fájlra, amelyet jól tudunk a táblázatkezelő programokban használni egy több lépésből álló algoritmus. Az algoritmus különböző típusú cseréket, táblázat-szöveg konverziókat és különböző típusú mentéseket tartalmaz, az előző fejezetekben ismertetett műveletek kombinációja.

<sup>19</sup> goodreads. <https://www.goodreads.com/>. Letöltve. 2014. augusztus 6.

	all votes	add books to this list
1	 <p><b>The Hunger Games (The Hunger Games, #1)</b> by Suzanne Collins ★★★★★ 4.41 avg rating — 2,767,441 ratings score: 1,990,193, and 20,195 people voted</p>	<p>Want to Read </p> <p>Rate this book ★★★★★</p>
2	 <p><b>Harry Potter and the Order of the Phoenix (Harry Potter, #5)</b> by J.K. Rowling ★★★★★ 4.40 avg rating — 1,199,753 ratings score: 1,436,391, and 14,769 people voted</p>	<p>Want to Read </p> <p>Rate this book ★★★★★</p>
3	 <p><b>Harry Potter and the Goblet of Fire (Harry Potter, #4)</b> by J.K. Rowling ★★★★★ 4.47 avg rating — 1,212,651 ratings score: 1,320,552, and 13,702 people voted</p>	<p>Want to Read </p> <p>Rate this book ★★★★★</p>
4	 <p><b>Harry Potter and the Prisoner of Azkaban (Harry Potter, #3)</b> by J.K. Rowling ★★★★★ 4.46 avg rating — 1,251,612 ratings score: 1,314,898, and 13,717 people voted</p>	<p>Want to Read </p> <p>Rate this book ★★★★★</p>

## 24. ábra

*A Best Books Ever weblap első néhány könyve*

A Best Books Ever<sup>20</sup> weblap első ránézésre nem tűnik túlságosan barátságosnak, mivel egyetlen oszlopban van, minden a könyvekre és a szavazatokra vonatkozó adat. Táblázatkezelőben megnyitva minden félelünk beigazolódni látszik; a könyvek és a szavazatok adatai teljesen széttörve jelennek meg, tíz táblázatsorba rendezve. A weblap táblázatkezelőben, függvényekkel történő konvertálását a 3.3.2. fejezet tartalmazza. Érdekes azonban megpróbálni a konvertálást egy szövegszerkesztőben. A fájl megnyitása szövegszerkesztőben olyan sokáig tart, hogy ez még inkább elbizonytalaníthat bennünket, de a helyzet nem ilyen rossz. Ez egy barátságos fájl, csak ki kell törölni a szemetet a táblázat előtt, után és mellett.

A felesleges sorok és oszlopok törlése után látszik, hogy valóban egyetlen oszlopban van minden adat. Ez az oszlop viszont egységes, tehát jól konvertálható.

<sup>20</sup> goodreads. Best Books Ever. [https://www.goodreads.com/list/show/1.Best\\_Books\\_Ever](https://www.goodreads.com/list/show/1.Best_Books_Ever). Letöltve: 2014. augusztus 6.





25. ábra

*A Best Books Ever* weblap szövegszerkesztőben megnyitva és a fölösleges sorok és oszlopok kitörölve

A 25. ábra mintája alapján ki tudjuk válogatni, hogy melyek azok az adatok, amelyeket meg kell tartani, külön oszlopokba elhelyezni és melyeket kell kitörölni (3. táblázat).

	megtartani		törölni
1.	cím (The Hunger Games (The Hunger Games, #1))	2.	a <i>rating</i> kétszer szerepel
3.	szerező (Suzanne Collins)		
4.	<i>rating</i> tizedes tört (4.41)	5.	az első <i>rating</i> és a <i>ratings</i> egész szám között mindent
6.	<i>ratings</i> egész szám (2,776,080)	7.	<i>ratings</i> sztring
8.	<i>score</i> egész szám (1,984,532)	9.	<i>score</i> sztring
10.	<i>people voted</i> egész szám (20,239)	11.	<i>and</i> szting
		12.	<i>people voted</i> sztring

3. táblázat

*A Best Books Ever* táblázat konvertálásához szükséges feltételek



A konvertálás algoritmusánál érdemes mérlegelni, hogy melyek azok a műveletek, amelyekkel új oszlopokat fogunk létrehozni és melyek azok, amelyekkel egy oszlopon belüli adatmódosítást végzünk (4. táblázat). Ennek eldöntése azért fontos, mert az oszlopok létrehozási szerkezeti kérdés, ami nagyban befolyásolja a további adatkezelés menetét, míg az oszlopon belüli konverziók csak a fölösleges adatok és az adatkezelés módja miatt érdekesek. Ezen utóbbi szempontot azonban csak akkor kell szem előtt tartani, ha táblázatkezelő beállításai indokolják, tehát ezek elvégzése nem is feltétlenül szükséges.

### Algoritmus

- hat oszlopos táblázat létrehozása
- fölösleges karaktere, sztringek törlése
- számok beállításokhoz igazítása

### Hat oszlopos táblázat létrehozása

Új oszlopok kialakításához szükséges algoritmust a következő felsorolás adja, míg a végrehajtott cserék részleteit a 4. táblázat mutatja.

- a legszembetűnőbb hiba a táblázatban, hogy kétszer tartalmazza a rating értéket, az első számtól jobbra érdemes törni az egységes, fölösleges adatokkal (4. táblázat 0. csere)
- a ratings egész számok leválasztására használható a számtól közvetlenül balra elhelyezkedő egységes — karakter (4. táblázat 0. csere)
- ezzel és az előző cserével létrehoztunk három oszlopot, ahol a középső teljesen felesleges és később törölhető lesz
- az and sztring cserével külön oszlopba tudjuk tenni a score és a people voted egész számokat (4. táblázat 0. csere)
- a sortörés és a by sztring helyettesítése szétválasztja a címet és a szerzőt (4. táblázat 0. csere)
- a szóköz-sortörés cserével a szerzőt és a rating tizedes törtet tudjuk szétválasztani (4. táblázat 0. csere)
- a bekezdés vége jel cseréjével szétválasztjuk a ratings egész számot a score egész számtól (4. táblázat 0. csere)

Ezzel a hat cserével létrehozunk hét oszlopot.

A 4. táblázat 0–0. cseréi már nem szerkezeti cserék, csak azt a célt szolgálják, hogy a megmaradt fölösleges karaktereket és sztringeket kitöröljük a számok mellől.

	Keresett szöveg	Csere erre	Cserék száma	Konvertálás típusa
1.	□of□5□stars□	^t	100	új oszlop
2.	□—□	^t	100	új oszlop
3.	,□and□	^t	100	új oszlop
4.	^lby□	^t	100	új oszlop
5.	□^l	^t	100	új oszlop
6.	□ratings□^p	^t	100	új oszlop
7.				
8.	score:□		100	törlés
9.	□people voted		100	törlés

#### 4. táblázat

*Új oszlopok létrehozásához szükséges cserék  
a Magyarország megyei fájlban*

Ahhoz, hogy egyrészt áttekinthetőbb legyen a táblázat, másrészt, hogy a fölösleges oszlopot ki tudjuk törölni, érdemes az egy oszlopos táblázatot szöveggé alakítani (*Táblázateszközök* → *Elrendezés* → *Szöveggé alakítás* → *Cellahatároló: Tabulátor*). Ezt követően, az így kapott tabulátorral határolt szöveget újra táblázattá alakítani (kijelölés → *Beszúrás* → *Táblázat* → *Szövegből táblázat*). Az így létrehozott táblázatnak már hét oszlopa lesz (26. ábra). Ennek a táblázatnak a 4. oszlopa fölösleges, amit egy oszloptörléssel egyetlen lépésben ki tudunk törölni (27. ábra).

<a href="#">The Hunger Games (The Hunger Games, #1)</a>	<a href="#">Suzanne Collins</a>	4.41	-4.41 avg-rating	2,776,080	<a href="#">1,984,532</a>	<a href="#">20,239</a>
<a href="#">Harry Potter and the Order of the Phoenix (Harry Potter, #5)</a>	<a href="#">J.K. Rowling</a>	4.4	-4.40 avg-rating	1,203,927	<a href="#">1,439,237</a>	<a href="#">14,798</a>
<a href="#">Harry Potter and the Goblet of Fire (Harry Potter, #4)</a>	<a href="#">J.K. Rowling</a>	4.47	-4.47 avg-rating	1,214,573	<a href="#">1,323,293</a>	<a href="#">13,730</a>

#### 26. ábra

*Az egy oszlopos táblázat szöveggé alakítása, majd ismételt táblázattá alakítása után kapott hét oszlopos táblázat*

<a href="#">The Hunger Games (The Hunger Games, #1)</a>	<a href="#">Suzanne Collins</a>	4.41	2,776,080	<a href="#">1,984,532</a>	<a href="#">20,239</a>
<a href="#">Harry Potter and the Order of the Phoenix (Harry Potter, #5)</a>	<a href="#">J.K. Rowling</a>	4.4	1,203,927	<a href="#">1,439,237</a>	<a href="#">14,798</a>
<a href="#">Harry Potter and the Goblet of Fire (Harry Potter, #4)</a>	<a href="#">J.K. Rowling</a>	4.47	1,214,573	<a href="#">1,323,293</a>	<a href="#">13,730</a>

### 27. ábra

A fölösleges oszlop törlése után kapott hat oszlopos táblázat

#### Fölösleges karakterek, sztringek törlése

A táblázat az átalakítás után még tartalmaz néhány fölösleges karaktert és sztringet, amelyeket az előző átalakításokkal nem sikerült törölni. Ez leginkább azzal magyarázható, hogy ezek a karakterek a hiperhivatkozásban vannak, és mint ilyen nem törölhető a hivatkozáson kívül karakterekkel (27. ábra).

Ennél a pontnál tehát érdemes megválni a hiperhivatkozásoktól és minden egyéb felesleges objektumtól, ami még a táblázatban maradt. Ilyen például az utolsó oszlop napocskája. Ezt legegyszerűbben a fájl szövegfájllá alakításával tudjuk megtenni. Nem szabad azonban elfelejteni, hogy a szövegfájl létrehozását meg kell előznie a hat oszlopos táblázat tabulátorral határolt szöveggé alakítása. Ennek a műveletnek a végeredménye a szokásos káosz, ami egyébként nagyon barátságos (28. ábra).

```

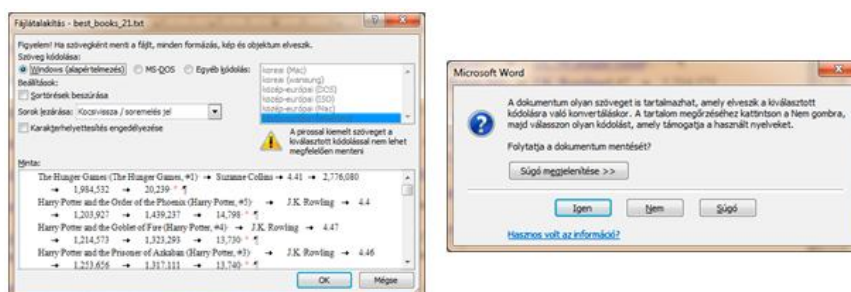
The Hunger Games (The Hunger Games, #1) → Suzanne Collins → 4.41
→ 2,776,080 → 1,984,532 → 20,239
Harry Potter and the Order of the Phoenix (Harry Potter, #5) → J.K. Rowling → 4.4
→ 1,203,927 → 1,439,237 → 14,798
Harry Potter and the Goblet of Fire (Harry Potter, #4) → J.K. Rowling 4.47 → 1,214,573
→ 1,323,293 → 13,730
Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) → J.K. Rowling → 4.46
→ 1,253,656 → 1,317,111 → 13,740

```

### 28. ábra

A fölösleges oszlop törlése után kapott hat oszlopos táblázat

A szöveget ekkor érdemes szövegfájlá alakítani, hogy minden formázást törölni tudjunk (*Fájl* → *Mentés másként* → *Fájl típusa: Egyszerű szöveg*). A tájékoztató és figyelmeztető üzenetek (29. ábra) csak annyit mondanak, hogy a táblázat tartalmaz nem szövegfájl-kompatibilis karaktereket és, hogy minden formázást el fogunk veszíteni a mentéssel. Ez nagyon jó hír, éppen ezt akartuk.



**29. ábra**

*Szövegfájl mentésekor megjelenő tájékoztató és figyelmeztető üzenetek*

A szövegfájl visszanyitása után csak a tabulátorral elválasztott oszlopok maradtak és a bekezdés határok. Ebben a szövegben tisztán látszik, hogy a cím után van egy fölösleges szóköz és az utolsó oszlopban □?□ karakterek (30. ábra).

```
The Hunger Games (The Hunger Games, #1) → Suzanne Collins → 4.41
→ 2,776,080 → 1,984,532 → 20,239 ·?·¶
Harry Potter and the Order of the Phoenix (Harry Potter, #5) → J.K. Rowling
→ 4.4 → 1,203,927 → 1,439,237 → 14,798 ·?·¶
Harry Potter and the Goblet of Fire (Harry Potter, #4) → J.K. Rowling
→ 4.47 → 1,214,573 → 1,323,293 → 13,730 ·?·¶
Harry Potter and the Prisoner of Azkaban (Harry Potter, #3) → J.K. Rowling
→ 4.46 → 1,253,656 → 1,317,111 → 13,740 ·?·¶
Twilight (Twilight, #1) → Stephenie Meyer → 3.56 → 2,449,313 → 1,200,604
→ 12,213 ·?·¶
```

### 30. ábra

*A hat oszlopos tabulátorokkal tagolt oszlopok, az első és utolsó oszlopokban a fölösleges karakterekkel*

A címet követő szóközt egymagában nem lehet cserélni, mivel a szöveg rengeteg egyéb szóközt is tartalmaz. A szóköz és az utána következő tabulátor azonban egyedi, tehát ezt a sztringet lecserélve egy tabulátor karakterre ki tudjuk törölni a szóközt is és az oszlophatároló is megmarad (5. táblázat).

Az utolsó oszlop egymást követő három karakterének törlése egyszerű: a sztring semmire cserélése (5. táblázat) (31. ábra).

Keresett szöveg	Csere erre	Cserék száma
□→	→	100
□?□		100

### 5. táblázat

*A fölösleges karakterek törlése*

Ha nincs szükség további nyelvi beállításokra, akkor menthető a szöveg-fájl, majd megnyitható táblázatkezelőben.

```

The ·Hunger ·Games ·(The ·Hunger ·Games, ·#1)·Suzanne ·Collins→4.41→2,776,080
→ 1,984,532→20,239¶
Harry ·Potter ·and ·the ·Order ·of ·the ·Phoenix ·(Harry ·Potter, ·#5)→J.K. ·
Rowling → 4.4 → 1,203,927→1,439,237→14,798¶
Harry ·Potter ·and ·the ·Goblet ·of ·Fire ·(Harry ·Potter, ·#4)→J.K. ·Rowling
→ 4.47→1,214,573→1,323,293→13,730¶
Harry ·Potter ·and ·the ·Prisoner ·of ·Azkaban ·(Harry ·Potter, ·#3)→J.K. ·
Rowling → 4.46→1,253,656→1,317,111→13,740¶
Twilight ·(Twilight, ·#1) → Stephenie ·Meyer→3.56→2,449,313→1,200,604
→ 12,213¶

```

### 31. ábra

*A hat oszlopos tabulátorokkal tagolt oszlopok,  
a fölösleges karakterek törlése után*

### Számok és a nyelvi beállítások egységesítése

A további feldolgozás nyelvétől függően lehet a tizedes és az ezres-  
elválasztó karaktert meghagyni vagy módosítani. Ebben az esetben vi-  
szont a tabulátorral határolt szöveget újra táblázattá kell alakítani, hogy  
oszloponként tudjuk a karakterek cseréjét elvégezni. Ennek következmé-  
nye, hogy a mentésnél a fájl típusát ismét a szövegszerkesztő program  
alapértelmezett típusára kell állítani.

The ·Hunger ·Games ·(The ·Hunger ·Games, ·#1) ¶	Suzanne ·Collins¶	4.41¶	2,776,080¶	1,984,532¶	20,239¶
Harry ·Potter ·and ·the ·Order ·of ·the ·Phoenix ·(Harry ·Potter, ·#5) ¶	J.K. ·Rowling¶	4.4¶	1,203,927¶	1,439,237¶	14,798¶
Harry ·Potter ·and ·the ·Goblet ·of ·Fire ·(Harry ·Potter, ·#4) ¶	J.K. ·Rowling¶	4.47¶	1,214,573¶	1,323,293¶	13,730¶
Harry ·Potter ·and ·the ·Prisoner ·of ·Azkaban ·(Harry ·Potter, ·#3) ¶	J.K. ·Rowling¶	4.46¶	1,253,656¶	1,317,111¶	13,740¶

### 32. ábra

*A hat oszlopos táblázat az angol tizedes ponttal  
és az ezreselválasztó vesszővel*

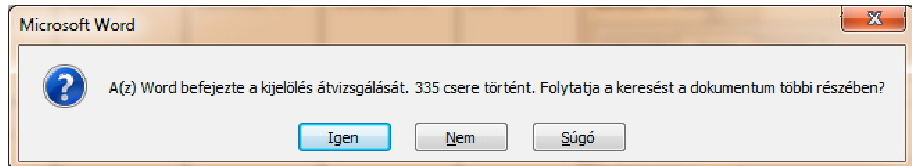
A tizedes és ezreselválasztó karakterek cseréjénél figyelni kell arra,  
hogy nem az egész dokumentumon kell elvégezni a cserét, hanem csak a  
kijelölt területen. Ez indokolta, hogy ismételten visszaállítsuk az oszlopok

elrendezést.<sup>21</sup> A 33. ábra üzenetére ebben az esetben, szemben az alapértelmezett *Igen*-nel, *Nem* a válasz. Hasonlóan járunk el a 3. oszlop karaktereinek a cseréjénél is. A végeredményt a 34. ábra mutatja.

Keresett szöveg	Csere helye	Csere erre	Cserék száma
,	utolsó három oszlop		335
.	3. oszlop	,	100

**6. táblázat**

*A feldolgozás nyelvétől függő karaktercserék*



**33. ábra**

*A hat oszlopos táblátorokkal tagolt oszlopok, a fölösleges karakterek törlése után*

The Hunger Games (The Hunger Games, -#1) ✕	Suzanne Collins ✕	4,41 ✕	2776080 ✕	1984532 ✕	20239 ✕
Harry Potter and the Order of the Phoenix (Harry Potter, -#5) ✕	J.K. Rowling ✕	4,4 ✕	1203927 ✕	1439237 ✕	14798 ✕
Harry Potter and the Goblet of Fire (Harry Potter, -#4) ✕	J.K. Rowling ✕	4,47 ✕	1214573 ✕	1323293 ✕	13730 ✕
Harry Potter and the Prisoner of Azkaban (Harry Potter, -#3) ✕	J.K. Rowling ✕	4,46 ✕	1253656 ✕	1317111 ✕	13740 ✕

**34. ábra**

*A tizedesjel és az ezreselválasztó karakterek cseréje a magyar nyelvű beállításoknak megfelelően*

<sup>21</sup> A táblátorral tagolt szövegnél is meg lehet csinálni az oszlopos kijelölt, ha az Alt segédbillentyűt használjuk, de ehhez először tisztán szét kell választani az oszlopokat. Nem feltétlenül éri meg.

A konvertálás legutolsó lépéseként a táblázatot ismételtlen tabulátorokkal elválasztott szöveggé kell alakítani, majd szövegfájlként menteni. A szövegfájl táblázatkezelőben megnyitva a 35. ábra mutatja.

	A	B	C	D	E	F
1	The Hunger Games (The Hunger Games, #1)	Suzanne Collins	4,41	2776080	1984532	20239
2	Harry Potter and the Order of the Phoenix (Harry Potter, #5)	J.K. Rowling	4,4	1203927	1439237	14738
3	Harry Potter and the Goblet of Fire (Harry Potter, #4)	J.K. Rowling	4,47	1214573	1323293	13730
4	Harry Potter and the Prisoner of Azkaban (Harry Potter, #3)	J.K. Rowling	4,46	1253656	1317111	13740
5	Twilight (Twilight, #1)	Stephenie Meyer	3,56	2449213	1290604	12213
6	Harry Potter and the Half-Blood Prince (Harry Potter, #6)	J.K. Rowling	4,48	1167696	1191292	12548
7	To Kill a Mockingbird	Harper Lee	4,23	2016224	1137846	11823
8	Harry Potter and the Deathly Hallows (Harry Potter #7)	J.K. Rowling	4,57	1207646	1009831	10712
9	Harry Potter and the Chamber of Secrets (Harry Potter #2)	J.K. Rowling	4,29	1208910	917609	9878
10	Pride and Prejudice	Jane Austen	4,24	1331619	913228	9552
11	Catching Fire (The Hunger Games, #2)	Suzanne Collins	4,31	1379774	820282	8704

### 35. ábra

*A Best Books Ever hat oszlopos táblázat táblázatkezelőben megnyitva*



## 2. Szöveges dokumentumok hibatípusai (17–20. óra)

### 2.1. Szövegkezelés (17. óra)

#### 2.1.1. Mit csináltunk rosszul?

A szövegszerkesztés oktatás valahol, valamikor nagyon rossz irányba tévedt. Nehéz megmondani, hogy mi történt, mivel számtalan összetevője lehet annak, hogy a ma létező szöveges e-dokumentumok szinte kivétel nélkül hibásak. Az egyik lehetséges magyarázat mindenképpen a szövegszerkesztés hibás módszertani megközelítése. A kerettantervi megfogalmazások, sajnálatos módon igazolják ezen feltételezésünket (Kerettanterv: 2008, Kerettanterv: 2013). Még tovább rontják a helyzetet azok a szélsőséges vélemények, amelyekben az fogalmazódik meg, hogy a szöveg- és táblázatkezelő programok az okai az informatikaoktatás eredménytelenségének (Gove: 2012, Gove: 2014).

*„Imagine the dramatic change which could be possible in just a few years, once we remove the roadblock of the existing ICT curriculum. Instead of children bored out of their minds being taught how to use Word and Excel by bored teachers, we could have 11 year-olds able to write simple 2D computer animations using an MIT tool called Scratch. By 16, they could have an understanding of formal logic previously covered only in University courses and be writing their own Apps for smartphones.”<sup>1</sup>*

Véleményem szerint Gove több helyen is tévedett:

Nem Wordöt és Excelt kell tanítani, hanem koncepció és algoritmus alapú szöveg- és táblázatkezelést (Csernoch–Bujdosó: 2014), és erre nagyon nagy szükség van mind a digitális írástudás megszerzéséhez, mind a digitális kompetencia fejlesztéséhez (Digital literacy, Digitalliteracy.gov, Glistler: 1997, IEEE&ACM Report 2013, Informatics education: 2013, Kerettanterv: 2008, Kerettanterv: 2013, Microsoft Digital Literacy, Northstar Digital Literacy Project, Welcome to Digital Literacy).

---

<sup>1</sup> Képzeljük el néhány éven belül azt az óriási változást, ami jelenlegi informatika tantervekben meghúzódód akadályok feloldásával lehetségessé válik. A Word és Excel használatát tanuló unatkozó gyerekek, és ezeket tanító unatkozó tanárok helyett, elérhetjük, hogy 11 éves gyerekek képesek lesznek 2D-s animációkat írni Scratch-csel. 16 éves korukra megértik a formális logikát, ami korábban csak egyetemi tananyag volt, és képesek lesznek saját Appokat írni okostelefonokra.

- A programok nem hibásak, ők csak programok, de mint ilyenek algoritmus alapúak, tehát alkalmasak arra, hogy rajtuk keresztül fejlesszük a tanulók számítógépes gondolkodását, algoritmikus készségét (Csernoch–Biró: 2015, Wing, J. M.: 2006).
- Az pedig, hogy a tanárok unalmasak és unalmas órákat tartanak nagyrészt a tanárképzésnek a hibája, amely nem készíti fel a hallgatókat a nem tradicionális számítógépes környezetekben a számítógépes gondolkodás, az algoritmikus készség fejlesztésére.
- Ugyanezek a tanárok nem fognak izgalmasabb Scratch órát tartani. Ha nem tanulták meg a szöveg- és a táblázatkezelést, akkor miből következik, hogy meg fogják tanul a Scratch programozást, a mobil programozást és még a formális logikát is?
- A táblázatkezelés programozás, egy funkcionális nyelv, amely egyrészt szolgálhat kezdő nyelvként, másrészt pedig végső programozási nyelvként azoknak, akik felhasználók maradnak (Csernoch: 2014, Csernoch–Balogh: 2010, Csernoch–Biró: 2013, Csernoch–Biró: 2015, Sestoft: 2010, Warren: 2004).
- Autentikus táblázatokkal táblázatkezelőben is ugyanúgy végezhetünk igazi információ visszanyerést, mint hagyományos programozási nyelveken (Csernoch: 2014, Csernoch–Balogh: 2010, Csernoch–Biró: 2015).

*„Now, our new curriculum teaches children computer science, information technology, and digital literacy: teaching them how to code, and how to create their own programs; not just how to work a computer; but how a computer works, and how to make it work for you.”<sup>2</sup>*

Gove-nak abban teljesen igaza van, hogy a tanárok feladata a számítógép koncepció kialakítása. Véleményével az a gond, hogy ezt leszűkíti a programozásra. Az informatikatanároknak valóban a számítógép koncepció kialakítása, az algoritmikus gondolkodás kialakítása és fejlesztése a feladata. Ehhez természetesen a számítógép, mint eszköz használható, de nem az kell legyen az elsődleges, hogy használatot tanítsunk, ezt minden esetben meg kell előznie a koncepció kialakítása (IEEE&ACM Report:

---

<sup>2</sup> Az új tantervünk tanít a gyerekeknek számítógép tudományt, információ technológiát és digitális írástudást: megtanítja, kódolást, hogy hogyan tudnak saját programokat írni, nem csak azt, hogy hogyan működik a számítógép, hanem azt is, hogy hogyan lehet a számítógépet rávenni arra, hogy végrehajtsa az utasításait.

2013). További gond Gove megközelítésével, hogy úgy gondolja, hogy a tantervek tanítják meg a tanulókat. Nem, a tantervek ilyen nem csinálnak, a tanárok tanítanak. A tanárok gondolkodás módját kell megváltoztatni, olyan módszereket kell a kezükbe adni, amivel képesek a tanulók és a saját algoritmikus készségüket is fejleszteni, bármilyen számítógépes környezetben, és nem kizárólagosan a programozás-oktatásban (Csernoch–Biró: 2015).

### **2.1.2. Szöveges e-dokumentumok**

Szöveges e-dokumentumok csoportjába tartozik az összes olyan szövegalapú dokumentum, amelynek létrehozásához, módosításához, megjelenítéséhez számítógépet használunk. A meghatározás alapján nyilvánvalóan ebbe a csoportba tartozik minden szöveg- és kiadványszerkesztővel készített dokumentum. Ezen dokumentumok többségére igaz, hogy végső formájukban papír alapú dokumentumok, a programok a dokumentum előállításában játszanak szerepet. Ezen túl azonban a szöveges e-dokumentumok csoportjába tartoznak olyan dokumentumok is, amelyeknél nem csak az előállításhoz, hanem a végső megjelenítéshez is szükséges a számítógép. Ezek közül a két leginkább ismert és elterjedt a dokumentum típus a weblapok és a prezentációk.

A nyomtatott dokumentum fogalma az előzőeknek megfelelően kiszélesedik. A hagyományos, papír alapú dokumentumok mellett megjelennek a digitális kijelzőn megjelenített dokumentumok is. A tananyagban a nyomtatott dokumentumot ebben a szélesebb értelemben fogjuk használni.

### **2.1.3. Helyesen formázott dokumentum**

Elsőként annak definiálása szükséges, hogy mikor tekintünk egy dokumentumot helyesen formázottnak (Csernoch 2009: 119–137). Helyesen formázottság alatt azt a tág fogalmat értjük, amely a tényleges formázottságon túl magába foglalja egy e-szöveggel szemben állított valamennyi követelmény teljesülését. Mindaddig, amíg ez a fogalom nem egyértelmű a szöveg előállítás folyamatában részt vevő valamennyi partnernek, addig nincs is értelme hibákról beszélni.

### **2.1.4. Definíció hiányának következményei**

Az 1. ábra elemzéséből egyértelmű, hogy a helyesen formázott szöveg definíciójának hiányában milyen értelmezésbeli különbségek adódhatnak.



$$\begin{array}{l}
 \text{d) } \frac{d+c-2d-3c+d-d-c}{3 \cdot 4 \cdot 6 \cdot 12 \cdot 12 \cdot 6 \cdot 6} \\
 \frac{4d-4d+d-2d+3c-3c-2c}{12 \cdot 12 \cdot 12 \cdot 12 \cdot 12 \cdot 12 \cdot 12} \\
 \frac{d-c-18+12-3+2-1}{12 \cdot 6 \cdot 12 \cdot 6 \cdot 2}
 \end{array}
 \qquad
 \begin{array}{l}
 \text{d) } \frac{d+c-2d-3c+d-d-c}{3 \cdot 4 \cdot 6 \cdot 12 \cdot 12 \cdot 6 \cdot 6} \\
 \frac{4d-4d+d-2d+3c-3c-2c}{12 \cdot 12 \cdot 12 \cdot 12 \cdot 12 \cdot 12 \cdot 12} \\
 \frac{d-c-18+12-3+2-1}{12 \cdot 6 \cdot 12 \cdot 6 \cdot 2}
 \end{array}$$

## 2. ábra

Barkácsolással készített képletek egy dokumentum két különböző pontján. A két kép nem teljesen egyforma, ami azt is mutatja, hogy nagy valószínűséggel nem másolással készült a képletek második verziója

számlálók	egy bekezdés
nevezők	másik bekezdés
számlálók, nevezők	szóközők
=	két szóköz dupla aláhúzással
törtvonal	aláhúzás
+	alsó index és betűméret növelése
-	alávonás karakter
egész számok	alsó index mindkét ábrán, bal oldali ábrán a betűméret növelése is
sorvégi =	3. ábra
feladat számozása	kézi
bekezdések közötti térköz	üres bekezdések
lábjegyzet	első képletnél igazi lábjegyzet, többi képletnél csillag felső indexben

## 1. táblázat

A 2. ábra képleteiben felhasznált trükkök



### 3. ábra

*A 2. ábra készítője a szövegszerkesztőt hibáztatja azért, mert nem tudott a sorok végére egyenlőségjelet elhelyezni, a magabiztos nem tudásnak egy rendkívül magas szintje*

#### 2.1.5. Definíció

Az e-dokumentumok előnye minden eddigi szöveg előállítási eszközzel és módszerrel szemben, hogy módosíthatóak. A definíciónak tehát megengedőnek kell lennie, a módosítást továbbra is engedélyeznie kell. A kérdés az, hogy melyek azok a módosítások, amelyek megfelelnek a helyesen formázottságnak, és melyek azok, amelyek már nem.

Röviden megfogalmazva azt mondhatjuk, hogy egy dokumentum helyesen szerkesztett, helyesen formázott, ha az módosításokkal szemben invariáns (Csernoch 2009: 119–137). Tovább részletezve, ez annyit jelent, hogy engedélyezett a módosítás, de csak és kizárólag olyan módosítások, amelyek a felhasználói szándékból erednek. A szövegetest módosítása tehát ne vonja maga után a szöveg újragépelést, újraformázását.

A dokumentum a helyesen formázottságon túl eleget kell tegyen a nyelvi, tartalmi, stilisztikai, tipográfiai és számos más, szöveg és tartalom specifikus szabálynak is, amelyek áttekintésére a 2.2. fejezetekben kerül sor.

Amennyiben az 1. ábra szövegét módosítani kívánjuk, akkor az a szóközök és tabulátorok törlése és hozzáadása nélkül nem megvalósítható, ami nyilvánvalóan túlmutat a felhasználói szándékon. Ez egyértelmű jele annak, hogy dokumentum helytelenül formázott.

#### 2.2. Szöveges e-dokumentumok hibái, hibafelismerés (18. óra)

Sajnálatos módon, a szöveges e-dokumentumok többségére jellemző, hogy kisebb, nagyobb mértékben, de hibásak. Alig találunk olyan dokumentumot, amelyik hibáktól mentes, és nem kevés azoknak a száma, amelyek halmozottan tartalmaznak hibákat. A hagyományos kézírásos, valamint a nyomdai forrásból származó dokumentumok sem hibamentesek, de a szövegszerkesztővel készült dokumentumok esetében megnőtt a

hibalehetőségek, hibaforrások száma, amit a felhasználók, sajnálatos módon, maximálisan ki is használnak.

### **2.2.1. Lehetséges hibaforrások**

A szövegszerkesztésben az a nehéz, hogy a szöveg rendkívül sok szabályrendszernek kell megfeleljen, ugyanakkor minimális segítséget nyújtanak az elérhető szoftverek. A hibák és hibatípusok ismertetését követően (2.2.1.1. – 2.2.2. fejezetek) egyértelművé válik, hogy az egyetlen ma elérhető segédeszköz a szöveges e-dokumentumok debuggolására a nem túlságosan megbízható helyesírás-ellenőrök (Kis–Prószéky: 1999). Nincsenek azonban szoftveres eszközeink a további hibák javítására, ennek következtében ez a felhasználó feladata. A felhasználó azonban csak abban az esetben tudja saját szövegének az ellenőrzését, tesztelését (debugging) hatékonyan elvégezni, ha ismeri a szükséges szabályrendszereket és alkalmazni is tudja ezeket a szabályokat. Fontos tehát hangsúlyozni, hogy a koncepció kialakítása mindenképpen meg kell előzze a használatot.

Azt a kérdést is fel lehet tenni, hogy programozni nehezebb vagy szöveget szerkeszteni. Nem feltétlenül kapunk választ a kérdésünkre, mivel a válasz rendkívül összetett, de el lehet gondolkodni a problémán. A kétféle tevékenységet összehasonlítva az elvitathatatlan tény, hogy a programozási nyelveknél legalább rendelkezésünkre áll egy tökéletes szintaktikai elemző, ennek következtében, legalább szintaktikai hibát nem tartalmaznak a programok. A nyelv szintaktikája is lényegesen egyszerűbb, mint a természetes nyelvek esetében, ezen túl a mesterséges nyelvek egyértelműen definiáltak. Szemantikai hiba még mindig lehet a programokban, de míg programok esetében az output ellenőrzése egy bevált gyakorlat, addig szövegszerkesztésnél ez esetleges, nagyban a felhasználói igényesség függvénye. Programok esetében ezzel a két hibacsoporttal kell számoljunk, míg szövegszerkesztés esetén lényegesen magasabb a lehetséges hibatípusok száma.

#### **2.2.1.1. E-dokumentumok nyelvi problémái**

A helyesírási szabályok továbbra is érvényben vannak, azonban egyes, korábban kevésbé hangsúlyos területek felerősödnek, míg mások újonnan jelennek meg. A nyelvhelyességi hibák a végső nyomtatott formában is nyilvánvalóak, és mindenképpen kerülendőek. Napjainkban egy olyan

irányzat kezd elterjedni, hogy az e-dokumentumokban megbocsátható a nyelvi hiba. Ez azonban nem így van, továbbra is igaz, hogy nyelvi hibáktól mentes dokumentumok előállítása az elsődleges cél (Antalné–Balázs: 1998, Bódi: 2014a, Bódi: 2014b, Graf 199: 128, Javaslat a magyar nyelv alkotmányi szintű védelmével kapcsolatban: 2010, Szabó 2005: 59–70, Z. Karvalics: 2002). A rendelkezésre álló számítógépes eszközök, beleértve a hardver és a szoftver eszközöket is, azt a látszatot keltik, hogy használatukkal nagyobb esélyünk van nyelvi szempontból hibamentes dokumentumok előállítására. Ez egyrészt igaz, másrészt azonban nem szabad megfeledkeznünk arról, hogy ezek az eszközök algoritmusok alapján működnek, és napjainkban a természetes nyelvi szövegek számítógépes feldolgozásának problémája nem megoldott (Figueredo–Varnhage 2006: 721–732, Kis–Prószéky: 1999). Tehát ezen eszközök használata mindenképpen javasolt, de körültekintéssel és fokozott odafigyeléssel. Az alábbi felsorolás a természetes nyelvű e-szövegek feldolgozásával kapcsolatos nyelvi aspektusokat foglalja össze.

– Végleges forma

Az e-dokumentumok nyomtatott formában jelennek meg – legyen az papír vagy számítógép alapú –, és a nyomtatott forma egyértelművé teszi a karaktereket. Ennek egyik következménye, hogy a kézírásos dokumentumokkal szemben, megnőtt az írásjelek helyesírásának szerepe (4. ábra) (Bujdosó–Csernoch 2008: 122–128, Csernoch 1997: 375–382, Csernoch 2010: 247–262, Csernoch–Bujdosó 2009: 5–10, Csernoch–Bujdosó 2009).

*A házi feladat korlátai, mennyiségének elvei:*

- a) Inkább kevesebbet adjunk, de rendszeresen!
- b) A szóbeli és írásbeli házi feladat együttes mennyisége nem lehet túlzó.  
(Tantárgyanként 15-30 perc alatt elvégezhető legyen)
- c) A fogalmazások maximális hosszát a szaktanár határozza meg (5-6. oszt.: 1 oldal 7-8. osztály 1 ½ oldal), ezzel is tekintetbe véve az életkori sajátosságokat.

*A házi feladat korlátai, mennyiségének elvei.\**

- a) • Inkább kevesebbet adjunk, de rendszeresen.\*
- b) • A szóbeli és írásbeli házi feladat együttes mennyisége nem lehet túlzó.\*  
.....(Tantárgyanként 15-30 perc alatt elvégezhető legyen).\*
- c) • A fogalmazások maximális hosszát a szaktanár határozza meg (5-6. oszt.: 1 oldal \*  
.....7-8. osztály 1 ½ oldal), ezzel is tekintetbe véve az életkori sajátosságokat.\*

#### 4. ábra

*Nyomtatásban is jól látható nyelvi és tipográfiai hibák (bal), a nem-nyomtatódó karakterek egyértelművé teszik a szöveg tördelési hibáit (ahazifeladat.doc)*



## – Nem-nyomtatódó karakterek

Az e-dokumentumok karakterkészletének másik sajátossága az újonnan megjelent karakterek, amelyek közül a meghatározóak a nem-nyomtatódó karakterek. Ezek közül a leggyakoribbak a szóköz, a bekezdés vége és a tabulátor karakter (Csernoch 2009: 119–137, Csernoch 2010: 247–262). Ezen karakterek intenzív és indokolatlan használata az e-dokumentumok egyik leggyakoribb problémája. A mintaként csatolt dokumentumok mindegyike tartalmaz ilyen jellegű hibát. Sajnálatos módon a nem-nyomtatódó karakterek helyes használata jelenleg nem szabályozott.

## – Karakterkódolás

A számítógépes szövegfeldolgozás további problémája a karakterkódolás. A Unicode egy lehetséges megoldásnak tűnik, de a korábbi kezdetleges megoldások még ma is kísértének (5. ábra). Ennek következménye, hogy napjainkban is nagy számban találunk olyan dokumentumokat, amelyek nem képesek a kiválasztott nyelv karaktereinek helyes megjelenítésére, valamint folyamatosan tapasztalható a nem megfelelő karakterek használata (5. ábra), (Bujdosó–Csernoch 2008: 122–128, Bujdosó–Csernoch 2014, Csernoch–Bujdosó: 2009 a: 5–10, Csernoch–Bujdosó: 2009b, German Punctuation, Language Specific Quoting and Quotation Marks, Quotation Marks and their Unicode numbers, Quotation Marks a, Quotation Marks b).

7258-an vettek részt az idei, sorrendben 32. Balaton-átúszáson Révfülöp és Balatonboglár között. A férfiak között a 15 éves óbecsei Balzam Bence érkezett elsőként a célba 1 óra 10 perces idővel, a nőknél pedig Horváth Zsanett vihette el a fődíjat. A verseny jelenleg is zajlik, az utolsó induló fél kettőkor ugrott a vízbe. Így az, hogy a legjobb időt ki ússza az 5200 méteres távon, még nem derült ki.

Procedure	Time	Interaction
Teacher greets students then students greet T.	Approx. 1 minute	T->Ss Ss->T
Teacher asks Ss questions about their lottery habits: have they played the lottery yet? If yes, why? If no, why not?	Approx. 8 minutes	T->Ss Ss->T

**5. ábra**

*Helytelen karakterkódolás<sup>3</sup> (bal), karaktert helyettesítő barkácsolás (jobb)*

<sup>3</sup> 15 éves fiatalember ért először a célba az idei Balaton-átúszáson. <http://www.origo.hu/itthon/20140719-15-eves-fiatalember-ert-eloszor-a-celba-az-idei-balaton-atuszason.html>. Letöltve: 2014. 07. 19.

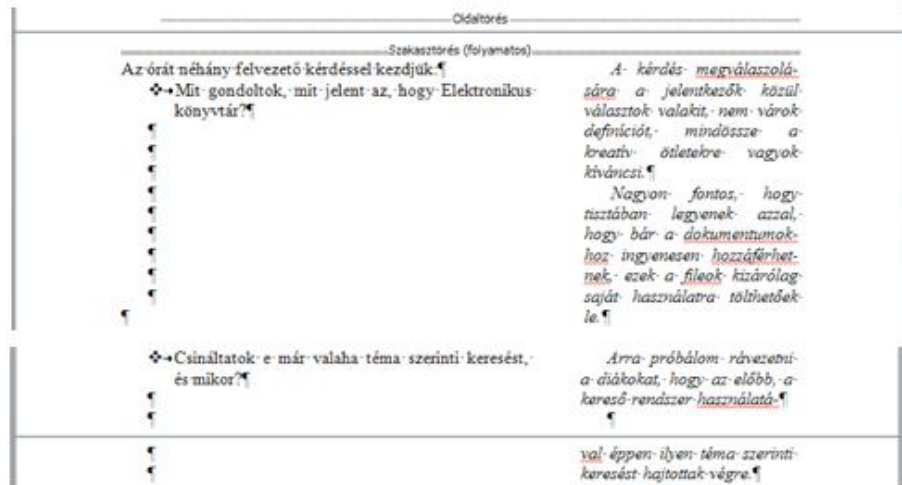
vesszük a "jelleg-töltést":	$n \rightarrow p^+ + e^- + \nu^-$	vesszük a "jelleg-töltést":	$\bar{n} \rightarrow \bar{p}^+ + e^- + \bar{\nu}^-$
barion-töltés: 1	1 0 0	barion-töltés: -1	-1 0 0
lepton-töltés: 0	0 +1 -1	lepton-töltés: 0	0 +1 -1

**6. ábra**

*Karaktereket helyettesítő barkácsolások, hibás idézőjel*

– Elválasztás

Az elválasztás is átértelmeződött az e-szövegekben. A kézírásban használt elválasztási szabályok továbbra is érvényesek, ám a technikai megoldás gyökeresen megváltozott. A kézírásban használt elválasztási módszer e-dokumentumokban nem alkalmazható, mert tördelési hibához vezet.



**7. ábra**

*Kézi elválasztások. Mindkét ábra egyértelműen mutatja, hogy kézi elválasztást használva a helyesírásellenőrző nem ismeri fel a szótöredékeket. Az alsó mintája a használatával szóban a kötőjelen túl még tartalmaz egy üres bekezdést is*

## – Nyelvi beállítások

A digitális nyelvhelyesség magában foglalja a nyelvi beállításokat, a helyesírásellenőrzők installálását és ezek használatát. A nyelvi problémák kezelésére azért van szükség, mert egyrészt a szövegek kódolása megköveteli a megfelelő nyelvi beállításokat, másrészt a helyesírásellenőrzők nem 100%-os működése a felhasználók részéről felelős döntéshozatalt vár. Ezen döntések alapja természetesen a hagyományos helyesírási ismeretek (Csernoch–Bujdosó: 2009 b, Figueredo– Varnhagen 2006: 721–732).

**4. Unterrichtsphase: Selektives Leseverstehen (10 Minuten)**

Die Aufgabe ist die folgende: Lehrbuch 24/5. In Bekannschftsanzeigen suchen die Jugendlichen Brieffreunde. Sie schreiben über sich selbst und ihre Hobbys. Die Schüler arbeiten paarweise. Ein Paar liest je eine Anzeige und sie füllen die Tabelle aus. Die nötigen Informationen sind: Name, Alter, Hobbys. Beim kontrollieren stellen sie die Person vor – eventuell helfe ich ihnen mit Fragen – und die anderen schreiben die fehlenden Informationen ein. Ich korrigiere die Fehler.

**4. Unterrichtsphase: Selektives Leseverstehen (10 Minuten)**

Die Aufgabe ist die folgende: Lehrbuch 24/5. In Bekannschftsanzeigen suchen die Jugendlichen Brieffreunde. Sie schreiben über sich selbst und ihre Hobbys. Die Schüler arbeiten paarweise. Ein Paar liest je eine Anzeige und sie füllen die Tabelle aus. Die nötigen Informationen sind: Name, Alter, Hobbys. Beim kontrollieren stellen sie die Person vor – eventuell helfe ich ihnen mit Fragen – und die anderen schreiben die fehlenden Informationen ein. Ich korrigiere die Fehler.

**8. ábra**

*Hibásan (felső) és helyesen (alsó) beállított nyelv*

## – Szövegbevitel módja

A szövegbevitel, javítás módja is megváltozott. A kézírást felváltotta a beviteli eszközökön keresztül történő szöveg előállítás: a gépelés, a nyomtatott vagy a hangalapú szövegfelismerés. Bármelyik módszert is választjuk, a szövegbevitel során lényegesen magasabb számban történik hiba, mint a kézírás során. A gépelés során keletkezett hibák oka, hogy nem tudunk olyan biztonsággal gépelni, mint ahogy kézírással írunk. A nyomtatott és hangalapú szövegfelismerésnél pedig a szoftver nem biztonságos működésének következtében várhatóak hibák.

A felsorolt nyelvi hibaforrások elkerülése, annak megtanítása, hogy a számítógépes nyelvi eszközöket hogyan lehet hatékonyan használni, nem csak az informatikaoktatás feladata. Ez az a tudományterület, amelynek részterületeit a nyelv- az informatikatanárok között meg kell osztani.

Mindenképpen kiemelkedő szerepe van az anyanyelvi tanároknak, de az idegen nyelvet oktató tanárok felelőssége is rendkívül fontos és jelentős.

### **2.2.1.2. Tipográfiai ismeretek hiánya**

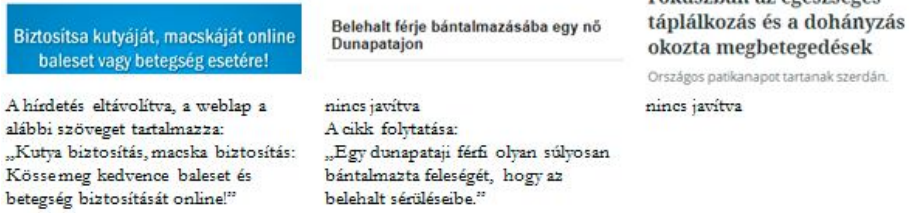
A szöveges e-dokumentumok, a megjelenítési platformhoz igazodva, meg kell feleljenek a tipográfiai szabályoknak. Az egyes megjelenítési platformoknak megvannak a sajátos tipográfiai szabályai, amelyek betartása elengedhetetlenül fontos ahhoz, hogy az információátadást a megjelenítés formája ne akadályozza, másrészt azért, hogy esztétikailag is élvezhető dokumentumot állítsunk elő, amelyet élvezettel vesz a felhasználó a kezébe (Bujdosó–Csernoch: 2009, Jury: 2004, Jury: 2006, Jury: 2007, Virágvölgyi Péter: 2004).

A tipográfiai egyik sajátossága, szemben a nyelvi ismeretekkel, hogy a számítógépes szövegszerkesztés megjelenése és elterjedése előtt, szinte kizárólagosan a nyomdaiparban alkalmazott szakemberek végezték a dokumentumok tördelését és végső formájának kialakítását. Tehát egy nagyon szűk kör foglalkozott ezzel a területtel. A szövegszerkesztők elterjedésével azonban ugrásszerűen megnőtt az önkéntes tipográfusok száma, akik többsége azt se tudja, hogy létezik ez a tudományterület. A tudatlanság tudatlanságának kellemes állapota ez (Kruger–Dunning 1999: 1121–1134). Az eredménye azonban olyan dokumentumok sokasága, amelyek minden létező tipográfiai szabályt félrúgva készülnek.

A tipográfiai ismeretek tanítása azonban nem feltétlenül az informatikaoktatás feladata. A rajz- és művészeti oktatás tanárai az erre felkészült szakemberek, akik felsőfokú tanulmányaik során megfelelő szintű képzésben részesülnek. Sajnálatos módon, az informatikatanárok a jelenlegi oktatási rendszerben semmiféle ilyen jellegű, szervezett oktatási formában nem vesznek részt. Érdemes megjegyezni, hogy a tipográfiai ismeretek nem csak a szövegszerkesztésnél követelmény, hanem bármilyen olyan számítógépes tevékenységnél, amelynek végeredménye publikálásra kerül.

### **2.2.1.3. Tartalmi, szemantikai hibák**

A szöveg tartalmi hibái egyértelműen a nyelvszakos tanárok és a témától függően az egyéb szakos tanárok a felelősek. Az informatikatanárok nem lehet a 21. század polihisztorai, a tantervek bármennyire is ezt sugallják.



### 9. ábra

*Szemantikai hibákat tartalmazó szövegrészletek bal<sup>4</sup>, közép<sup>5</sup>, jobb<sup>6</sup>*

#### 2.2.1.4. Tördelési hibák

A tördelési hibák az e-dokumentumok legkritikusabb hibacsoportja (Csernoch 2009: 119–137, Csernoch 2010: 247–262). Mindaddig, amíg ilyen típusú hiba/hibák találhatóak a dokumentumban, addig biztosan nem felel meg a helyesen szerkesztett dokumentum definíciójának. A tördelési hibák számtalan formája ismert, de a leggyakoribbak a szóközzel, enterrel, tabulátorral barkácsolt szövegek (1. ábra, 13. ábra, *diable.doc*, *csapat-verseny.doc*). Nem ritka azonban az objektumok hibás formázásából, a hasábok, táblázatok helytelen használatából, az elválasztás egyéni módon történő értelmezéséből származó tördelési hiba sem, és a lista még tovább bővíthető.

<sup>4</sup> <http://www.aegon.hu/kisallat-biztositas/kisallat-biztositas.html>. Letöltve: 2013. 04. 14. Ellenőrizve: 2014. 07.28.

<sup>5</sup> <http://www.haon.hu/belehalt-ferje-bantalmazasaba-egy-no-dunapatajon/2287253>. Letöltve: 2013. 08.04. Ellenőrizve: 2014. 07.28.

<sup>6</sup> <http://www.webbeteg.hu/cikkek/egeszseges/16173/patikanap--fokuszban-az-egeszseges-taplalkozas-es-a-dohanyzas-okozta-megbetegedések>. Letöltve 2014. 06. 03. Ellenőrizve:2014. 07.28.

Feladatok témaí	Időtartam	Pontszám
Idégen nyelv	15 perc	15 pont
Informatika	15 perc	15 pont
Készségfejlesztés	15 perc	15 pont
Belégtrendezési ismeretek	45 perc	45 pont
Akadémpálya	5 perc	45 pont
Fizikai állapotfelmérés	30 perc	125 pont (2x25)
Lövészet	5 perc	25 pont
Katasztrófavédelemi (tűz- és polgári védelmi) szakmai és ügyességi gyakorlat	5 perc	20 pont
<b>ÖSSZESEN</b>	<b>135 perc</b>	<b>305 pont</b>

10. ábra

Formázási hibák: táblázat elhelyezése – szöveg körbefuttatása: körbe –, számok középre igazítása – középre – (még az üres enterek is eltérő formátumokat kaptak) (csapatverseny.doc)

zi-ták exponenciálisan csökken. 18.5 szerint, amiből.

$$\frac{-\ln(\Phi/\Phi_0)}{nt} = 2R^2\pi \quad (2.4)$$

A 2.2 ábra az abszorpciós kísérlet elvét mutatja. A mintával és nélküle végzett mérésekből  $\Phi/\Phi_0$  elvileg nagyon pontosan meghatározható.  $R \sim 1,3 \cdot A^{1,3}$  a tömegszámfüggés itt is.

d) **Atom spektroszkópia.**  
A Coulomb-erő  $1/r^2$ -függése miatt az elektronok kötési energiáját az atommag töltésseloszlásának mérete befolyásolja. Az energiaszintek helye is a spektrumban. Ebből a mag sugárizotópfüggése meghatározható.

A Bohr-elmélet szerinti  $n$  kvantumok körpálya sugara a  $n$  kerületű részecske tömegétől,  $Z_1$  töltésétől, a  $Z_1$  rendszám-tól  $r_n = n^2 \cdot h^2 / (Z_1 Z_2 e^2 m)$  alakban függ. Ha az atomi K-héjon elektron helyett a nála 207,4-szer nehezebb

céltárgy, abszorbers

bombázó részecskék

detektor

$\Phi_0$

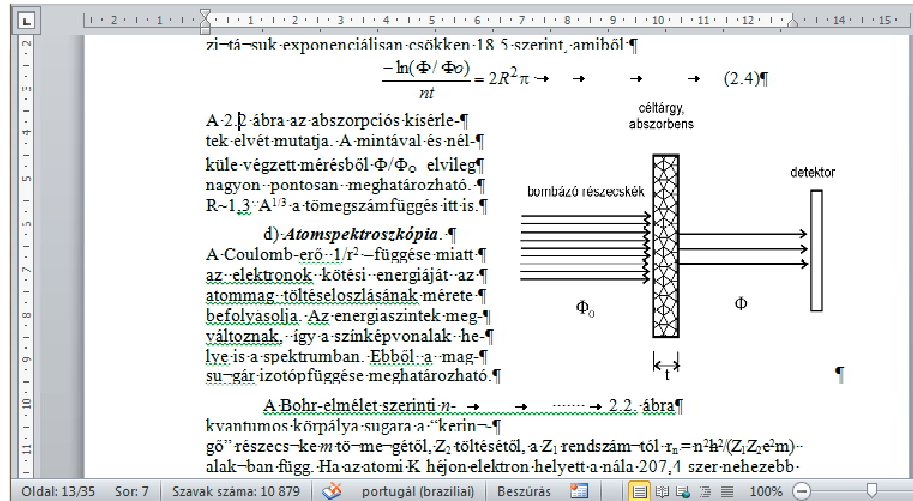
$\Phi$

$t$

2.2. ábra

11. ábra

Barkácsolás magas fokon: Tördelési és formázási hibák halmozása: Kép elhelyezése szövegdobozban, szövegdoboz elhelyezése: Szöveg elé, hibás nyelv beállítás (angol), sorok végén kézi tördelés, kézi elválasztás, kézi sorkizárás szóközökkel, igazítás szóközökkel és tabulátorokkal, bekezdés és ábra kézi számozása, feltételes elválasztások, hibás idézőjelek



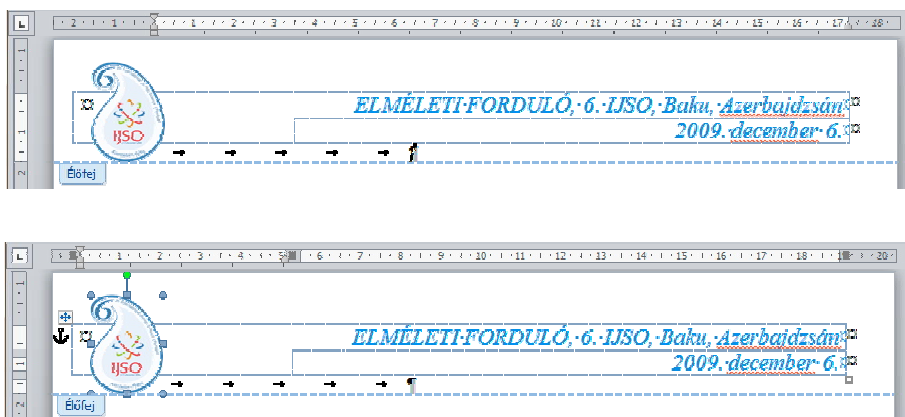
12. ábra

Kézi keresztivatkozás, hibás nyelv beállítás (portugál)

### 2.2.1.5. Formázási hibák

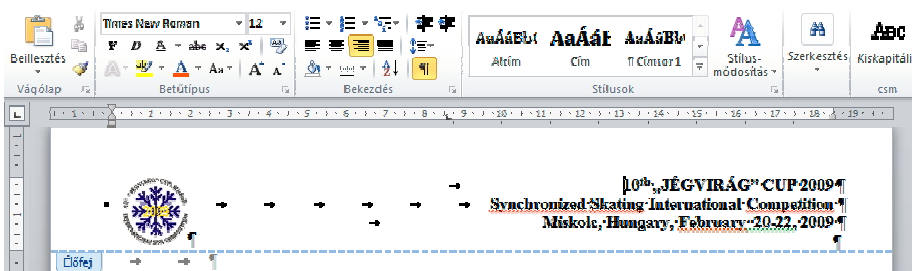
Az e-dokumentumok egy új keletű hiba csoportot is létrehoztak, ez pedig nem más, mint a formázási hibák csoportja. Azok a formázási parancsok tartoznak ebbe a csoportba, amelyeket a felhasználó kiadott, de hibásan. A felhasználó, tehát formázott, végrehajtott egy formázási parancsot, de abban a környezetben az hibás. A leggyakoribb ebbe a csoportba tartozó hibák: igazítás helyett egy behúzás alkalmazása (*csapatverseny.doc*, *diable.doc*), nyelv hibás megválasztása (13. ábra), táblázatok elméretezése (13. ábra) vagy fölösleges formázások (13. ábra felső, dőlt).

Előfordulhat az is, hogy egy hibásan kiadott formázási parancs más típusú hibát is generál. Erre példa az aláhúzás alkalmazása: egy hibás formázási parancs következménye egy tipográfiai hiba (22. ábra). Hasonlóan, a képek hibás elrendezése egy formázási paranccsal valósítható meg, de a végeredmény ez esetek többségében egy tördelési hiba (10. ábra, 13. ábra alsó, 15. ábra, *PI-11-11-ffHH-d.doc*, *le\_carving\_word.doc*).



13. ábra

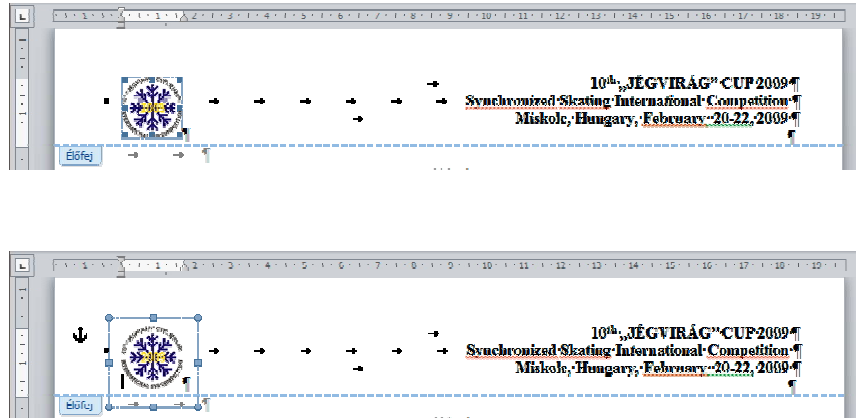
Formázási hibák: táblázat szélessége, kép elhelyezése, nyelv beállítása, dőlt (6thijs\_hun.doc)



14. ábra

Formázási hibák: fölösleges balra tabulátor egy jobbra igazított bekezdésben (announcement.doc)



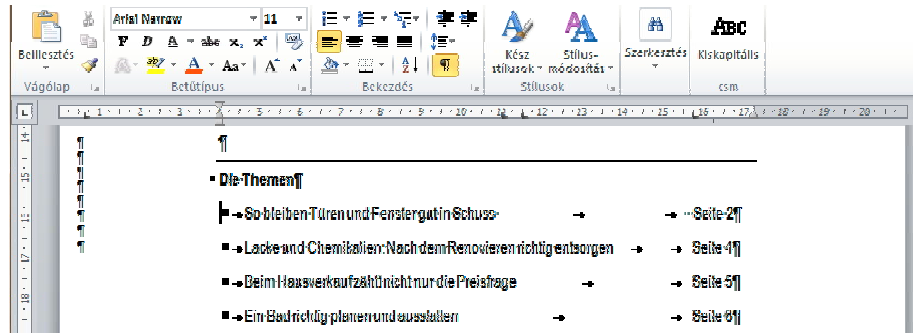


**15. ábra**

*Formázási hibák: kép elhelyezése szövegdobozban (announcement.doc)*

A 13. ábra és az 15. ábra összehasonlításából egyértelmű, hogy a 13. ábra táblázata egy jó megoldás lett volna a kép és a szöveg elrendezésére, de táblázat létrehozását követően a szerzők már nem tudták hogyan kell a kép elhelyezését beállítani. Az 15. ábra logója kapott még egy szövegdobozt is, amit ugyanolyan elhelyezéssel formáztak, mint a 13. ábra logóját, tehát egy kép elhelyezéséhez két formázási hibát is elkövettek. További tipikus formázási hibák találhatók a *csapatverseny.doc* dokumentumban.

Már a kezdetektől kísért a margó–behúzás probléma. Margó helyett behúzás állítása két formázási hibát is magában hordoz: a hibás margóbeállítás követően a behúzás barkácsolása (16. ábra, *bauenwohnen.doc* és *mozirajongok.rtf*).



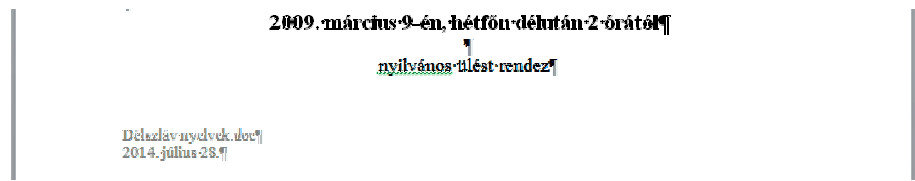
16. ábra

Formázási hibák: a 0 cm-es bal margót egy 4 cm-es bal behúzással próbálta korrigálni, fölösleges és hibás tabulátorok, üres enterekkel teli szövegdoboz (bauenwohnen.doc)

### 2.2.1.6. Hivatkozások

A szövegen belüli hivatkozások nem megfelelő kezelése gyakori hibaforrás. A hivatkozásokat mindenképpen a szövegszerkesztő programok erre alkalmas eszközeivel érdemes létrehozni, mivel ezek az automatikus frissítést is elvégzik. A kézzel vagy nem megfelelő eszközökkel létrehozott hivatkozások nem módosíthatóak, így nem tesznek eleget a helyesen formázott szöveg definíciójának. Egy hosszabb lélegzetű dokumentum teljes hivatkozásrendszerének a kézi újraszámozása rengeteg hibaforrást rejt magában, ezen túl rendkívül időigényes folyamat.

Az ilyen típusú hibák közül a leggyakoribbak a következők: a kézi tartalomjegyzék (16. ábra), a kézi lábjegy (fenstersystem.doc), ábrák és táblázatok kézi számozása (le\_carving\_word.doc), majd kézi hivatkozás ezekre (le\_carving\_word.doc), kézi sorszámzású irodalomjegyzék és kézi hivatkozás a forrásokra.



### 17. ábra

*A meghívóra hibásan beillesztett dátum mező eredményeként a dokumentum újbóli megnyitásakor az éppen aktuális dátum jelenik meg az élőlábban és nem a meghívó készítésének dátuma*

#### 2.2.1.7. Körlevelek

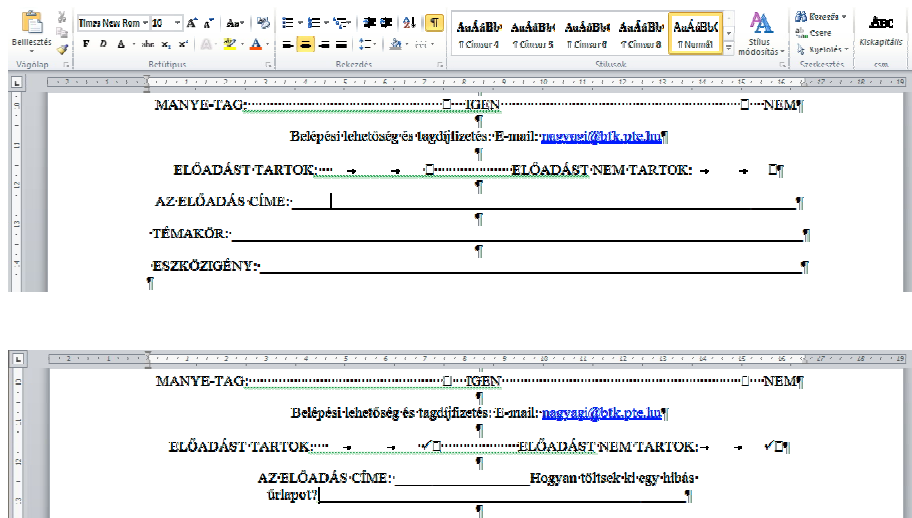
A körlevelek egy népszerű, ám még sem elég széles körben használt lehetősége a szövegszerkesztő programoknak. A körlevél, jellegéből adódóan az a tipikus fogalom, amely a levelezési szokások gyökeres változáshoz vezetett, és amely megvalósításához a szövegszerkesztő programok egy tökéletes eszköz. Gyakorta visszatérő feladatok a különböző vizsgákon. Az ECDL vizsga például minden feladatban megköveteli a körlevél készítését. Ilyen gyakoriságú számonkérés, azonban nem indokolt. Az egyesével készülő dokumentumok száma jóval meghaladja a körlevelek – törzsdokumentumok – számát.

#### 2.2.1.8. Űrlapok

A körlevelekkel szemben, az űrlapok tűnnek a szövegszerkesztő programok fekete bárányának. Míg a körlevélkészítést a különböző szövegszerkesztői vizsgák is megkövetelik, szinte említés sem történik az űrlapokról. Annak ellenére, hogy a szövegszerkesztő programok remek eszköz az űrlapok létrehozásához, és az űrlapok létrehozásához szükséges felületi eszközök is rendkívül egyszerűen használhatóak. Fontos azonban megjegyezni, hogy a webes felületeken létrehozott űrlapok azonban lényegesen könnyebben kezelhetőek, mint a birotikai szoftverekben.

Az űrlapokkal nap mint nap találkozunk: fájlban megkapott űrlap kitöltését kérik számítógéppel, majd a kitöltött fájl visszaküldését. Nem egyszer .pdf formátumban kapjuk a kitöltendő fájlt. Ilyenkor nincs más teendő, mint kinyomtatjuk a fájlt, kézzel kitöltjük, beszkenyeljük és ezt a fájlt küldjük vissza. Amennyiben szerkeszthető formátumban kapjuk a

hibás űrlapot és nem akarjuk az egész dokumentumot újra formázni, akkor csak a dokumentum barkácsolása marad.



**18. ábra**

*Hibásan szerkesztett üres űrlap (felső) (manye.doc), kísérlet az űrlap kitöltésére szövegszerkesztő programmal (alsó) (manye\_csm.doc)*

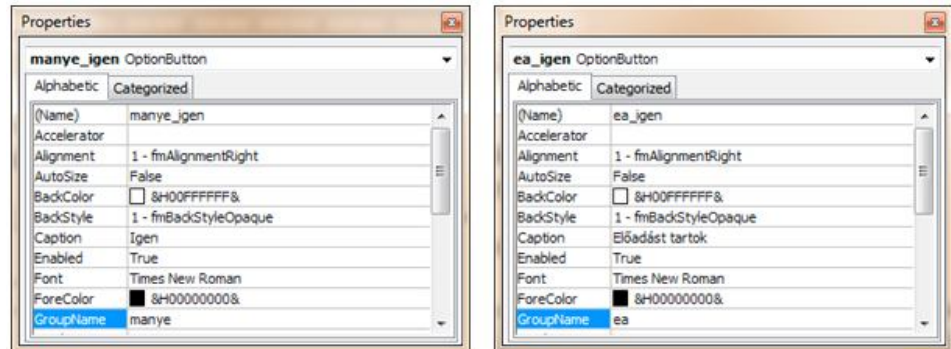
A 18. ábra felső mintája egy üres űrlapot mutat, amit kitöltve e-szöveggként kellett volna visszaküldeni. A nem-nyomtatódó karakterek és a kurzor pozíciója egyértelműen mutatja, hogy ez a szöveg Mekk Elek stílusban készül (18. ábra felső). A kitöltése pedig igen komoly nehézségekbe ütközik (18. ábra alsó):

A választási lehetőségeknél egy téglalap alakú karaktert helyeztek, amibe bele kellene tenni a pipát. Egy pozícióra azonban két karaktert nem tudunk elhelyezni, így ennek a résznek a kitöltése barkácsolás nélkül nem oldható meg.

A választási lehetőségeknél mindkettő bejelölhető, annak ellenére, hogy szemantikai tartalmát tekintve a két-két választási lehetőség kizárja egymást. Az előzőt és ezt a szempontot is figyelembe véve két pár rádiógomb elhelyezésére lett volna szükség, amelyek párosával külön csoportba tartoznak.

A szövegbevitelt megnehezíti egyrészt, hogy valamennyi bekezdés közepe van igazítva (18. ábra felső), ezen túl pedig a rengeteg aláhúzás karaktert tartalmaz, amelyek beszúrásos beviteli módban csak tolnak a szöveg gépelésekor (18. ábra alsó).

További hátránya a normál dokumentumként szerkesztett ál űrlapoknak, hogy bármikor módosítható a szövegtest, így a felhasználó akaratlanul is belenyúlhat a szövegbe. Az ilyen típusú űrlapokkal gyűjtött adatok feldolgozása pedig egy további időigényes és kevésbé biztonságos folyamat.



19. ábra

Rádiógombok tulajdonság ablaka

MANYE-tag:¶  
 Igen  Nem ¶  
 Belépési lehetőség és tagdíjfizetés: E-mail: [nagyagi@btk.pte.hu](mailto:nagyagi@btk.pte.hu)¶

Előadás:¶  
 Előadást tartok  Előadást nem tartok ¶  
 Az előadás címe: ○○○○¶

MANYE-tag:¶  
 Igen  Nem ¶  
 Belépési lehetőség és tagdíjfizetés: E-mail: [nagyagi@btk.pte.hu](mailto:nagyagi@btk.pte.hu)¶

Előadás:¶  
 Előadást tartok  Előadást nem tartok ¶  
 Az előadás címe: Hogyan töltsék ki egy hibás űrlapot?¶

20. ábra

Űrlap objektumai (felső) (*manye\_csm\_urlap.doc* és *manye\_csm\_urlap\_vedett.doc*), védett űrlap kitöltve (alsó) (*manye\_csm\_urlap\_kitoltott.doc*)

### **2.2.2. Hibakategóriák megnevezése**

Az előző fejezetekben összegyűjtött lehetséges források alapján a leggyakrabban előforduló hibák az alábbi kategóriákba sorolhatóak. A szintaktikai és a szemantikai kategóriák ilyen elnevezését a programozási nyelvekkel való harmonizáció indokolja.

- tördelési
- formázási
- szintaktika
- szemantikai
- tipográfiai

További hiba kategóriák is léteznek, de azok inkább szöveg specifikusak, így felsorolásuktól eltekintik. Fontosnak tartom azonban megemlíteni a levelezési szabályokat (Balázs Géza 2007: 11–47), valamint a stilisztikai elvárásokat (Szikszainé: 2007), amelyek ismeretének hiánya gyakori hibaforrás, de ezek már nagyban túl mutatnak a szövegszerkesztés témakörén.

### **2.2.3. Hibafelismerés**

A hibák megkeresése, felismerése a számítástechnika oktatásban nem ismeretlen gyakorlat. A programozási nyelvek tanításánál egy gyakran alkalmazott módszer, hogy hibás programokat adunk a tanulónak, amelyeket elemezni, majd javítani kell (Dormann–Frese 1994: 365–372, Frese–Altmann: 1989, Frese tsa: 1989). A szövegszerkesztés oktatásánál a hibák keresése, a hibákkal tanítás nem más, mint egy már rég ismert módszer adaptálása ehhez az új környezethez. A módszer használata során megtanítjuk, hogy a hibák önmagukban nem rosszak, az a rossz, ha benne maradnak a dokumentum, a program végleges változatában. A hibák felismerése, elemzése, javítása mind segítenek fejleszteni az algoritmikus és a debugging készségeket (Lazar–Norcio 2003: 361–377), valamint elkerülni ezeket a hibákat saját dokumentumok, programok készítésekor.

#### **2.2.3.1. A hibafelismerés lépései és eszközei**

Kezdő felhasználóknál mindenképpen figyelni kell a fokozatosságra. Figyelembe kell venni, hogy ők még nem tudják a szövegszerkesztő programokat használni, tehát érdemes az alábbi sorrendet betartani.

- nyomtatott dokumentumban hibakeresés
- e-dokumentumban hibakeresés
- hibák javítása.

Fontos továbbá a dokumentumok megfelelő előkészítése. A hibakeresést autentikus szövegeken érdemes tanítani. Egyrészt a hitelesség miatt, másrészt a tanulók egy idő után nagyon büszkék arra, hogy ők nem követik el ugyanazokat a hibákat. A dokumentumok kiválogatásához és előkészítéséhez a 2.4. fejezetben térünk vissza.

### **2.2.3.2. 1-es típusú hibatáblázat**

A hibafelismeréshez segítséget nyújthat egy hibatáblázat. A legegyszerűbb hibatáblázatot kezdőknél érdemes alkalmazni, és akkor igazán hatékony segítség, ha a mintadokumentumon előre sorszámozzuk a hibákat (21. ábra). A tanulók beazonosítják a hibákat és megjelölik a hiba típusát a megfelelő cellában. A hibatáblázatban azt is meg lehet jelölni, hogy a hiba felismerhető-e nyomtatásban. A hibatáblázat természetesen a tanulói teljesítmények értékelésekor is használható, megfelelően előkészített szövegek esetén.

A hibatáblázat egy magasabb verziószámú változatában már a hiba megnevezését is kérjük az utolsó sorban (23. ábra). Egy még magasabb szintje a hibafelismerésnek, amikor a hibák már nincsenek megjelölve a dokumentumon, hanem a tanulóknak kell azokat megkeresni, beszámozni, megnevezni a hibát és a hibatípust. Ha alaposan meggondoljuk, ahhoz, hogy erre a szintre eljussunk, még nincs szükség a tényleges szövegszerkesztésre. A tanulók már felismerik a hibát nyomtatásban, meg tudják nevezni a hiba típusát, ellenőrizni tudják a hiba forrását a szövegszerkesztő programban. Összességében, ez a módszer remekül alkalmazható a koncepció kialakításához, ami az ismeretszintek legalsóbb szintje (IEEE&ACM Report 2013).

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
helyes																
tördelési hiba																
formázási hiba																
szintaktikai hiba																
szemantikai hiba																
tipográfiai hiba																

21. ábra

Üres hibatáblázat a *gyogyszerek.doc* szöveghez

### 2.2.3.3. 2-es típusú hibatáblázat

A következő szinten még eggyel továbblépünk, is itt már azt is kérjük, hogy hogyan lehetne a hibát kijavítani. Ez az a szintje a hibafelismerésnek, amelyik átvezet bennünket a tényleges szövegformázáshoz, tehát a koncepció kialakítását követő fázisba, a használatba (IEEE&ACM Report 2013).

### Szövegek keresése az interneten

A hibakeresésnek egy még magasabb szintje az adott feltételeket kielégítő hibás dokumentumok keresése az interneten. Erről részletesen „Az alkalmazások közötti átjárhatóság” című fejezetben olvashatnak. A hibás dokumentumok keresése már önmagában is egy hibafelismerés, de ezt követően, hasonlóan a mintafeladatokhoz, el kell végezni a dokumentum javítását és helyes formázását is. Nagyon nehéz és időigényes feladat, viszont rendkívül hatékony.

## 2.3. Szövegszerkesztés lépései (19. óra)

### 2.3.1. Szövegbevitel

A tényleges szövegszerkesztés első lépése a szöveg előállítása (Tóth: 2001). Napjainkban számos olyan lehetőség van, amellyel felgyorsíthatjuk a hagyományos gépeléssel történő szövegbevitelt. Nem szabad azonban megfélekednünk ezen lehetőségek korlátairól, a használatukhoz köthető infoetikai feltételekről, valamint arról, hogy nem az informatika óra feladata, hogy a tanulók megtanuljanak gépelni.



### 2.3.1.1. Szövegbevitel lehetséges módjai

A leggyakoribb szövegbeviteli módok a következők: gépelés, szövegrész másolása, fájl megnyitása, karakter felismerés, hangfelismerés.

#### Gépelés

A szövegbevitel leggyakoribb formája a gépelés. A billentyűzet az erre leginkább alkalmas eszköz, de a mobil eszközök virtuális billentyűzete is nehézkesen ugyan, de használható. Érdeemes megtanulni a vakon gépelést, mert ezzel felgyorsíthatjuk és biztonságosabbá tehetjük az adatbevitel folyamatát.

#### A billentyűzet írógéphez hasonlítása

Az írógépek még mindig kísértenek. Még a legújabb kiadású tankönyvekben is találunk olyan szövegrészeket, amelyekben a billentyűzetet úgy próbálják elmagyarázni a tanulóknak, hogy az írógéphez hasonlítják (Csernoch: 2014). A billentyűzet ily módon történő bevezetése azonban mára idejét vesztette és félrevezető is lehet.

- Napjaink általános iskolás tanulói nem ismerik az írógépet, így nem érdemes egy eszközt, amiről éppen tanulnak, egy másik ismeretlenhez hasonlítani. Ez további fogalomzavarhoz és indokolatlanul sok egyszerre érkező új információhoz vezethet.
- A billentyűzet írógéphez hasonlítása ellen szól az is, hogy a szövegszerkesztésben helytelen megoldások nagy része az írógéphasználatra vezethető vissza – például indokolatlan Tabulátor (Tab), Szóköz és Enter használat. Ezt a szempontot is figyelembe véve, érdemes inkább elkerülni, mint felhívni a figyelmet erre a napjainkban már nem alkalmazható gyakorlatra.
- Az írógéphez leginkább a számítógép billentyűzet alfanumerikus része hasonlítható, ez azonban csak egy része a billentyűzetnek. Ennek megfelelően a két eszköz között inkább részalmaz-halmaz kapcsolat van, mintsem egymásnak egy az egyben való megfeleltethetőség.

#### Gépelés az informatika órán

Egy erősen megkérdőjelezhető gyakorlat alakult ki az informatika órákon: a gyerekek nem fájlban kapják meg a feldolgozandó szöveget, hanem nekik kell begépelni. Ezzel a tanulókat hosszú, unalmas, időigényes gépelésre kényszerítjük. Számptalan oka van annak, hogy informatika órán

ne vagy csak végszükség esetén gépeltessük a tanulókat. A következő lista ezek közül sorol fel néhányat.

- A gépelés megtanítása nem az informatika tantárgy feladata.
- A gépeléssel töltött időt fel lehetne használni az informatika célirányos oktatására, ami a rendkívül szűkös órakeretek között mindenképpen indokolt lenne.
- Gépelés helyett fájlok megnyitása, mentése, mentése más névvel és/vagy más formátumban sokkal inkább az informatikaoktatás feladata.
- A gépelés időigényes folyamata alatt elveszti a feladat a varázsát; mire begépelik a tanulók több-kevesebb hibával az adatokat, addigra bizony rá is unnak.
- Olyan adatmennyiséget nem tudunk begépelteni, amellyel be tudjuk mutatni azt a csodát, amire egy táblázat-, egy adatbáziskezelő, egy szövegszerkesztő program képes, ami különösen általános iskolás tanulók esetében nagyon fontos lenne.
- A gépelési hibák megnehezítik az óravezetést, ahány tanuló, annyi hiba, ami teljes káoszhoz vezethet, míg egy célirányosan hibás szöveggel módszertanilag jól felépített, hatékony órát lehet tartani.

A billentyűzet helyes és tudatos használata elsősorban nem azért fontos, hogy a tanulók informatika órán gépeljenek, hanem azért, hogy felhasználóként tudatosan használják azt, meggyorsítva és hibamentessé téve munkájukat. Érdeemes továbbá megjegyezni, hogy a világ számtalan iskolarendszerében a gépelés ugyanúgy részévé vált a tananyagoknak, ahogy a hagyományos közismereti tárgyak, felismerve ennek szükségességét és fontosságát (Starr: 2005, Baum, Lavton 2013). Sajnálatos módon a mi iskolarendszerünk jelenleg egyáltalán nincs felkészülve egy ilyen jellegű tantárgyra és a tantárgy oktatására sem. Következésképpen, továbbra is maradunk, és sajnos a következő nemzedék is marad az egyujjas, sok hibát generáló, időigényes, fárasztó pötyögésnél, amit tovább erősít az okostelefonok egyujjas használata.

### **Szövegrész másolása**

Szövegrészek másolása is egy lehetséges módja a szöveg előállításának, ebben az esetben azonban mindig meg kell jelölni a forrást. A másolásnál azonban egy további hibaforrás az eredeti szöveg stílusának az

átmásolása is a tartalommal együtt. Szövegrészek átvételénél mindenképpen érdemes csak a tartalmat beilleszteni az új pozícióba.

### **Fájl megnyitása**

Teljes dokumentumok megnyitása és használata csak olvasásra javallott. Ebből egyenesen következik, hogy megnyitás után mindenképpen szükséges a fájl mentése más néven, esetleg más formátumban. Teljes dokumentumok szövegszerkesztőben történő megnyitását elsősorban az indokolja, hogy ezekben a programokban nagyon könnyen tudunk megjegyzéseket elhelyezni és korrektúrázni, amely a szöveg feldolgozását nagyban elősegíti. Ezen túl remekül alkalmazható jelen szöveggörnyezetben, ahol a tanulónak érdemes már létező, autentikus vagy a tanár által megfelelő formátumra hozott szövegeket megnyitni tanulás, gyakorlás céljára.

### **Szöveg- és karakterfelismerés**

A szöveg- és karakterfelismeréssel elsősorban még a digitális világ előtt keletkezett dokumentumok e-szöveggé alakítását végezzük. Ezen szöveg-előállítási módhoz szükségünk van egy szkennerre és egy OCR programra (Optical Character Recognition), amely ténylegesen elvégzi a karakterfelismerést és a dokumentum szöveges formátumban való mentését.

### **Hangfelismerés**

A hangfelismerés (Speech-to-text vagy Speech Recognition) bármennyire is kényelmes megoldásnak tűnik, még nagyon gyerekcipőben jár. Nagyobb nyelvekhez már készültek alacsony hibaszázalékkal működő programok, de sajnos magyar nyelvű kész programról még nincs tudomásunk.

### **2.3.2. Hibajavítás**

A hiba felismerését, azonosítását követi a hibajavítás. Minden esetben a hiba határozza meg a hibajavítás bonyolultságát. Alapvetően két különböző hibajavítási módszert különböztetünk meg: gépeléssel és a formázással elvégezhető hibajavítás.

### 2.3.2.1. Gépeléssel javítható hibák

A karakterekkel létrehozott tördelési hibák, a szintaktikai és szemantikai hibák gépeléssel javíthatók. A felismerést követően, ezek javítása a legegyszerűbb. A halmozottan barkácsolt szövegek azonban komoly fejtörést okozhatnak.

### 2.3.2.2. Formázással javítható hibák

A formázással létrehozott hibák javítása bonyolultabb. Érdeemes az ilyen típusú hibák javítását a teljes szövegen elvégzett *Formázás törlése* paranccsal kezdeni. Ez a parancs valamennyi betű- és bekezdésformázást eltávolítja a szövegből. Nem szabad azonban elfelejtenünk, hogy a parancs hatóköre a bekezdés, tehát ahhoz, hogy az egész szövegre érvényes legyen ki kell jelölni a szöveget. A hatókőről részletesen a 2.3.3.2. fejezetben lesz szó.

Minden egyéb hibás formázás eltávolítása már egy fordított formázás, ami megköveteli a szövegszerkesztő program használatát is. A szövegek megválasztása különösen fontos az ilyen típusú hibák javításánál. Csak olyan hibás szövegeket adjunk a tanulóknak, amelyekről már önállóan el tudja távolítani a hibás formázásokat, vagy amelyek az éppen tanulandó, tanítandó formázással kapcsolatos hibákat tartalmazzák. A szövegek kereséséről, kiválasztásáról a 2.4. fejezetben térünk vissza részletesen.

## 2.3.3. Formázás

### 2.3.3.1. Formázás lépései

A szövegek formázását a legnagyobb egységgel kezdjük, és haladunk a kisebbek felé. Ennek megfelelően legelőször a teljes dokumentumra vonatkozó beállításokat végezzük el, majd ezt követően bontjuk a szöveget kisebb formázható egységekre. Ez a formázási sorrend nagyban hasonlít ahhoz, amikor az általános iskolás gyerekek tanulnak fogalmazást írni, és az az első lépés, hogy vázlatot készítenek, hogy tisztán lehessen látni a szöveg szerkezetét. Az azonos szintű szövegegységek formázása mindenképpen megköveteli a stílusok használatát.

Informatika órákon, tankönyvekben gyakran tapasztalni, hogy a szövegformázást a betű és bekezdésformázásokkal kezdik, arra hivatkozva, hogy ez a könnyű. Ennek egyenes következménye, hogy olyan formázásokat tanítunk meg, amelyekre a későbbiekben nem lesz szükség, mert

többségük tipográfiai helytelen. Ezen túl, a betűformázások halmozásával elveszünk az apró részletekben, és ezzel elveszítjük a dokumentum szerkezetét. A stílusok használata mentesíthet bennünket az értelmetlen karakter- és bekezdésformázásoktól. Fontos azonban megjegyezni, hogy a stílusok létrehozásához alapos tipográfiai ismereteket szükségesek.

#### **2.3.3.2. Hatókör**

A felhasználóknak örök probléma, hogy kell-e kijelölni vagy sem. Annak eldöntésére, hogy szükség van-e kijelölésre vagy sem ismernünk kell a parancs hatókörét. A hatókör ismerete egyrészt összhangban van az előző fejezetben megfogalmazott formázási sorrenddel, másrészt, a fölösleges kijelölések elhagyásával rendkívüli módon felgyorsíthatjuk a szövegszerkesztés menetét.

Elsőként tehát szükség van a hatókör definíciójára. A hatókör az a terület, amelyre a formázás (formázási parancs) érvényes kijelölés nélkül.

A Microsoft, OpenOffice, LibreOffice szövegszerkesztőket használva a legnagyobb hatókör a teljes dokumentum, majd a nála kisebb a bekezdés és ezt követi a szó. A teljes dokumentum és a bekezdés között még elhelyezkedik a szakasz, amit azonban érdemes egy későbbi fázisban tanítani, amikor azt a problémák indokolják.

#### **2.3.4. Tévhitek**

Az alábbi felsorolásban a leggyakoribb tévhitek közül gyűjtöttem össze néhányat.

- A szövegszerkesztő program egy olyan eszköz, mellyel könnyen, gyorsan szép, esztétikus kiadványokat tudsz készíteni...
- A nem-nyomtatódó karakterek zavarják a szövegszerkesztést.
- Egyedül az számít, hogy nyomtatásban jól nézzen ki a szöveg.
- Program használatának leírása.

##### **2.3.4.1. Gyorsan, könnyen**

A szövegszerkesztő programok nem kiadványszerkesztésre készültek, hanem sokkal inkább rövidebb dokumentumok létrehozására, módosítására. Természetesen, lehet velük hosszabb kiadványokat is készíteni, de nem az az elsődleges funkciójuk, és nem is rendelkeznek olyan eszközkészlettel, amelyek a kiadványszerkesztést támogatnák.

Az, hogy mennyire lesz esztétikus a dokumentum, egyértelműen a felhasználó tipográfiai ismereteitől függ, és nem a szövegszerkesztő programok sajátossága.

A rengeteg hibás dokumentum azt bizonyítja, hogy a szövegszerkesztő programok használata nem egyszerű. Használatuk csak abban az esetben könnyű és gyors, ha valaki ismeri a szövegszerkesztés-koncepciót, és ennek ismeretében tudatosan használja a rendelkezésre álló programot. Ellenkező esetben a szövegszerkesztés nem más, mint a szakirodalomban csak barkácsolásnak (bricolage) nevezett tevékenység (Ben-Ari–Yeshno: 2006: 1336–1350, Ben-Ari: 1999), ami alatt a Mekk Elek féle barkácsolást értjük. Sok-sok kellemetlenséget okozva mind a szerzőnek, mind azoknak, akik a későbbiekben használják a dokumentumot, akár nyomtatott, akár e-fórmátumban. A szövegszerkesztés is akkor lehet hatékony, mint bármilyen más számítógépes tevékenység, ha a koncepció és algoritmus-alapú (Carroll–Mack 1984: 13–51, Csernoch–Biró: 2015, Nielsen: 1993). Mindaddig, amíg nem használjuk ki valamennyi számítógépes problémamegoldás során a mély megközelítésű módszerek hatását a számítógépes gondolkodás, az algoritmikus készség fejlesztésére, addig továbbra is hátrányban leszünk azon országokkal szemben, akik ezt már felismerték és alkalmazzák. A PISA2009 Students On Line felmérés egyértelműen bizonyítja lemaradásunkat (Kritikai érzékkel és társadalmi felelősséggel a PISA-eredményekről, OECD: 2011). A csatolmányban megadott mintadokumentumokban, kivétel nélkül mindben felfedezhető kisebb, nagyobb mértékű barkácsolás. Mindenképpen említést érdemelnek azonban az alábbi dokumentumok:

- lanourriture.doc, rally.doc, maupassant.doc és pozvanka.doc – A dokumentumok csak és kizárólag szóközt, entert és tabulátort használnak formázások helyett. Az elsőnek érdekessége még, hogy be van állítva három balra tabulátor, de ezek teljesen random módon kerülnek felhasználásra, míg a másodikban a behúzások vándorolnak random módon, a harmadik pedig a bántó színes pacákkal még tovább rontja a helyzetet. A negyedik dokumentumban érthetetlen, hogy miért volt szükség a képek hátraküldésére, a szóközök mögé.
- csapatverseny.doc – A dokumentum jellegzetessége az ismétlődő elemeknél a stílusok teljes hiánya. Ennek következtében olyan hatást kelt a dokumentum, mintha az kalákában készült volna több Mekk Elek közreműködésével.

- animeaux.doc – A táblázatok és a táblázaton belüli szövegek, objektumok barkácsolására egy klasszikus példa.
- le\_carving\_word.doc – A dokumentum a képek, képfeliratok hibás elhelyezésére egy rémisztő példa. Annyira rossz, hogy még kijavítani is alig lehet, egy igazi kihívás.
- oceania.doc – Példa arra, hogy hogyan lehet a képekkel, szövegdobozokkal profi módon barkácsolni, nulla szövegszerkesztői ismerettel. Ez a dokumentum a tipikus példa arra, hogy készítője azt tanulta meg, hogy hogyan kell a programot kezelni és nem azt, hogy hogyan kell szöveget kezelni. A koncepció teljes hiányára példa.
- pitagfeladatok.doc – Megismétli a szokásos tördelési hibákat, a stílusok használatának hiányát. További érdekessége a dokumentumnak, hogy a képleteket nem a szövegszerkesztőben hozták létre, hanem képként szúrták be valamennyit, ami azt jelenti, hogy a képleteket egy másik szoftverben barkácsolták meg.
- swot.doc – Hasonlóan az előző fájl képleteihez, klasszikus barkácsolási trükknek számít, amikor a táblázat kereteit megrajzolják egy másik programban és képként szúróják be a dokumentumba. Ez a megoldás még tovább rontja helyzetet azzal, hogy a képet belerakja egy rosszul elhelyezett szövegdobozba.
- edesanya.doc – Az egyik valaha látott legrosszabb dokumentum. Minden létező barkácsolásra és csalásra mutat példát, de felsorakoztat teljesen egyedi megoldásokat is.
- Hasonló minőségű egy matematika – informatikus könyvtáros szakos ötödéves tanárjelölt zárótanítási óraterve (zarotanitasi\_kesz.doc).

Arra azért mindenképpen felhívnom a figyelmet, hogy ezek a dokumentumok tanítási órán csak rendkívüli körülményekkel használhatóak, mivel olyan rengeteg hibát tartalmaznak, hogy a javítások profi felhasználót igényelnek.

Mindenképpen fontosnak tartom azt is megjegyezni, ahogy a minták is mutatják, hogy a hibák nyelvtől, régiótól függetlenek, tehát nem magyar specialitással állunk szembe.

#### **2.3.4.2. Nem-nyomtató karakterek**

A nem-nyomtató karakterek nem zavarják a szövegszerkesztést, mivel ezek ugyanúgy részei a szövegnek, mint bármelyik másik karakter. Az állítással szemben, éppen, hogy segítik a szövegszerkesztést. Bármelyik

nem-nyomtatódó karakter ugyanúgy része egy dokumentumnak, mint azok, amelyek nyomtatásra kerülnek. A nem-nyomtatódó karakterek elrejtése egyenértékű bármelyik más karakter elrejtésével. Lényegesen megnehezítené a szöveg létrehozását például egy *a* betű következetes elrejtése.

### **2.3.4.3. Nyomtatott forma**

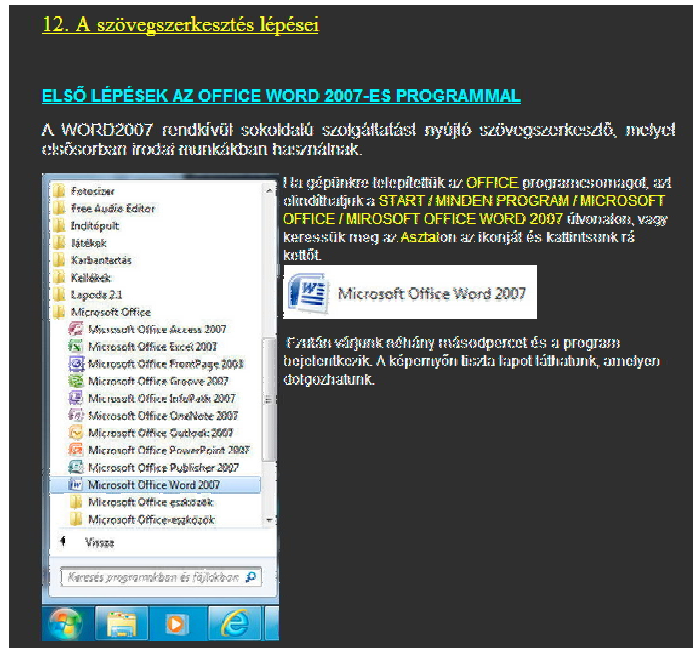
A szövegek módosíthatóságáról elmondottak alapján, nem csak az számít, hogy nyomtatásban hogyan néz ki a szöveg, az is fontos, hogy egy szövegszerkesztővel készített szöveg bármikor módosítható legyen, figyelembe véve a helyesen szerkesztett szöveg definícióját. Ezen túl, a Gyakorlatok fejezetben bemutatott minták szinte kivétel nélkül tartalmazznak olyan hibákat, amelyek, vagy amelyek következményei már nyomtatásban is látszanak.

### **2.3.4.4. Programhasználat**

Sok esetben tapasztalni, hogy a szövegszerkesztést és a szövegszerkesztő programok használatát szinonimaként kezelik (22. ábra). Ez azonban nem így van. Míg a szövegszerkesztő programok használata jellemzően felületi, addig a szövegkezelés koncepció és algoritmus-alapú megközelítés (Csernoch–Biró: 2015). Az 22. ábra mintája példa egy tipográfiai helytelen szövegre, amely további terminológiai hibákat is tartalmaz.

- „Kattintsunk az ikonra kettőt.” A kettő kattintásra nem fog a program elindulni, mivel a kettő kattintással az átnevezés végezhető el.
- A program nem bejelentkezik, hanem elindul.
- A képernyőn nem tiszta lapot láthatunk, hanem egy üres dokumentumot, amely tartalmaz egy üres bekezdést.
- A dokumentumban folyamatosan keverednek a szövegszerkesztés lépései és a program használatának leírása, még keveredik bele billentyűhasználat is.
- A dokumentum helyesen tartalmazza a szövegszerkesztés lépéseit, de éppen a lényeg vész el, a nem ide tartozó részletek és a tipográfiai következetlenségek miatt.
- A teljes dokumentumra, és nemcsak a bemutatott mintára igaz, hogy tele van terminológiai és tipográfiai hibákkal.





## 22. ábra

*A lap szerzője összekeveri a szövegszerkesztés és a szövegszerkesztő programok használatát, ezen túl számtalan hibát tartalmaz a weblap<sup>7</sup>*

### 2.4. Hibafelismerés, hibajavítás (20. óra)

Az eredeti fájlokat megtartva a mintadokumentumok letölthetők a tananyag mellékleteként megadott könyvtárból.

<sup>7</sup> A szövegszerkesztés lépései. <http://users.atw.hu/infoteszt/tananyag/6-osztaly/szovegszerk/12.htm>. Letöltve: 2014. 07. 18.

### 2.4.1. Gyógyszerek

Eredeti fájlnev: *gyogyszerek.doc*

#### 2.4.1.1. Hibafelismerés

A *gyogyszerek.doc* tördelési, tipográfiai és szintaktikai hibákat tartalmaz, valamint hiányoznak olyan formázások, amelyekkel a tervezett megjelenítés elérhető. A dokumentum sorszámokkal megjelölt hibáit mutatja az 24. ábra, míg a kitöltött hibatáblázatát az 23. ábra. A szürkével megjelölt sorszámok azokra a hibákra utalnak, amelyek már nyomtatásban is láthatók. A sorszámok véletlenszerűen vannak lerakva, a hibák típusától és a felismerhetőségétől függetlenül. Ezzel a hibaelrendezéssel és sorszámmozással megvalósítható a tetszőleges sorrendű hibafelismerés.

A tipográfiai (#1, #6) és a szintaktikai (#5, #7, #10, #11, #12, #14, #16), valamint a szemantikai<sup>8</sup> hibák már nyomtatásban is felismerhetők. A tördelési (#2,#3, #4, #8,#9, #15) és a formázási hibák (#13) felismeréséhez mindenképpen szükséges a dokumentum megnyitása.

#### 2.4.1.2. Hibajavítás menete

Fájlnev: *gyogyszerek\_javitott.doc*

A hibajavítást érdemes a betű- és bekezdésformázások törlésével kezdeni, mivel ezek a főleg formázások csak megnehezítik a hibajavítást, az pedig, hogy az egész szöveg dőlt még tovább ront a helyzeten, mert nagyon nehéz a kurzor pozícióját beazonosítani (teljes szöveg kijelölése → Formázás törlése). A formázás törlésével megszüntettük az #1, #6 és #13 hibákat.

A szintaktikai hibák javítása is viszonylag egyszerű, érdemes azokkal folytatni (#5, #7, #10, #11, #12, #14). A szintaktikai hibák javítása remek alkalom arra, hogy megtanítsuk, hogy hogyan lehet billentyűzettel gyorsan navigálni a szövegben, valamint a különböző törlési és javítási lehetőségeket. Az ilyen típusú szövegek sokkal inkább alkalmasak a billentyűzet használat megtanítására, mint a vég nélküli gépelések.

A tördelési hibák javítását érdemes a számozással létrehozott csoportok alapján elvégezni. A legegyszerűbb eltávolítani a #3 és #15 hibákat, hiszen itt semmi mást nem kell tenni, csak kitörölni a főleg bekezdés

---

<sup>8</sup> Tartalmaz a dokumentum szemantikai hibákat is, de ezek ismertetésére most nem kerül sor.

vége karaktereket (¶) és a szóközöket, amit egy-egy ésszerű kijelöléssel fel lehet gyorsítani.<sup>9</sup>

A #4 hiba törlését érdemes blokkonként végezni. A blokkok kijelöléséhez az Alt segédbillentyűt érdemes használni. A #9 hibában szereplő szóközök törléséhez érdemes egy szókjelölést végezni (dupla kattintás).

A #2 hiba törlése már több figyelmet igényel, mivel azt is nézni kell, hogy a ¶ törlése után szükség van-e szóközpótlásra vagy sem.

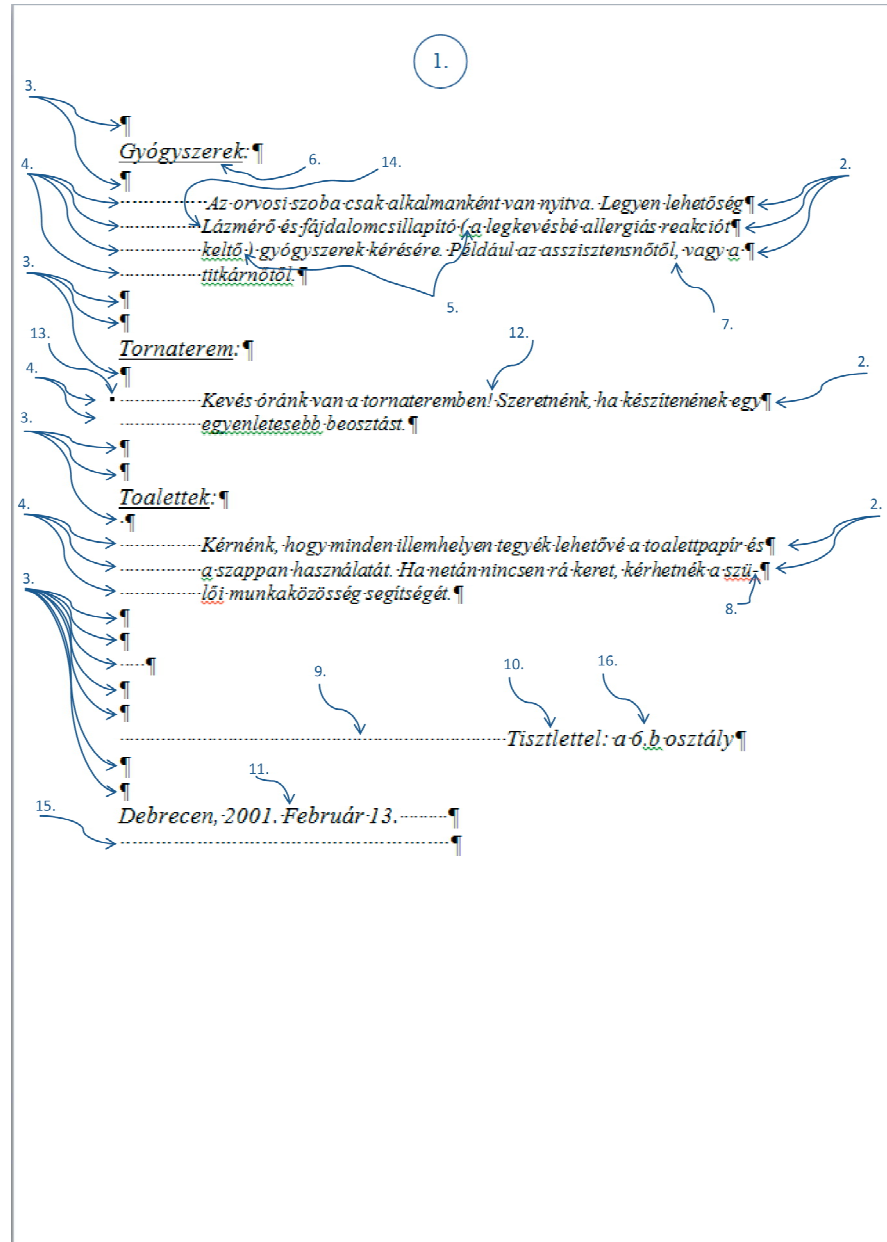
A #8 hibánál ki kell törölni a kötőjelet, mert továbbra is két szónak tekinteni a két szótöredéket.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
helyes																
tördelési hiba		×	×	×				×	×						×	
formázási hiba													×			
szintaktikai hiba					×		×			×	×	×		×		×
szemantikai hiba																
tipográfiai hiba	×					×										
hiba megnevezése	az egész szöveg dőlt	sorvégi enterek	üres bekezdések (enterek)	főlősleges szóközök, behúzás helyett	zárójelen belüli főlősleges szóköz	aláhúzás	hibás vessző	kézi elválasztás	főlősleges szóközök igazítás helyett	hiányzik egy karakter	hónap nagybetűvel	hibás felkiáltójel	főlősleges stílus	szókezdő nagybetű	főlősleges bekezdés főlősleges szóközökkel	hiányzó szóköz

### 23. ábra

*A gyogyszerek.doc minta alapján kitöltött hibatáblázata (24. ábra). A szürke mezők mutatják a nyomtatásban is felismerhető hibákat*

<sup>9</sup> Egy sor kijelölése: kijelölő sávon egy kattintás. Többsoros kijelölés: húzással vagy a Shift segédbillentyű a kurzormozgató nyilakkal.



24. ábra

A gyógyszerek.doc a hibák megjelölésével

**Gyógyszerek:**

Az orvosi szoba csak alkalmanként van nyitva. Legyen lehetőség láz mérő és fájdalomcsillapító (a legkevésbé allergiás reakciót keltő) gyógyszerek keresésére. Például az asszisztensnőtől vagy a titkárnőtől.

**Tornaterem:**

Kévszóránk van a tornateremben. Szeretnénk, ha készítenének egy egyenletesebb beosztást.

**Toalették:**

Kérnénk, hogy minden illemhelyen tegyék lehetővé a toalettpapír és a szappanhasználatát.

Ha netán nincsen rá keret, kérhetnék a szállói munkaközösség segítségét.

Tisztelettel: a 6. osztály

Debrecen, 2001. február 13.

**25. ábra**

*A gyógyszerek\_javitott.doc végső formája*

**2.4.1.3. Formázás**

Fájlnév: *gyogyszerek\_formazott.doc*

A formázás során a legnagyobb egységtől haladunk a legkisebb felé.

**Teljes szövegre vonatkozó formázások**

- papírméret és margó beállítása (hatókör: teljes dokumentum)
- automatikus elválasztás (hatókör: teljes dokumentum) (Lap elrendezés → Elválasztás → Automatikus)
- igazítás: sorkizárt (hatókör: bekezdés)

**Bekezdésformázások**

Még a formázások megkezdése előtt érdemes eldönteni, hogy indokolt-e stílusok létrehozása vagy elegendő a formátummásolás. A *gyogyszerek.doc* egy rövid dokumentum, így formátummásolással megoldható a meggyező formátumú bekezdések egységesítése (*Kezdőlap* → *Formátummásoló*).

- bekezdések térköze a bekezdések előtt és után (hatókör: bekezdés), formátummásolás
- bekezdések 2 cm-es bal oldali behúzása (hatókör: bekezdés), formátummásolás
- „Tisztelettel...” bekezdés jobbra igazítása (hatókör: bekezdés)

**Karakterformázások**

A szöveg nem igényel további formázásokat.

### 2.4.2. NLP felhívás

Eredeti fájlnev: *NLP\_felhivas.doc*

#### 2.4.2.1. Hibafelismerés

Az *NLP\_felhivas.doc* egyrészt tartalmazza *gyogyszerek.doc* dokumentum hibáit (#2, #3, #4, #6, #7, #9, #10, #13, #19, #22), de megjelennek újak is. A tördelési hibáknál újdonság a kézi számozás (#8) és az aláírás barkácsolása (#5). Találni továbbá a dokumentumban két formázási hibát, amely közül az egyik remekül szemlélteti azt a fogalomzavart, amely a margó és a behúzás körül alakult ki (#1), míg a másik a nyelvi beállítások egyik gyakori hibája (#23). Mindezeket túl a dokumentum sok-sok szintaktikai hibát is tartalmaz (#11, #12, #14, #15, #16, #17, #18). A #20 hiba elfogadható, de ebben az esetben szerencsésebb sortörés alkalmazása. A #21 hiba a csupa nagybetűvel írás, ami teljesen indokolatlan és tipográfiai helytelen.

#### 2.4.2.2. Hibajavítás

Fájlnev: *NLP\_felhivas\_javitott.doc*

A hibajavítást ebben az esetben is érdemes a karakter- és bekezdésformázások törlésével kezdeni, így sokkal áttekinthetőbb lesz a dokumentum, jobban látható a kurzor pozíciója. A formázás törlésével megszüntettük az #1, #10, #13, #19 hibákat.

A szintaktikai hibák javítása a szokásos módon történik törlésekkel és felülírásokkal, attól függően, hogy melyik a gyorsabb, egyszerűbb (#11, #12, #14, #15, #16, #17, #18).

A tördelési hibákat érdemes a számozással létrehozott csoportok alapján megszüntetni. A főleges, üres enterek törlésével célszerű kezdeni (#2, #6). Az #5 és #7 típusú szóközök legegyszerűbben egy szó kijelölést követő törléssel távolíthatók el. A kézi számozást (#8) ki kell törölni – a számot a pont, a per és a szóköz karakterekkel együtt, majd ezt követően a kézi számozáshoz kapcsolódó #4 hiba szóközeit is. Ez a törlés legegyszerűbben egy oszlop kijelöléssel végezhető el – ebben az esetben a kijelölés segédbillentyűje az Alt. A sorvégi enterek (#3) törlése következhet. A szövegben még maradtak felesleges szóközök, ezeket is törölni kell (#4, #22, #24). A #20 hiba entereit helyettesíthetjük sortöréssel. A #21 és #23 hibák javítását formázással fogjuk elvégezni.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	
helyes																	
tördelési hiba		×	×	×	×	×	×	×	×								
formázási hiba	×												×				
szintaktikai hiba											×	×		×	×	×	
szemantikai hiba																	
tipográfiai hiba										×							
hiba megnevezése	bal oldali behúzás az egész dokumentumon	üres bekezdések (enterek)	sorvégi hibás enterek	sorvégi fölösleges szóközök	szóköz igazítás helyett üres bekezdés szóközökkel és enterral	szóköz igazítás helyett	bekezdések kézi számozása	kézi elválasztás	aláhúzás	hiányzó szóköz	hibás óra-perc elválasztó karakter	Címsor1 stílus	díj	hibás ezresválasztó karakter	a szám és Ft közé egyetlen nem törhető szóköz kell		

## 26. ábra

Az *NLP\_felhivas.doc* minta alapján kitöltött hibatablázata (28. ábra).

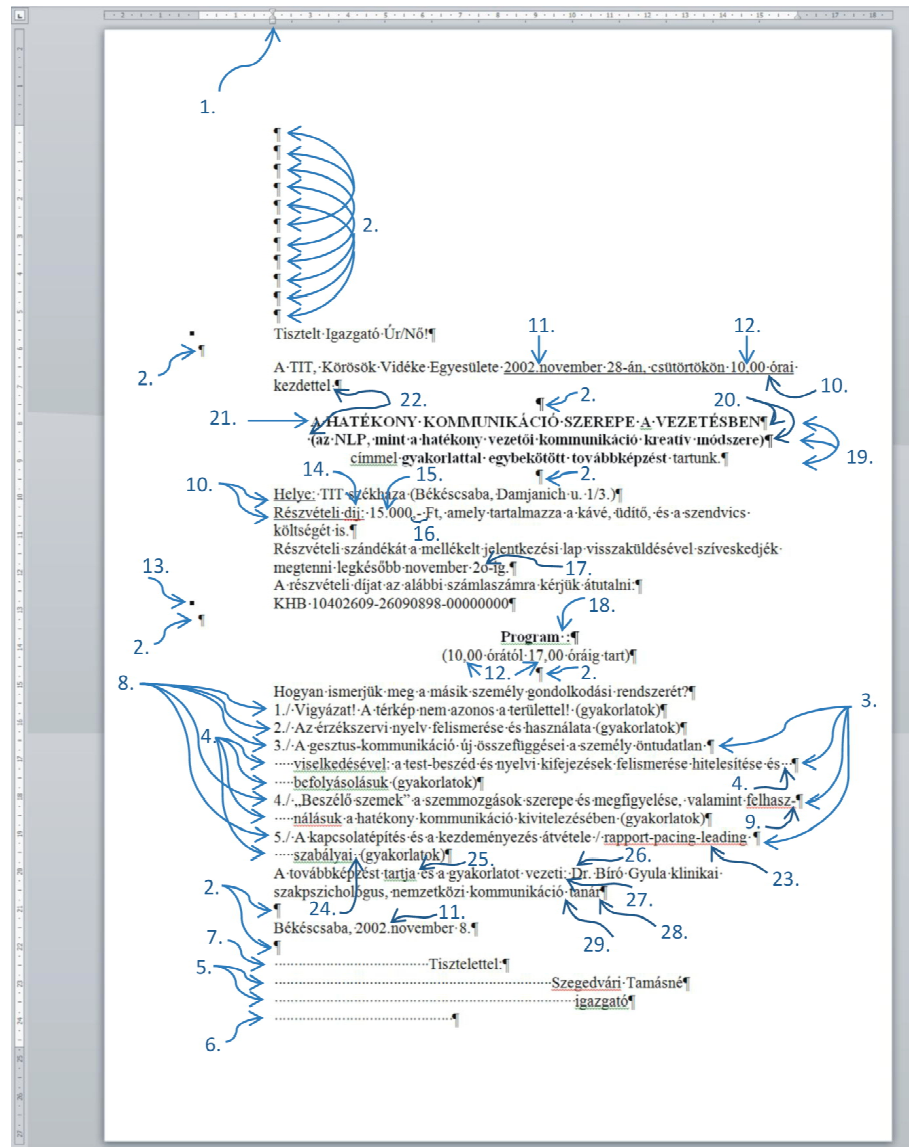
A szürke mezők mutatják a nyomtatásban is felismerhető hibákat

	17.	18.	19.	20.	21.	22.	23.	24.	25.	26.	27.	28.	29.			
helyes																
formázási hiba				×		×										
szintaktikai hiba	×	×						×	×	×	×	×	×			
szemantikai hiba																
tipográfiai hiba	×		×		×			×								
hiba megnevezése	2o	fölsleges szóköz	hibás kiemelés félkövérrrel	hibás bekezdés vége, sortörés helyett	hibás kiemelés nagybetűkkel	fölsleges szóközök	hibás nyelv beállítás	dupla szóközök	hiányzik a mondat végi írásjel	dr. (kisbetűvel kell írni)	és előtt hiányzik a vessző	fölsleges vessző az és előtt	hiányzik a névelő a az idéző elől			

## 27. ábra

*Az NLP\_felhivas.doc minta alapján kitöltött hibatáblázata (28. ábra).  
A szürke mezők mutatják a nyomtatásban is felismerhető hibákat*





28. ábra  
Az NLP\_felhivas.doc a hibák megjelölésével

Tisztelt Igazgató Úr/Nő!

A TIT, Kőrösök Vidéke Egyesülete 2002. november 28-án, csütörtökön 10.00 órai kezdettel

A HATÉKONY KOMMUNIKÁCIÓ SZEREPE A VEZETÉSBEN

(az NLP, mint a hatékony vezetői kommunikáció kreatív módszere)

címmel gyakorlattal egybekötött továbbképzést tartunk.

Helye: TIT székháza (Békéscsaba, Damjanich u. 1/3.)

Részvételi díj: 15 000 Ft, amely tartalmazza a kávé, az üdítő és a szendvics költségét is.

Részvételi szándékát a mellékelt jelentkezési lap visszaküldésével szíveskedjék megtenni legkésőbb november 20-ig.

A részvételi díjat az alábbi számlaszámra kérjük átutalni:

KHB:10402609-26090898-00000000

Program:

(10.00 órától 17.00 óráig tart)

Hogyan ismerjük meg a másik személy gondolkodási rendszerét?

Vigyázat! A térkép nem azonos a területtel! (gyakorlatok)

Az érzékszervi nyelv felismerése és használata (gyakorlatok)

A gesztus-kommunikáció összefüggései a személy öntudatlan viselkedésével: a test-beszéd és nyelvi kifejezések felismerése, hitelesítése és befolyásolásuk (gyakorlatok)

„Beszélő szemek” a szemmozgások szerepe és megfigyelése, valamint felhasználásuk a hatékony kommunikáció kivitelezésében (gyakorlatok)

A kapcsolatépítés és a kezdeményezés átvétele / rapport-pacing-leading szabályai (gyakorlatok)

A továbbképzést tartja, és a gyakorlatot vezeti dr. Bíró Gyula klinikai szakpszichológus, nemzetközi kommunikációtanár.

Békéscsaba, 2002. november 8.

Tisztelettel:

Szegedvári Tamásné

igazgató

### 29. ábra

*Az NLP\_felhivas\_javitott.doc végső formája*

#### 2.4.2.3. Formázás

Fájlnév: *NLP\_felhivas\_formazott.doc*

#### Teljes szövegre vonatkozó formázások

- papírméret és margó beállítása (hatókör: teljes dokumentum)
- automatikus elválasztás (hatókör: teljes dokumentum)
- igazítás: sorkizárt (hatókör: bekezdés)

#### Bekezdésformázások

Érdemes a térközök elhelyezésével kezdeni a formázást, majd ezt követően a középre igazításokat elvégezni. Ebben a dokumentumban egyik formázáshoz sem szükséges kijelölés, mivel mind a térköz, mind az igazítás parancsok hatóköre a bekezdés. A formátummásolással gyorsítható a munka, mivel a dokumentum tartalmaz megegyező formátumú bekezdéseket.

A bekezdések számozásához az automatikus számozást kell használni, így bármikor módosítható lesz a szöveg. Mivel a számozás hatóköre a bekezdés, ezért a Számozás parancs kiadása előtt ki kell jelölni a bekezdéseket<sup>10</sup> (Kezdőlap → Számozás)!

Az aláírás is bekezdésformázás, de annak részleteit a 2.4.2.4. fejezet tartalmazza.

### **Karakterformázások**

A szöveg tartalmaz két olyan hibát, amely nem javítható a Formázás törlése paranccsal. A hatóköröket figyelembe véve mindkét formázás esetén szükség lesz pontos kijelölésre, amely valamennyi karaktert magába foglalja.

- nagybetűs szöveg cseréje mondatkezdőre (Kezdőlap → Kisbetű – nagybetű → Mondatkezdő nagybetűk),
- majd ennek formázása kiskapitálisra, ha feltétlenül ragaszkodunk a nagybetűs írásmódhoz (Kezdőlap → Betűtípus párbeszédablak → Betűtípus lap → Kiskapitális),
- angol kifejezések nyelvének átállítása angolra (Állapotsor → Nyelv → angol vagy Korrektúra → Nyelv → Ellenőrzés nyelvének megadása → angol).

#### **2.4.2.4. Aláírás formázása**

Az aláírások elhelyezése egy gyakori probléma, alig találni olyan dokumentumot, amelyben helyesen van az aláírás formázva, annak ellenére, hogy több megoldás is létezik. Ebben a fejezetben három lehetséges megoldást mutatunk be. Mindhárom megoldás esetén a már formázott dokumentumot használjuk kiindulásként, és csak a legutolsó két bekezdés formázása lesz eltérő. A három eltérő megoldást egy-egy külön fájlba érdemes menteni.

Az aláírás formázásához használhatjuk a következő módszereket:

- középre tabulátor (NLP\_felhivas\_formazott\_behuzas.doc)
- bal oldali behúzás és középre igazítás (NLP\_felhivas\_formazott\_tabulator.doc)

<sup>10</sup> Bekezdés hatókörű parancsok esetén több bekezdés kijelöléséhez elegendő a bekezdések egyetlen karakterének a kijelölésbe foglalása. Nem kell azon egyensúlyozni, hogy az összes bekezdés, összes karaktere bekerüljön a kijelölésbe.

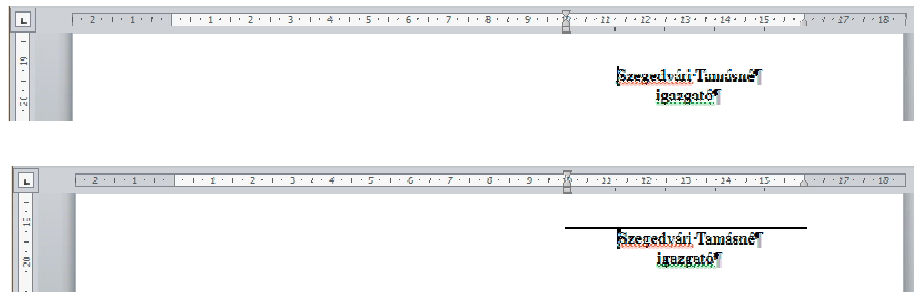
- egy oszlopos és két soros táblázat, a táblázat jobbra igazítva és méretezve, a cellák tartalma középre igazított (NLP\_felhivas\_formazott\_tablázat.doc)

### Aláírás behúzással

Mentsük el az *NLP\_felhivas\_formazott.doc* dokumentumot az *NLP\_felhivas\_formazott\_behuzas.doc* új névvel!

A három megoldás közül ez a legegyszerűbb, de hátránya, hogy csak akkor alkalmazható, ha egyetlen aláírást tartalmaz a dokumentum.

Mivel mind a behúzás, mind a középre igazítás parancsok hatóköre bekezdés, ezért mindkét bekezdést ki kell jelölni. Végezzünk el egy 10 cm-es bal oldali behúzást, majd igazítsuk a bekezdéseket középre! Amennyiben vonalat is szeretnénk az aláíráshoz adni, akkor a kijelölés megszüntetése után, a név bekezdéséhez adunk egy felső szegélyt is (30. ábra)!



**30. ábra**

*Aláírás behúzással*

### Aláírás középre tabulátorral

Nyissuk meg újra az *NLP\_felhivas\_formazott.doc* dokumentumot, majd mentsük az *NLP\_felhivas\_formazott\_tabulator.doc* névvel!

A tabulátor parancs hatóköre is bekezdés, ezért mindkét bekezdést ki kell jelölni. Az aláírások tabulátorhoz igazítása két lépésből áll:

- a tabulátor létrehozása (elhelyezése), majd a
- a tabulátor használata.

A tabulátorok elhelyezhetők a vonalzó és a Tabulátorok párbeszédablakon is. Ha nem akarunk vonalat húzni, akkor érdemes az egyszerűbb vonalzó megoldást választani.

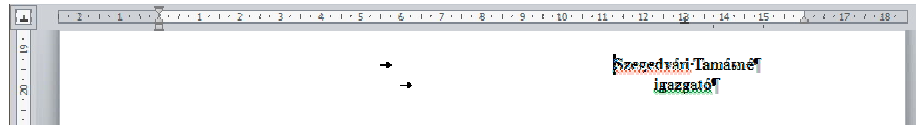
**Aláírás középre tabulátorral vonal nélkül**

Tabulátor létrehozása vonalzón

- típus kiválasztása (vízszintes vonalzótól balra, nevenincs gomb → tabulátor középre)
- hely megadása (kattintás a vonalzón, a 31. ábra mintájának megfelelően 13 cm)

Tabulátor használata

- a kijelölést meg kell szüntetni,
- a kurzort a név elé helyezzük (a névtől balra) → Tab billentyű megnyomása
- a kurzort a beosztás elé helyezzük (a pozíciótól balra) → Tab billentyű megnyomása



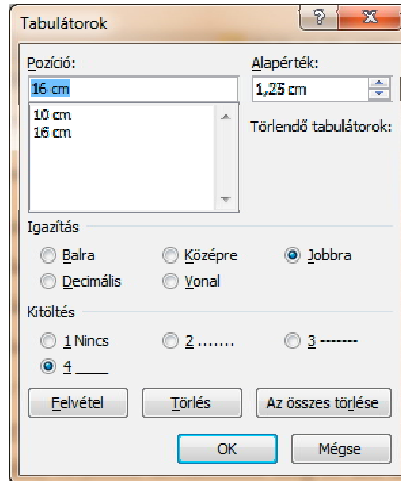
**31. ábra**

*Aláírás középre tabulátorral*

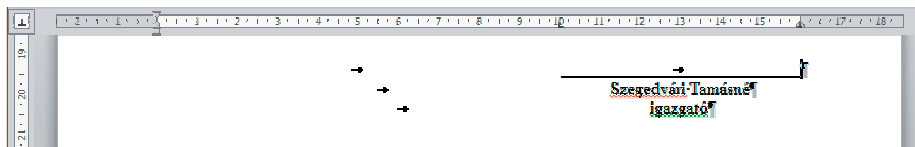
Gyakori formázási hiba, hogy az aláírásnál nem középre típusú tabulátort használnak, hanem két eltérő pozíciójú balra tabulátort a két bekezdésben. Ez a megoldás ugyanolyan hibás, mint a szóközökkel és az alapértelmezett tabulátorokkal barkácsolt aláírások, mivel nem felel meg a helyesen formázott szöveg definíciójának, hiszen újraformázás nélkül (tabulátor pozíció módosítása) nem módosítható sem a név, sem a beosztás.

**Aláírás középre tabulátorral vonal rajzolásával**

A vonal megrajzolása tabulátorral már egy kicsit trükkösebb, mivel úgy kell elhelyezni a vonalat egy önálló bekezdésben a név fölött, hogy annak kezdete és vége ugyanolyan távolságra legyen a középre tabulátor pozíciójától. Ezt minden gond nélkül meg lehet csinálni a vonalzón. A vonal megrajzolásához azonban szükség van a párbeszédablakra, így már érdemes a két tabulátort is ott létrehozni. A tabulátorokat érdemes 10 és 16 cm-re pozícionálni, míg típusuk balra és jobbra (mivel szöveg nem kerül a bekezdésbe, ezért a típusnak nincs igazán fontos szerepe, de ez a precíz megoldás). A jobbra tabulátorhoz még be kell állítani egy kitöltést is.



**32. ábra**  
*Tabulátorral létrehozott aláíráshoz vonal rajzolása,  
a szükséges tabulátorok beállítása párbeszédablakon*



**33. ábra**  
*Tabulátorral létrehozott aláíráshoz vonal rajzolása*



34. ábra

Tabulátorral létrehozott két aláírás megnövelt függőleges távolsággal a vonal és a név között. A felső minta a név és a beosztás, míg az alsó a vonalak bekezdésének vonalzóját mutatja

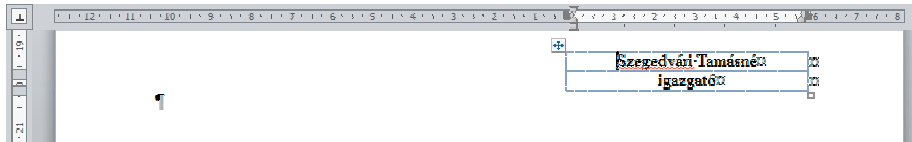
### Aláírás táblázattal

Nyissuk meg újra az *NLP\_felhivas\_formazott.doc* dokumentumot, majd mentjük az *NLP\_felhivas\_formazott\_tablázat.doc* névvel!

Aláírás táblázattal vonal nélkül

Az aláírás, beosztás bekezdéseket átalakítjuk egy 1 oszlopos és 2 soros táblázattá (bekezdések kijelölése → *Beszűrés* → *Táblázat* → *Szövegből táblázat*). Ezzel készen van a táblázat, de még további formázásokra lesz szükség.

- szegély eltávolítása (Kezdőlap → Bekezdés → Szegély és mintázat → Nincs szegély)
- táblázat szélességének módosítása (Táblázateszközök → Elrendezés → Tulajdonságok → Táblázat lap → Kívánt szélesség → 6 cm)
- táblázat pozíciója (Táblázateszközök → Elrendezés → Tulajdonságok → Táblázat lap → Igazítás → Jobbra)
- cellák tartalmának középre igazítása (kijelölés → Kezdőlap → Középre zárás)



### 35. ábra

*Táblázattal létrehozott aláírás vonal nélkül*

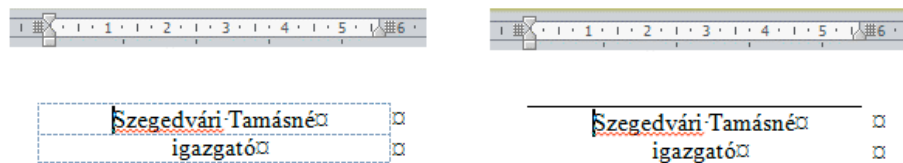
Ez a módszer használható több aláírás egymás mellé helyezésére is.

#### **Aláírás táblázattal vonal rajzolásával**

A táblázat létrehozása ugyanúgy történik, ahogy a vonal nélküli verzióban.

A vonal megrajzolásához szükségünk lesz a szegélyek párbeszédablakára. A név cellája kell legyen az aktuális (*Kezdőlap* → *Bekezdés* → *Szegély és mintázat menü* → *Szegély és mintázat párbeszédablak* → *Szegély lap* → *Hatókör: Cella vagy Bekezdés* → *Típus: Egyéni* → *felső szegély*).

Ahhoz, hogy vonalat is lássuk, érdemes kikapcsolni a Rácsvonalak megjelenítését (*Táblázateszközök* → *Elrendezés* → *Rácsvonalak megjelenítése*).



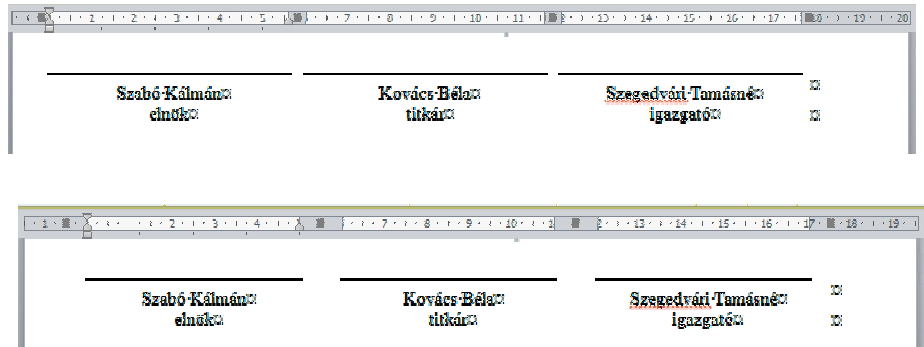
### 36. ábra

*Táblázattal létrehozott aláírás vonallal rácsvonalakkal (bal) és kikapcsolt rácsvonalakkal (jobb)*

A módszer használható több aláírás egymás mellé helyezésére is, abban az esetben azonban, amikor vonalakat is szeretnénk egymás mellé rajzolni nagyon fontos a hatókör Bekezdésre beállítása. Ezzel a megoldással tudjuk megoldani, hogy a vonalak között legyen valamennyi rés. A rés szélessége a cella margók szélességével egyezik meg, ami alapértelmezés szerint 0,19 cm, de bármikor átállítható (*Táblázateszközök* → *El-*



rendezés → Tulajdonságok → Beállítások → Alapértelmezett cellamar-  
gók → Bal és/vagy Jobb).



**37. ábra**

Táblázattal létrehozott három aláírás vonallal, megnövelt függőleges távolsággal a vonal és a név között, valamint megnövelt vízszintes távolsággal a vonalak között (alsó)

### Vonalak függőleges igazítása

A vonal és név közötti távolságot a legegyszerűbben a tabulátoros megoldásnál tudunk beállítani. Ennél a megoldásnál a vonal és a név külön bekezdésekben van, tehát semmi más teendő nincs, mint a két bekezdés közé térközt állítunk be (34. ábra).

A behúzásnál a szegély távolságát kell beállítani (*Kezdőlap* → *Bekezdés* → *Szegély és mintázat menü* → *Szegély és mintázat párbeszédablak* → *Beállítások* → *Szegély és mintázat beállításai párbeszédablak* → *Távolság: Fent!*)

Táblázat használatánál ugyanúgy járunk el, mint a behúzás esetén, de itt figyelni kell, hogy a hatókör *Bekezdésre* legyen állítva.

### 2.4.3. Mozirajongók

Eredeti fájlnev: *mozirajongok.rtf*

Az első mentéskor érdemes a fájl formátumot megváltoztatni, hogy a továbbiakban a szövegszerkesztő program saját formátumával dolgozzunk: *mozirajongok.doc*

#### 2.4.3.1. Hibafelismerés

Fájlnév: *mozirajongok\_javitott.doc*

Mind a hibákkal beszámozott minta, mind a hibarács egyértelműen mutatja, hogy bőven tartalmaz a dokumentum hibákat. A dokumentum érdekessége a hibás margó beállítás, és ennek következménye, hogy a nyomtató által nem látható területen lemaradtak karakterek, nem kerültek nyomtatásra. Ez a terület minden nyomtató esetén létezik, nagysága a nyomtató tulajdonságaitól függ, de mindenképpen tudni a létezéséről, mert ez azt jelenti, hogy a papír nem használható teljes egészében nyomtatásra. Leggyakrabban a túlságosan a lap aljára helyezett oldalszámnál találkozunk ezzel a problémával, de a *mozirajongok.rtf* fájlban bemutatott eset sem ritka. A hibás margó beállítás kompenzálására, a teljes dokumentumra érvényes behúzást találunk a dokumentum bal oldalán. Erre a hibára már láttunk példát az *LNP\_felhivas.doc* dokumentumban. A dokumentum tartalmazza a szokásos tördelési, tipográfiai és szintaktikai hibákat is, valamint bőven van benne szemantikai, stilisztikai hiba is. Ezen két utóbbi hibatípus javításától eltekintünk, mivel nem kapcsolódik szervesen az informatikaoktatáshoz.

A *mozirajongok.rtf* abban tér el a *gyogyszerek.doc* és az *NLP\_felhivas.doc* dokumentumoktól, hogy az élőfejben is van szöveg. Mivel az élőfej a dokumentumnak egy, a szövegtükörtől teljesen elhatárolt része, mintha egy kis önálló dokumentum lenne, ezért a szövegtükörben elvégzett módosítások nincsenek rá hatással. Ennek következménye, hogy az élőfej területére belépve a teljes módosítási és formázási sorozatot meg kell ismételni.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
helyes																
tördelési hiba				X	X	X						X	X			
formázási hiba		X												X		
szintaktikai hiba															X	X
szemantikai hiba																
tipográfiai hiba	X		X				X	X	X	X	X			X		
hiba megnevezése	hibás margó	bal oldali behúzás az egész dokumentumon	aláhúzás	főlöleges enter térköz helyett	főlöleges üres enterek	üres szóközpök és tabulátorok igazazítás helyett	a megadott email cím vegyesen formázott	nagybetűs	aláhúzás, félkövér	nagybetűs, félkövér	félkövér dőlt	főlöleges szóköz	kézi számozás	Nagybetűs, aláhúzott, nagyobb betűméret	elírás	hibás mondatvégi írásjel

## 38. ábra

*A mozirajongok.doc minta alapján kitöltött hibatáblázata (28. ábra).  
A szürke mezők mutatják a nyomtatásban is felismerhető hibákat*

	17.	18.	19.	20.	21.	22.												
helyes																		
tördelési hiba			×		×													
formázási hiba		×		×														
szintaktikai hiba	×					×												
szemantikai hiba																		
típoográfiai hiba	×	×																
hiba megnevezése	kojtjel, n-dash helyett	Courier New betűtípus	fölösleges enter, távolság a lapszélétől helyett	fölösleges tabulátor pozíciók az élőféjben	az élőféjben a szöveg után van még egy fölösleges, üres enter	vessző hibák a felsorolásban (nincs az ábrán jelölve)												

### 39. ábra

*A mozirajongok.rtf minta alapján kitöltött hibatáblázata (28. ábra).  
A szürke mezők mutatják a nyomtatásban is felismerhető hibákat*

#### 2.4.3.2. Hibajavítás menete

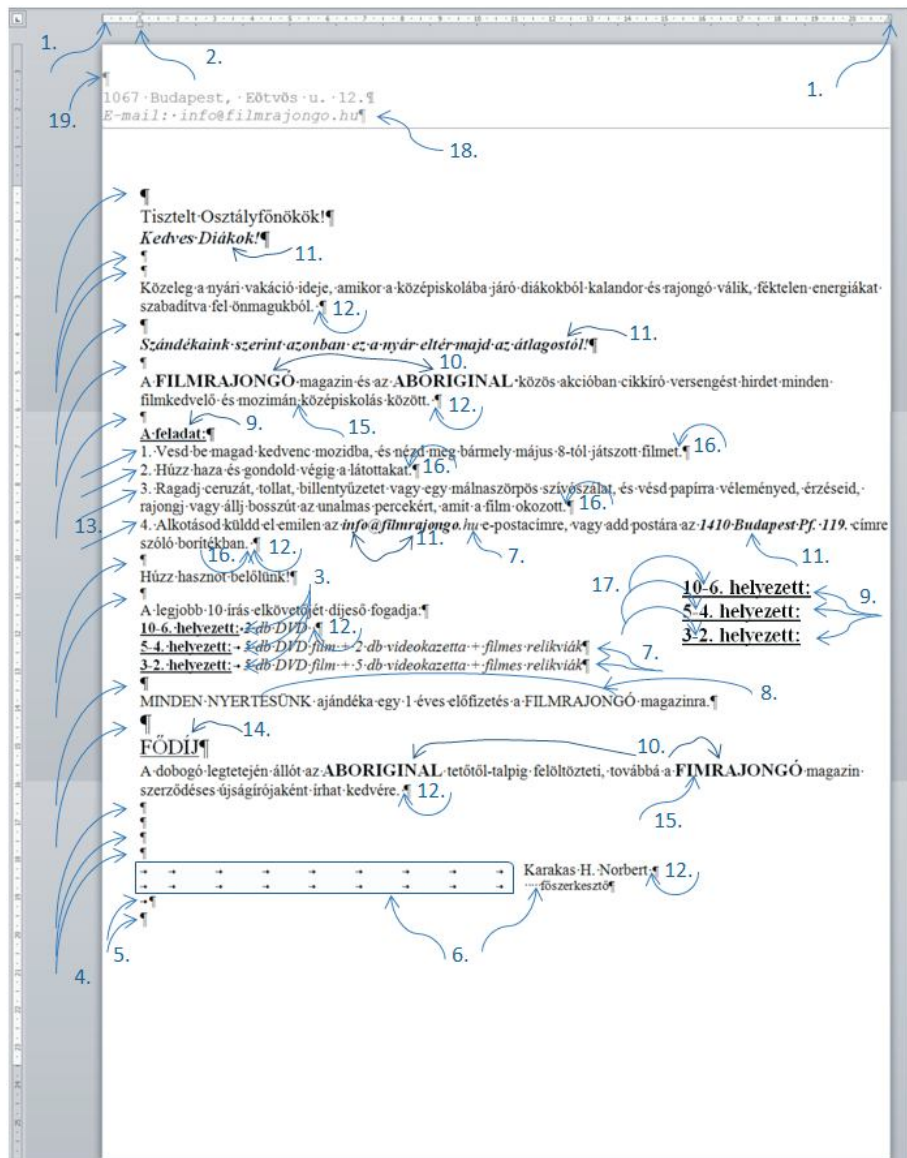
A hibajavítást ebben az esetben is érdemes a betű- és bekezdésformázások törlésével kezdeni, így sokkal áttekinthetőbb lesz a dokumentum, jobban látható a kurzor pozíciója. A formázás törlésével megszüntettük a #2, #7, #9, #11, részben a #8, #10, #18, #19 hibákat.

A szintaktikai hibák javításával megszűnnek a #15, #16, #17 hibák.

A tördelési hibáknál érdemes a bekezdések közötti enterekkel és a további üres bekezdések törlésével kezdeni #4, #5. A fölösleges tabulátorok és szóközök törlése különösen a #6 hibánál nagyon fontos. Érdemes az oszlopos kijelölést használni a tabulátorok törlésére, majd a szókjelölést a szóközök törlésére, mivel ez a leggyorsabb megoldás. A dokumentum nagyon sok esetben bekezdés végi fölösleges szóközöket használ, ezeket is törölni kell, #12. A bekezdések kézi számozásánál nem lesz szükség a számra, a pontra és szóközre, #20.

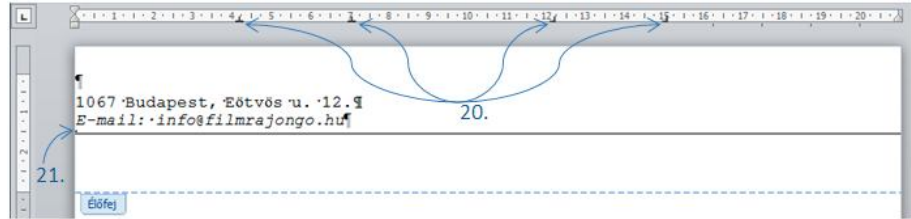
Fontosnak tartom megjegyezni, hogy a helyezéseknél található tabulátor karakterek jók, tehát azokat nem szabad törölni. Azzal a szövegrészsel az a gond, hogy nem hozták létre a megfelelő tabulátor pozíciót a második oszlop számára, de ez már formázási probléma #3. Az #1 hiba is formázással javítható.

Az élőfej területén törölni kell az összes formázást, ezzel megszüntetjük a #13 és #21 hibákat! Az élőfejben is törölni kell a fölösleges entert, #14! A formázás törlése és a #14 hiba javítása után látható, hogy az élőfejben még volt egy eddig rejtett enter, természetesen ezt is törölni kell, #22.



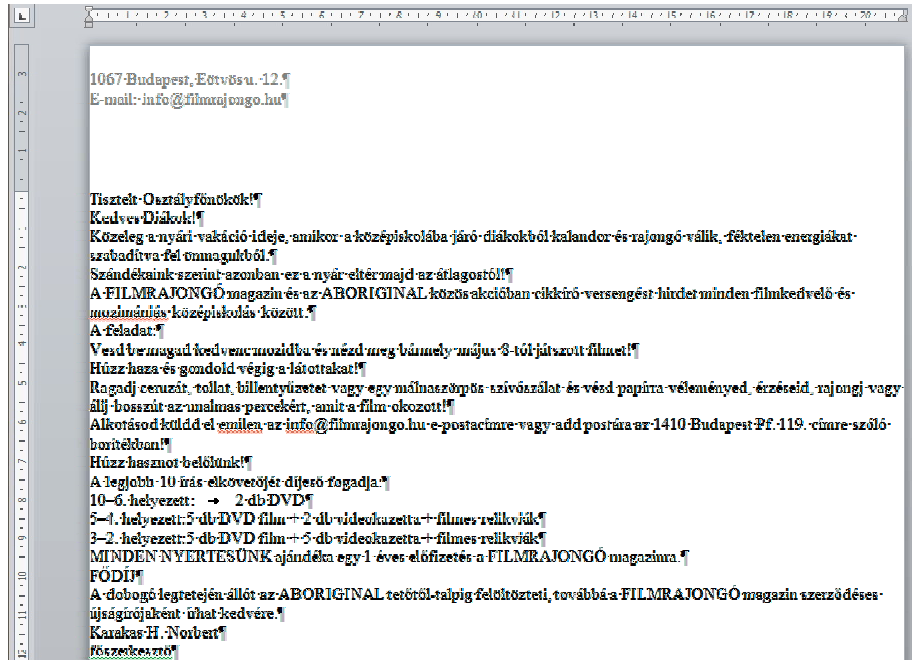
41. ábra

A mozirajongok.rtf a hibák megjelölésével



42. ábra

A mozirajongok.rtf fejlécben látható hibái



43. ábra

mozirajongok\_javitott.doc végső formája

### 2.4.3.3. Formázás

Fájlnév: *mozirajongok\_formazott.doc*

#### Teljes szövegre vonatkozó formázások

– papírméret és margó beállítása (hatókör: teljes dokumentum)

- automatikus elválasztás (hatókör: teljes dokumentum)
- igazítás: sorkizárt (hatókör: bekezdés)

Míg az előző két dokumentumban elegendő volt a papírméret és a margók ellenőrzése, ebben a dokumentumban ténylegesen el kell végezni a margók módosítását. A szerző eredeti szándékát figyelembe véve a bal, jobb és az alsó margót állítsuk vissza az alapértelmezett értékekre, míg a felső margót vegyük szélesebbre, 4 cm-re. Be kell még állítanunk az élőfej távolságát a lap szélétől 2 cm-re (*Lap elrendezése* → *Oldalbeállítás* → *Elrendezés* → *Távolság a lap szélétől* → *Élőfej*)! Ezekkel a beállításokkal javítjuk az #1, #14 és #23 hibákat.

Érdemes elvégezni a szokásos beállításokat: automatikus elválasztás (hatókör: teljes dokumentum, nem kell kijelölni) és a sorkizárt igazítás (hatókör: bekezdés, az egész dokumentum kijelölése).

### **Bekezdésformázások**

Térközök beállítása, automatikus számozás, aláírás formázása (#4, #6).

Bekezdés formázás még a helyezett második oszlopának kialakítása, amelyet egy balra igazított tabulátorral a legegyszerűbb elvégezni. A tabulátor hatóköre a bekezdés, tehát ki kell jelölni az ide tartozó bekezdéseket, majd el kell helyezni egy balra típusú tabulátor pozíciót, a szöveget figyelembe véve, 3,5 cm körül (#3)! A tabulátor létrehozásához a legegyszerűbb a vonalzót használni.

### **Karakterformázások**

A karakterformázásoknál el kell távolítani a fölösleges nagybetűket, a nevek esetében egy kiskapitális formázás még elfogadható! Sok időt megtakaríthatunk ha nem végzünk felesleges kijelöléseket. Ha csak egy-egy szóról kell eltávolítani a nagybetűt és ezt lecserélni kiskapitálisra, ekkor nincs szükség kijelölésre: elég ha a kurzort elhelyezzük a szó belsőjében, mivel a karakterformázások határáköre a szó.



### ***3. Táblázatkezelés mélyszerkezetű metakognitív megközelítése (21–24. óra)***

A táblázatkezelő programok a legnépszerűbb alkalmazói szoftverek közé sorolhatók (Borbély 2013: 120–127, Scaffidi–Shaw–Myers 2005: 207–214). Nemzetközi kutatások mutatják azonban, hogy a táblázatkezelőkkel készített dokumentumok közel 90%-a hibás (Panko–Aurigemma 2010: 235–244, Powell–Baker–Lawson 2008: 128–138, Powell–Baker–Lawson 2009a: 4–36, Powell–Baker–Lawson 2009b: 126–132, Teo–Tan 1999: 141–160, Tort–Blondel–Bruillard, É. 2008: 141–160, Tort: 2010). A nagyszámú hibás dokumentumok következménye, hogy az ezekből nyert téves információk óriási veszteségeket okoznak mind a készítőknél, mind a felhasználóknál, mind a megrendelőknél. Mindezen túl, a hozzá nem értés következtében óriási az az emberi-, erőforrás- és idővesztés, ami ezen dokumentumok létrehozásához és feldolgozásához szükséges (Abraham–Erwing: 2009, Jorgensen: 2013, Kadujevich 2009: 241–248, Kadujevich 2013: 19–33, Kwak: 2013, Panko–Aurigemma 2010: 235–244, Panko 2008: 15–21, Powell–Baker–Lawson 2008: 128–138, Powell–Baker–Lawson 2009a: 4–36, Powell–Baker–Lawson 2009b: 126–132, Report of JPMorgan Chase & Co: 2012, Teo–Tan: 1999: 141–160, Tort–Blondel–Bruillard 2008: 305–316, Tort 2010: 99–100, Van Deursen–Van Dijk: 2012).

Az okok keresése során egyértelművé vált, hogy a hibák megjelenésének elsődleges forrása az algoritmikus szemlélet hiánya. Ez a hiányosság megjelenik a táblázatkezelés valamennyi tevékenységi formájában; a dokumentumok tervezésénél, az adatbevitelnél, a képletek létrehozásánál (Anführungszeichen, Csernoch–Biró 2013a: 949–956, Csernoch–Biró: 2013b, , Csernoch–Biró: 2015a, Csernoch–Biró: 2015b, Sestoft 2010. Elvitathatatlan tény, hogy a táblázatkezelő programokon belül létrehozott képletek metakognitív döntések sorozatának eredményeként jönnek létre. Kérdés azonban, hogy ezek a metakognitív döntések felületi (TAEW, Trial-And-Error Wizard based, varázslóalapú próbálgatások sorozata) vagy mélyszerkezetű (CAAD, Computer Algorithmic And Debugging based, algoritmusépítés és tesztelés) megközelítések (Csernoch–Biró 2013a: 949–956, Csernoch–Biró: 2013b, Biró–Chernoch: 2014, Cox 2005: 104–141, Csernoch–Biró: 2013c, Csernoch–Biró: 2015b). A grafikus felületek

(GUI) használata során hajlamosak vagyunk megfélekedni a számítógépek és az azon futó programok algoritmikus jellegéről. A felhasználók többsége azt gondolja, hogy kattintgatások ad hoc sorozatával képesek vagyunk eljutni a kívánt eredményekhez. Ez azonban nem feltétlenül van így. Csak abban az esetben juthatunk el a kiindulási adatoktól a kívánt végeredményhez, ha tudatosan megépítjük az algoritmust, helyesen kódoljuk ezt az algoritmust, majd képesek vagyunk tesztelni a kapott eredményeket. Ez a lépéssorozat azonban csakis mélyszerkezetű metakognitív döntéssorozattal valósítható meg (Case–Gunstone–Lewis 2001: 331–355, Case–Gunstone 2002: 459–470, Case–Gunstone 2003: 801–819, Case 2000, Cox 2005: 104–141, Csernoch–Biró: 2013c, Csernoch–Biró: 2015b, Csíkos 2006, Koriat–Levy–Sadot 2000: 193–202). Grafikus felületen, így a táblázatkezelő programokon belül is a CAAD-típusú megközelítés egy lehetséges megoldása a problémának, amelyhez eszközként használhatóak a tömbképletek.

A mérési eredményeket figyelembe véve egyrészt állíthatjuk, hogy a táblázatkezelés algoritmikus megközelítése csökkenti a táblázatkezelői dokumentumokban található hibákat, másrészt előkészítője lehet a magas szintű programozási nyelvek tanulásának.

### **3.1. Sprego programozás (21. óra)**

A táblázatkezelő programok a funkcionális nyelvek csoportjába tartoznak (Sestoft: 2010), és mint ilyenek, alkalmasak programozás oktatására és használhatóak programozási nyelvként (Warren 2004: 327–333). Nem szabad azonban elfelejtkeznünk arról, hogy a táblázatkezelő programok lényegesen szűkebb lehetőségekkel rendelkeznek, mint a magas szintű programozási nyelvek.

A táblázatkezelő programok tehát programozási nyelvek, amelyek kezdőknek, valamint azon felhasználóknak, akik egy egyszerű környezetben szeretnének programozni – end-user programmers –, egy remek lehetőség. Alkalmas az algoritmikus készség fejlesztésére és megbízható programozási alapismereteket nyújt, amellyel tovább lehet lépni más programozási nyelvekre és tovább lehet fejleszteni a táblázatkezelőn belüli programozási ismereteket.

Az elterjedt TAEW típusú megközelítéssel szemben állítjuk, hogy táblázatkezelői környezetben fejleszthető az algoritmikus készség, a táblázatkezelő programok funkcionális nyelve szolgálhat bevezető nyelvként

már általános és középiskolában is. Ehhez dolgoztuk és próbáltuk ki a Sprego (Spreadsheet Lego) programozást (Biró–Csernoch: 2014, Csernoch 2014b, Csernoch–Balogh: 2010, Csernoch–Biró 2015a).

A Sprego programozás lényege, hogy a létező legkevesebb általános célú függvényt használva hozzunk létre összetett függvényeket. A Sprego programozás egy tucat függvényt használ kiindulásként, valamint megengedi ezen készlet bővítését további általános célú függvényekkel (1. táblázat). Ennek megfelelően három Sprego csoportot hoztunk létre: Sprego 1 és 2 csoportok tartalmazzák a feltétlenül szükséges egy tucat függvényt, míg Sprego 3 a további opcionális függvényeket. Az egyszerű függvényekből létrehozott összetett függvényekkel már bonyolult problémák megoldása is lehetséges.

A Sprego programozás további előnye egyrészt, hogy nagyban támaszkodik a matematikából ismert függvény fogalomra, ugyanakkor az összetett függvények intenzív használatával megerősíti ezt a fogalmat, különös tekintettel az értelmezési tartomány és az értékkészlet közötti kapcsolatra. Másrészt, a magas szintű programozási nyelvekkel szemben, óriási előny a nyelv egyszerűsége, tehát a hangsúly nem a kódolás részletein van, hanem az algoritmus építésén és tesztelésén. Sprego programozási környezetben kiemelt fontosságú a nyelv egyszerűsége, tehát a hangsúly még inkább a feladat elemzésén, az algoritmus megépítésén, a kapott eredmények tesztelhetőségén és tesztelésén van.

### 3.1.1. Sprego eszközei

A Sprego programozás egyrészt használja az általános célú Sprego függvényeket (1. táblázat), valamint a táblázatkezelő programok további, ezt a megközelítést támogató egyéb eszközeit. Ezek az eszközök mindegyike jelen van a népszerű táblázatkezelő programokban, de jelenleg csak marginális szerepet töltenek be. A Sprego bevezetésével célunk az, hogy ezeket az egyszerű eszközöket összegyűjtve létrehozzunk egy programozási környezetet, amellyel a táblázatkezelő programokon belül mély megközelítésű metakognitív problémamegoldás valósítható meg.

- Sprego függvények
- Összetett képletek
- Tömbképletek
- Tesztelés (debugging)
- Autentikus táblázatok használata

### 3.1.2. Sprego függvények

A Sprego függvények jellemzője, hogy általános célú, egyszerű függvények, amelyek verzió függetlenek. Ez magába foglalja a leggyakoribb táblázatkezelők – Microsoft Excel és OpenOffice, LibreOffice Calc – valamennyi verziója közötti teljes kompatibilitást. Mivel a Sprego csak általános célú egyszerű függvényeket használ, ezért nincs szükség sem a programok, sem a verziók folyamatos ellenőrzésére. Az egyszerű függvények használata mentesíti továbbá a felhasználókat a probléma specifikus függvények használatától, amelyek következtlen argumentumlistái, nehezen érthető varázslói és sűgői tovább növelik a dokumentumok hibaérzékenységét (Csernoch: 2014b). A táblázatkezelő programok használatát megnehezíti továbbá a függvények rendkívül magas, folyamatosan emelkedő száma. A felhasználók elvesznek ebben a függvénysokaságban, nem értik a sűgők és a varázslók szövegezését, tehát nem tudják érdemben használni a rendelkezésre álló függvényeket (Excel functions (alphabetical list), Excel functions (by category), Meet the spreadsheet: 2013, Microsoft Excel 2010 function translations).

Sprego 1	Sprego 2	Sprego 3
SZUM()	INDEX()	KICSI()
ÁTLAG()	HOL.VAN()	NAGY()
MIN()	HIBÁS()	SOR()
MAX()		OSZLOP()
BAL()		ÉS()
JOBBS()		VAGY()
HOSSZ()		NEM()
SZÖVEG.KERES()		HELYETTE()
HA()		ELTOLÁS()
		TRANSZPONÁLÁS()
		KEREKÍTÉS()
		VÉL()
		INT()

**1. táblázat**  
*Sprego függvények*

### 3.1.3. Összetett függvények

Összetett függvények létrehozása annyit jelent, hogy a függvényeket egymásba ágyazzuk, tehát egy függvény teljes egészében magába foglal egy másikat, ami leginkább a matrjoska babák összeépítéséhez (1. ábra) hasonlít. Ilyen összetett függvények létrehozása esetén a kiértékelés a legbelső függvény kiértékelésével kezdődik. A legelőször kiértékelte függvény visszaadott értéke lesz az őt magába foglaló függvény bemenő értéke, egyik argumentuma, és ez így folytatódik, amíg a legkülső függvény kiértékelése is megtörténik. A legkülső függvény visszaadott értéke nem más, mint az összetett függvény visszaadott értéke.



**1. ábra**

*Függvények egymásba ágyazása a matrjoska babák<sup>1</sup> egymás köré építéséhez hasonlóan történik*

### 3.1.4. Tömbképletek

A tömbképletek már a korai táblázatkezelő programokban is megjelentek, de valahogy nem került a figyelem középpontjába, annak ellenére, hogy rendkívül hatékonyan és biztonságosan használhatóak (Csernoch: 2012: 7270–7279, Pearson: 2007, Walkenbach–Wilcox: 2003, Wilcox–Walkenbach: 2003).

A tömbképletek egyik legnagyobb előnye, hogy helyettesítheti a másolással létrehozott képleteket, mivel így csak egyetlen képlet keletkezik, valamint az, hogy ezzel a módszerrel el tudjuk kerülni a másolásból származó hibákat is. A képletek másolásából származó hibák közül a leggyakoribbak:

<sup>1</sup> Matrjoska baba: <http://esemenyhorizont.uw.hu/2009/letfilo/beta01.jpg>, Letöltve 2012. május 17.

- a rögzítésekből származó hibák – a rögzítés elhagyása vagy túlzott használata,
- a képlet módosítása nem az első előfordulásnál,
- a módosítás nem történik meg az összes képletben,
- nem történik meg a módosított képlet átmásolása valamennyi előfordulásra.

Megkülönböztetünk egy- és többértékű tömbképleteket.

#### **3.1.4.1. Többértékű tömbképletek**

A tömbképletek létrehozása többféle módon is megtörténhet, de talán az egyik legbiztonságosabb megoldás a következő:

- Létrehozzuk a tömbképletet a tömb legelső cellájában. A képlet fordítását és kiértékelését a Ctrl+Shift+Enter billentyűkombinációval indítjuk el (Windows). A kiértékelés eredményeként kiíratásra kerül a tömb első cellájában az első output.
- Kijelöljük a teljes tömböt, ahova az eredményeket szeretnénk kiírítani.
- Újra szerkeszthetővé tesszük a képletet.
- Ismét fordítatjuk és kiértékeltetjük a képletet, tehát ismételten a Ctrl+Shift+Enter billentyűkombinációval zárjuk.

A Ctrl+Shift+Enter billentyűkombináció hatására megjelenik a képlet körül egy { } zárójelpár. Ezek a zárójelek nem gépelhetőek, csak és kizárólag a Ctrl+Shift+Enter billentyűkombinációval helyezhetők el.

#### **3.1.4.2. Egyértékű tömbképletek**

Az egyértékű tömbképletek annyiban térnek el a többértékűektől, hogy nem jelenítjük meg a tömb elemeit a táblázatban. Számolunk ezekkel az értékekkel, de a tömb elemeit nem íratjuk ki a táblázat celláiba. Az így létrehozott képlet egyik óriási előnye, hogy csökkentjük a táblázat méretét, ezzel lényegesen áttekinthetőbb lesz és kevesebb helyet foglal a tárolás során.

Az egyértékű tömbképletekben a műveletek elvégzése a teljes tömbön megtörténik, tehát a fordítás és a kiértékelés, hasonlóan a többértékű tömbképletekhez, a Ctrl+Shift+Enter billentyűkombináció indítható el.

Az egyértékű tömbképleteknél mindenképpen meg kell említenünk az \*IF?() függvények csoportját. Az \*IF?() függvények csoportja sem az Excel sem a Calc terminológiájában nem szerepel, ezt a kifejezést mi hoztuk

létre. Ezzel az összefoglaló névvel látjuk el azokat a függvényeket, amelyek feltételt/feltételeket foglalnak magukba. Az \*IF?() függvények nem mások, mint a feltételt tartalmazó egyértékű tömbképletek leegyszerűsített, sokkal inkább lebutított, véges számban rendelkezésre álló verziói. Használatuk több szempontból is nehézkes és csak speciális esetekben alkalmazhatóak.

angol	magyar	kategória	argumentumok
COUNTIF()	DARABTELI()	S	feltétel_tart;feltétel
COUNTIFS()	DARABHATÓBB()	S	feltétel_tart1;feltétel1;...
SUMIF()	SZUMHA()	M	feltétel_tart;feltétel;összeg_tart
SUMIFS()	SZUMHATÓBB()	M	összeg_tart;feltétel_tart1;feltétel1...
AVERAGEIF()	ÁTLAGHA()	S	feltétel_tart;feltétel;átlag_tart
AVERAGEIFS()	ÁTLAGHATÓBB()	S	átlag_tart;feltétel_tart1;feltétel1;...
COUNTBLANK()	DARABÜRES()	S	tartomány

## 2. ábra

*Az \*IF?() függvények angol és magyar nyelvű megnevezése, kategóriái (M: Matematikai és trigonometriai, S: Statisztikai) és argumentumai*

### Nehézségek az \*IF?() függvényekkel

A következőkben felsorolásra kerülnek azok a problémák, amelyeket az \*IF?() függvények magukban hordoznak.

- A különböző problémák megoldásához különböző függvények és különböző kategóriákba sorolt függvények szükségesek (Math and trigonometry functions, Statistical functions). Ezeket vagy meg kell jegyezni, vagy meg kell keresni a függvények bőséges listájában.
- Hatalmas hátránya az \*if?() függvényeknek, azok korlátozott száma.
- Az elérhető függvények száma verzióként változik.
- A függvények argumentumlistája nehezen megjegyezhető és követhető.
- A feltétel(ek) szintaktikája eltérő attól függően, hogy egyenlőséget vagy egyenlőtlenséget vizsgálunk, valamint attól, hogy konstans vagy változó szerepel a feltételben. Az \*if?() függvények feltételében egyenlőséget vizsgálva nincs szintaktikai eltérés konstans vagy változó esetén. Ezzel szemben, egyenlőtlenségek vizsgálata során más szintaktikai szabályok vonatkoznak a konstansokra és a válto-

zókra. Mindenképpen érdemes azonban megjegyezni, hogy más programozási nyelvekhez hasonlóan, nem célszerű egyik típusú képletnél sem konstansokat használni, mivel a bemenő érték módosulása esetén a képletet is módosítani kell, valamint az összes olyan képletet, amely az eredeti másolásával jött létre.

- A több-feltételes függvények esetén csak és kizárólag és kapcsolat valósítható meg a feltételek között.
- Számított értékek nem adhatók meg az \*if?() függvények argumentumaként. Ez a megszorítás nagyban korlátozza a függvények használhatóságát. Némi könnyebbséget jelent, hogy a feltételekben megadott helyettesítő karaktereket (\*, ?) értelmezni tudja a fordító program.
- Az \*if?() függvények helyettesíthetők az adatbázis függvényekkel is. Ezek használatára azonban a kezdő tanulók, felhasználók nincsenek felkészülve (Database functions).

### 3.1.5. Tesztelés

A programozási nyelvekből ismert tesztelés, ellenőrzés (debugging) a táblázatkezelő programok használata során nem terjedt el, annak ellenére, hogy ebben a környezetben is legalább olyan fontos, mint tradicionális programozási környezetekben. Ezentúl, rendelkezésre állnak a kiértékelés eszközei (Képletvizsgálat), amelyek közül érdemes kiemelni a *Képletkiértékelő* ablakot, valamint egy felületi eszközt, az *Elődök/Utódok mutatása* parancsot.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Ország	Földrész	Főváros	Terület	Lakosság (ezer fő)									
2	Afganistan	Asia	Kabul	647500	27756						nagyobb, mint 500 km <sup>2</sup>			500
3	Albania	Europe	Tirana	28748	3545						nagyobb, mint 500 km <sup>2</sup>			
4	Algeria	Africa	Algiers	2381740	32278						nagyobb, mint 500 km <sup>2</sup>			
5	American Samoa	Oceania	Pago Pago	199	69									
6	Andorra	Europe	Andorra la Vella	468	68									
7	Angola	Africa	Luanda	1246700	10593						nagyobb, mint 500 km <sup>2</sup>			
8	Anguilla	Amerika	The Valley	102	12									
9	Antigua and Barbuda	Amerika	Saint John's	443	67									

3. ábra

A képletben felhasznált cellák (Elődök) megmutatása a táblázatkezelői felületen

Az MS Excel képletkiértékelő ablaka – *Képletek* → *Képletkiértékelő* – egy jól működő kiértékelő felület, sajnos azonban még mindig pontosan



olyan fapados, mint 20 évvel ezelőtt. Egyetlen változtatás – a kiértékelő ablak méretezhetősége – azonban nagyban megkönnyítené a tesztelés menetét, mivel a jelenleg elérhető 5 soros és 50–60 karakter széles ablakban nem egyszerű a képleteket átlátni.

A tesztelés menetének felgyorsítása, a könnyebb navigáció érdekében érdemes egy csonkított táblázatot használni, és ezen ellenőrizni a képletek helyességét. A módszer lényege, hogy a tesztelést nem élesben végezzük a teljes táblázaton, hanem annak csak egy csonkított verzióján, amelyben szimuláljuk a nagy táblázat valamennyi problémás esetét. Kevesebb adattal lényegesen gyorsabban el tudjuk végezni a tesztelést, lényegesen áttekinthetőbbek az eredmények, valamint a Képletkiértékelő rögzített mérete miatt érdemes olyan mennyiségű tesztadatot választani, ami kényelmesen elfér a tesztablakban.



#### 4. ábra

*A Képletkiértékelő ablaka: dőlt jelzi az előző lépés outputját, az aláhúzás a következő lépést (2. ábra, 4. lépés). A bal oldali minta a teljes táblázat, míg a jobb oldali a csonkított tábla tesztelését mutatja*

	$SZUM(HA(D2:D11>N2;1))$	$SZUM(HA(D2:D11>N2;E2:E11))$
1. lépés	$SZUM(HA(\{647500;28748;2381740;199;468;1246700;102;443;2766890;29800\}>500;1))$	$SZUM(HA(\{647500;28748;2381740;199;468;1246700;102;443;2766890;29800\}>500;E2:E11))$
2. lépés	$SZUM(HA(\{IGAZ;IGAZ;IGAZ;HAMS;HAMS;IGAZ;HAMS;HAMS;IGAZ;IGAZ\};1))$	$SZUM(HA(\{IGAZ;IGAZ;IGAZ;HAMS;HAMS;IGAZ;HAMS;HAMS;IGAZ;IGAZ\};E2:E11))$
3. lépés	$SZUM(\{1;1;1;HAMS;HAMS;1;HAMS;HAMS;1;1\})$	$SZUM(\{27756;3545;32278;HAMS;HAMS;10593;HAMS;HAMS;37813;3330\})$
4. lépés	6	115315

### 2. táblázat

*Példa a Képletkiértékelő működésére két hasonló tömbképlet esetén*

A beépített függvények használatának egyik nagy hátránya az összetett függvényekkel szemben, hogy a beépített függvények esetében nem követhetőek a végrehajtás lépései; egylépéses a kiértékelés, így nem tudjuk, hogy milyen lépéssorozat eredményeként született meg a végeredmény. Beépített függvények kiértékelésére mutat példát a 3. táblázat. A táblázat adataiból egyértelmű, hogy a beépített függvények esetén nem tudjuk követni a kiértékelés lépéseit.

0. lépés	DARABTELI(C2:C251;évszám)	=DARABTELI(C2:C251;évszám)
1. lépés	DARABTELI(\$C\$2:\$C\$251;2001)	
2. lépés	6	
0. lépés	SZUMHA(C2:C251;évszám;D2:D251)	=SZUMHA(C2:C251;évszám;D2:D251)
1. lépés	SZUMHA(\$C\$2:\$C\$251;2001;D2:D251)	
2. lépés	1 800 178	
0. lépés	SZUMHATÖBB(D2:D251;C2:C251;évszám)	=SZUMHATÖBB(D2:D251;C2:C251;évszám)
1. lépés	SZUMHATÖBB(\$D\$2:\$D\$251;C\$2:C\$251;2001)	
2. lépés	1 800 178	
0. lépés	ÁTLAGHA(C2:C251;évszám;D2:D251)	=ÁTLAGHA(C2:C251;évszám;D2:D251)
1. lépés	ÁTLAGHA(\$C\$2:\$C\$251;2001;D2:D251)	
2. lépés	30 0029,6667	
0. lépés	ÁTLAGHATÖBB(D2:D251;C2:C251;évszám)	=ÁTLAGHATÖBB(D2:D251;C2:C251;évszám)
1. lépés	ÁTLAGHATÖBB(\$D\$2:\$D\$251;C\$2:C\$251;2001)	
2. lépés	30 0029,6667	

### 3. táblázat

Az \*IF?() és \*IFS() függvények kiértékelésének menete a Képletkiértékelőben

#### 3.1.6. Autentikus táblázatok

A táblázatkezelőben történő programozás egy híd a végfelhasználói tevékenységek és a valódi programozás között. A táblázatkezelő programok azonban rendelkeznek azzal a tulajdonsággal, amivel több iskolai célú programozási nyelv nem; egy olyan eszköz, amelyre az iskola elhagyása után is szükség lesz, valamint már a tanulása során lehet valódi, autenti-

kus problémák megoldására használni. Tehát van „értelme” annak, hogy megtanulják a használatát.

Az internet egy kimeríthetetlen táblázat forrás, bármilyen életkorhoz, érdeklődési körhöz, tudásszinthez található táblázatokat. A megfelelően előkészített, valódi tartalmakat hordozó táblázatok érdekelhetik és érdeklik is a tanulókat. Kíváncsivá teszi őket a jól kiválasztott táblázat, ennek következtében valódi adatfeldolgozást, lekérdezést, információszerzést tudunk végezni a táblázatkezelés tanulása közben.

### 3.2. Sprego feladatok (22–24. óra)

A problémák megoldása CAAD típusú, mély metakognitív megközelítéssel történik, a tradicionális programozási környezetekben használt módszerek adaptálásával. Megfogalmazzuk a problémát, elemizzük a rendelkezésre álló inputot, értelmezzük a várható outputot, ezt követően megépítjük az algoritmust a bemeneti és a kimeneti adatok között, kódoljuk az algoritmust, futtatjuk, majd teszteljük a kapott eredményt.

#### 3.2.1. US states (22–23. óra)

„List of states and territories of the United States”<sup>2</sup> című weblap államokat felsoroló táblázatának részletét mutatja az 5. ábra, majd ennek táblázatkezelő dokumentummá konvertált verzióját a 6. ábra.

State	Abbr.	Capital	Largest city <sup>(ST)</sup>	Statehood	Population (2013 est) <sup>(1)</sup>	Total area in mi <sup>2</sup> (km <sup>2</sup> ) <sup>(2)(3)</sup>	Land area in mi <sup>2</sup> (km <sup>2</sup> ) <sup>(2)(3)</sup>	Water area in mi <sup>2</sup> (km <sup>2</sup> ) <sup>(2)(3)</sup>	House seat(s)
 Alabama	AL	Montgomery	Birmingham	December 14, 1819	4,833,722	52,420 (135,767)	50,645 (131,170)	1,775 (4,597)	7
 Alaska	AK	Juneau	Anchorage	January 3, 1959	735,132	665,384 (1,723,337)	570,641 (1,477,950)	94,743 (245,383)	1
 Arizona	AZ	Phoenix	Phoenix	February 14, 1912	6,626,624	113,990 (295,233)	113,594 (294,207)	396 (1,026)	9
 Arkansas	AR	Little Rock	Little Rock	June 15, 1836	2,959,373	53,179 (137,733)	52,035 (134,770)	1,143 (2,960)	4
 California	CA	Sacramento	Los Angeles	September 9, 1850	38,332,521	163,696 (423,968)	155,779 (403,466)	7,916 (20,502)	53
 Colorado	CO	Denver	Denver	August 1, 1876	5,268,367	104,094 (269,602)	103,642 (269,432)	452 (1,171)	7
 Connecticut	CT	Hartford	Bridgeport	January 9, 1788	3,596,080	5,543 (14,356)	4,842 (12,541)	701 (1,816)	5
 Delaware	DE	Dover	Wilmington	December 7, 1787	925,749	2,489 (6,446)	1,949 (5,048)	540 (1,399)	1
 Utah	UT	Salt Lake City	Salt Lake City	January 4, 1896	2,900,872	84,897 (219,882)	82,170 (212,819)	2,727 (7,063)	4
 Vermont	VT	Montpelier	Burlington	March 4, 1791	626,630	9,616 (24,905)	9,217 (23,872)	400 (1,036)	1
 Virginia <sup>(2)</sup>	VA	Richmond	Virginia Beach	June 25, 1788	8,260,405	42,775 (110,787)	39,490 (102,279)	3,285 (8,508)	11
 Washington	WA	Olympia	Seattle	November 11, 1889	6,971,406	71,298 (184,661)	66,456 (172,120)	4,842 (12,541)	10
 West Virginia	WV	Charleston	Charleston	June 20, 1863	1,854,304	24,230 (62,755)	24,039 (62,258)	192 (497)	3
 Wisconsin	WI	Madison	Milwaukee	May 29, 1848	5,742,713	65,496 (169,634)	54,158 (140,269)	11,339 (29,368)	8
 Wyoming	WY	Cheyenne	Cheyenne	July 10, 1890	582,658	97,813 (253,335)	97,093 (251,470)	720 (1,865)	1

#### 5. ábra

*Az USA államok weblap táblázatának első és utolsó néhány sora*

<sup>2</sup> List of states and territories of the United States. [http://en.wikipedia.org/wiki/List\\_of\\_states\\_and\\_territories\\_of\\_the\\_United\\_States](http://en.wikipedia.org/wiki/List_of_states_and_territories_of_the_United_States). Letöltve: 2014. november 5.

	A	B	C	D	E	F	G
1	State	Abbr.	Capital	Largest city	Statehood	Population	Total area in mi2 (km2)
2	Alabama	AL	Montgomery	Birmingham	December 14, 1819	4,833,722	70045242000000000052,420 (135,767)
3	Alaska	AK	Juneau	Anchorage	January 3, 1959	735,132	7005665384000000000665,384 (1,723,337)
16	Iowa	IA	Des Moines	Des Moines	December 28, 1846	3,090,416	700456273000000000056,273 (145,746)
17	Kansas	KS	Topeka	Wichita	January 29, 1861	2,893,957	700482278000000000082,278 (213,099)
18	Kentucky[C]	KY	Frankfort	Louisville	June 1, 1792	4,395,295	700440408000000000040,408 (104,656)
19	Louisiana	LA	Baton Rouge	New Orleans	April 30, 1812	4,625,470	700452378000000000052,378 (135,658)
20	Maine	ME	Augusta	Portland	March 15, 1820	1,328,302	700435380000000000035,380 (91,634)
46	Vermont	VT	Montpelier	Burlington	March 4, 1791	626,63	70039616000000000009,616 (24,905)
47	Virginia[G]	VA	Richmond	Virginia Beach	June 25, 1788	8,260,405	700442775000000000042,775 (110,787)
50	Wisconsin	WI	Madison	Milwaukee	May 29, 1848	5,742,713	700465496000000000065,496 (169,634)
51	Wyoming	WY	Cheyenne	Cheyenne	July 10, 1890	582,658	700497813000000000097,813 (253,335)

### 6. ábra

Az USA államok weblap konvertálásával létrehozott táblázat (válogatott sorok)

A táblázat oszlopai a következő adatokat tartalmazzák:

- A – államok neve, további átalakítás szükséges (3.2.1.1. fejezet)
- B – államok nevének rövidítése, nincs teendő
- C – állam fővárosa, nincs teendő
- D – állam legnagyobb városa, nincs teendő
- E – csatlakozás dátuma, további átalakítás szükséges
- F – lakosság, további átalakítás szükséges
- G – terület mi<sup>2</sup> és km<sup>2</sup>-ben megadva, további átalakítás szükséges

#### 3.2.1.1. Államok neve

Az államok nevei főlegesen karaktereket tartalmaznak – A oszlop (5. ábra és 6. ábra). Feladatunk ezen karakterek eltávolítása, tehát az államok nevének letisztázása.

#### Államnevek jellemzői

- Minden állam neve előtt – a névtől balra – található egy szóköz.
- Néhány állam nevét követően található egy sztring, amely az alábbi tulajdonságokkal rendelkezik (5. ábra Virginia, 6. ábra Kentucky és Virginia államok mutatja):
  - három karakter hosszúságú
  - az eredeti sztring jobb oldalán helyezkedik el.
  - szögletes zárójelbe van foglalva

**Algoritmus: szóközök eltávolítása**

- A szóköztől jobbra az összes karaktert kivágjuk és ez lesz a visszaadott érték.
- A kivágott és visszaadott sztring hossza eggyel kevesebb karakter, mint az eredeti sztring hossza.

**Kódolás: szóközök eltávolítása**

- Meghatározzuk az eredeti sztring hosszát (M1)
- Meghatározzuk az kivágandó sztring hosszát (M2)
- Az eredeti sztring jobb oldaláról kivágjuk a csonka, szóköz nélküli sztringet. Ez lesz a képlet visszaadott értéke (M3).

Az eredeti sztring hosszának meghatározásához a HOSSZ() függvényt használjuk.

M1 {=HOSSZ(A2:A51)}

M2 {=HOSSZ(A2:A51)-1}

Az M2 képlet visszaadott értéke, outputja az új sztring hossza. Ez egy egész szám. Ez a szám lesz a JOBB() függvény egyik inputja, a második argumentuma. A függvények összeépítésével a JOBB() függvény teljes egészében magába foglalja a HOSSZ() függvényt (4. táblázat).

M3 {=JOBB(A2:A51;HOSSZ(A2:A51)-1)}

	M1	M2	M3	M7	M8	M9	M10	M11
Alabama	8	7	Alabama	Alab	#ÉRTÉK!	IGAZ	Alabama	Alabama
Alaska	7	6	Alaska	Ala	#ÉRTÉK!	IGAZ	Alaska	Alaska
Iowa	5	4	Iowa	I	#ÉRTÉK!	IGAZ	Iowa	Iowa
Kansas	7	6	Kansas	Kan	#ÉRTÉK!	IGAZ	Kansas	Kansas
Kentucky[C]	12	11	Kentucky[C]	Kentucky	10	HAMIS	HAMIS	Kentucky
Louisiana	10	9	Louisiana	Louisi	#ÉRTÉK!	IGAZ	Louisiana	Louisiana
Maine	6	5	Maine	Ma	#ÉRTÉK!	IGAZ	Maine	Maine
Vermont	8	7	Vermont	Verm	#ÉRTÉK!	IGAZ	Vermont	Vermont
Virginia[G]	12	11	Virginia[G]	Virginia	10	HAMIS	HAMIS	Virginia
Wisconsin	10	9	Wisconsin	Wiscon	#ÉRTÉK!	IGAZ	Wisconsin	Wyoming
Wyoming	8	7	Wyoming	Wyom	#ÉRTÉK!	IGAZ	Wyoming	Alabama

**4. táblázat**

*Az államok neveinek tisztázásakor keletkezett visszaadott értékek*

**Algoritmus: államok neve utáni karakterek eltávolítás**

- Az államnevek hosszának meghatározása.
- A felesleges karakterek nélküli sztring kivágása a szóközökkel csonkított sztringből.

**Kódolás: államok neve utáni karakterek eltávolítás**

- Az államnevek hosszának meghatározásához használhatjuk a szóközmentes vagy az eredeti sztringet is. A szóközmentes sztringből kiindulva az M4, majd az M5 képletek, az eredeti sztringet használva az M6 adja az eredményt.
- A szóközmentes sztring bal széléről kivágjuk az állam nevét a bal() függvénnyel (M7)

M4 {=HOSSZ(JOBB(A2:A51;HOSSZ(A2:A51)-1))}

M5 {=HOSSZ(JOBB(A2:A51;HOSSZ(A2:A51)-1))-3}

M6 {=HOSSZ(A2:A51)-4}

A BAL() függvény első argumentuma a szóközmentes sztring, míg a második az állam nevének a hossza, ami az M5 vagy az M6 képletből visszaadott értéke.

M7 {=BAL(JOBB(A2:A51;HOSSZ(A2:A51)-1);HOSSZ(A2:A51)-4)}

**Algoritmus: szétválogatás a fölösleges karakterek alapján**

A gond azonban ott van, hogy a fölösleges három karakter csak alkalmanként jelenik meg az államok neve után. Tehát gondoskodnunk kell arról, hogy megkülönböztessük a kétféle sztringet az alapján, hogy van-e a végén fölösleges három karakter vagy nincs.

- Két eset lehetséges attól függően, hogy van fölösleges karakter a sztring végén vagy nincs.
- Csak abban az esetben kell a karakterek levágását elvégezni, ha létezik a három fölösleges karakter, egyébként a szóközmentes sztringet kell visszaadni.

**Kódolás: szétválogatás a fölösleges karakterek alapján**

- Megnézzük, hogy van-e fölösleges három karakter a sztring végén. Ezt megtehetjük úgy, hogy rákeresünk valamelyik szögletes zárójelre a szöveg.keres() függvénnyel (M8).

- Megnézzük, hogy a szöveg.keres() függvénynek mi a visszaadott értéke. Ha megtalálta a zárójelet, akkor a függvény egy egész számmal tér vissza, a zárójel pozíciójával. Ha nem talált a függvény szögletes zárójelet, akkor hibaüzenettel tér vissza.
- Leellenőrizzük a szöveg.keres() függvény visszaadott értékét a hibás() függvénnyel. A hibás() függvény két értékkel térhet vissza: igaz, hamis, attól függően, hogy hibával jött vissza a szöveg.keres() függvény vagy sem (M9).
- A hibás() függvény outputjától függően kiíratjuk vagy a teljes szóközmentes sztringet: (M10) vagy a szóközmentes és csonkított sztringet (M11).

Ennél a feladatrésznél is érdemes a szögletes zárójelet az eredeti rövid sztringben keresni, tehát átmenetileg kimásolhatjuk a csonkított sztring képletét egy cellába.

M8 {=SZÖVEG.KERES("[";A2:A51)}

M9 {=HIBÁS(SZÖVEG.KERES("[";A2:A51))}

M10 {=HA(HIBÁS(SZÖVEG.KERES("[";A2:A51));JOBB(A2:A51;HOSSZ(A2:A51)-1))}

M11 {=HA(HIBÁS(SZÖVEG.KERES("[";A2:A51));

JOBB(A2:A51;HOSSZ(A2:A51)-1);

BAL(JOBB(A2:A51;HOSSZ(A2:A51)-1);HOSSZ(A2:A51)-4))}

### 3.2.1.2. Population adatainak egységesítése

A lakosság egész számai az angol írásmód szerinti ezreselválasztó vessző karaktereket tartalmazzák (5. ábra). Az eredetileg egész számokat, a vessző számától függően a magyar Excel kétféleképpen értelmezi:

- egy vessző esetén valós számként – az eredeti érték ezred részének (6. ábra: Alaska, Vermont, Wyoming),
- egynél több vessző esetén sztringként (6. ábra: Alabama, Iowa, Kansas, Kentucky, Louisiana, Maine, Virginia, Wisconsin)

Az angol egész szám magyar valós számmá konvertálása során azonban adatvesztés is történhet, amennyiben az utolsó helyiértékeken 0 található. A feldolgozás során, a végrehajtás lépéseinek sorrendjénél, erre mindenképpen érdemes odafigyelni.



### Population jellemzői

- egy vesszőt tartalmazó valós számok, amelyek az eredeti érték ezred részei
- több vesszőt tartalmazó sztringek

A kétféle értéket szét kell választanunk és külön-külön elvégezni a konverziót, hogy visszakapjuk az eredeti értékeket. Érdeemes először külön oszlopokban megcsinálni a kétféle konvertálást, majd ezt követően összemásolni egy képletbe, amely mindkét esetet tudja kezelni.

	M12	M13	M14	M15	M16	M17
4,833,722	#ÉRTÉK!	4833722	4833722	IGAZ	0	4833722
735,132	735132	735132	735132	HAMIS	735132	735132
3,090,416	#ÉRTÉK!	3090416	3090416	IGAZ	0	3090416
2,893,957	#ÉRTÉK!	2893957	2893957	IGAZ	0	2893957
4,395,295	#ÉRTÉK!	4395295	4395295	IGAZ	0	4395295
4,625,470	#ÉRTÉK!	4625470	4625470	IGAZ	0	4625470
1,328,302	#ÉRTÉK!	1328302	1328302	IGAZ	0	1328302
626,63	626630	62663	62663	HAMIS	626630	626630
8,260,405	#ÉRTÉK!	8260405	8260405	IGAZ	0	8260405
5,742,713	#ÉRTÉK!	5742713	5742713	IGAZ	0	5742713
582,658	582658	582658	582658	HAMIS	582658	582658

#### 5. táblázat

*A lakosság (Population) értékeinek helyreállításához használt képletek visszaadott értéke*

#### Algoritmus: valós számok visszaalakítása egész számmá

- a szám szorzása 1000-rel

#### Kódolás: valós számok visszaalakítása egész számmá

M12 {=F2:F51\*1000}

#### Algoritmus: sztringek egész számmá alakítása

- vesszők eltávolítása
- sztring számmá alakítása

#### Kódolás: sztringek egész számmá alakítása

- A vesszők eltávolítását a sztringből a helyette függvénnyel érdemes elvégezni, ahol a vessző karaktert az üres sztringre cseréljük.

- A sztring a vessző eltávolítása után már csak számjegyeket tartalmaz, tehát egy szorzással számmá alakítható. A legegyszerűbb, ha a sztringet 1-gyel szorozzuk meg, hogy a szám értéke nem változzon.

M13 {=HELYETTE(F2:F51;"","")}

M14 {=HELYETTE(F2:F51;"","")\*1}

#### **Algoritmus: A kétféle adattípus szétválogatása**

- Rákérdezzük arra, hogy a cella számot tartalmaz-e.
- A választól függően vagy a szám konverziót vagy a sztring konverziót végezzük el.

#### **Kódolás: A kétféle adattípus szétválogatása**

- Megpróbáljuk szorzással számmá alakítani az eredeti értékeket. A visszakapott érték vagy egy szám vagy egy hibaüzenet. Ezzel már készen vagyunk (M12)
- A hibás() függvénnyel lekezeljük a visszaadott értéket, melynek visszaadott értéke vagy igaz vagy hamis. Kifelé bővíthetjük az M12 képletet (M15).
- Tovább bővíthetjük kifelé a képletet a ha() függvénnyel, amelynek első argumentuma lesz a teljes hibás() függvény, harmadik argumentuma, a ha() függvény hamis ága, pedig az átalakított szám. Ebben az esetben a ha() függvény igaz ágán 0 értéket kapunk visszaadott értéként (M16).
- Feltöltjük a ha() függvény igaz ágát a sztringből számmá konvertálás képlettel (M14). A teljes képletet az M17 mutatja.

M15 {=HIBÁS(F2:F51\*1000)}

M16 {=HA(HIBÁS(F2:F51\*1000));;F2:F51\*1000)}

M17 {=HA(HIBÁS(F2:F51\*1000);HELYETTE(F2:F51;"","")\*1;F2:F51\*1000)}

### **3.2.1.3. Dátum átalakítása**

#### **Dátumok jellemzői**

A táblázatban használt aktuális dátum formátumot a magyar Excel felismeri, van azonban két további probléma, ami miatt a sztring dátummá konvertálása nem történik meg automatikusan:

- a hónapok angol neve,
- azokban a hónapokban, ahol a hónap angol és magyar neve megegyezik megtörténik az automatikus dátum felismerés (november, december)
- az évek 1900 előttiék.

Az 1900 előtti évek kezelésére le kell tölteni az XDATE Add-Int, ám ezzel itt nem kívánunk részletesen foglalkozni.<sup>3</sup>

Az angol hónapnevek lecseréléshez szükségünk van egy segéd táblára, amely egyik vektora tartalmazza a hónapok nevét angolul, míg a másik magyarul.

	1900-as évek	M18	M19	M20	M21
14.dec.19	14.dec.19	#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!
January 3, 1959	January 3, 1959	8	7	January	1
December 28, 1846	28.dec.46	#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!
January 29, 1861	January 29, 1961	8	7	January	1
June 1, 1792	June 1, 1992	5	4	June	6
April 30, 1812	April 30, 1912	6	5	April	4
March 15, 1820	March 15, 1920	6	5	March	3
March 4, 1791	March 4, 1991	6	5	March	3
June 25, 1788	June 25, 1988	5	4	June	6
May 29, 1848	May 29, 1948	4	3	May	5
July 10, 1890	July 10, 1990	5	4	July	7

**6. táblázat**

*A dátum (Statehood) konvertálásához használt képletek visszaadott értéke*

<sup>3</sup> A képletek kipróbálhatók, ha az éveket átírjuk 1900-as évekre.

**Algoritmus: Angol nyelvű hónap nevek lecserélése magyarra**

- Kivágjuk a dátum sztringből a hónapot. A hónap nevének hossza eggyel kevesebb, mint az azt követő szóköz pozíciója.
- Az angol névhez megkeressük a magyar párját a segédtáblázatban.
- Az angol hónap nevet lecseréljük a magyar hónap nevekre.
- A sztringet számmá alakítjuk.

**Kódolás: Angol nyelvű hónap nevek lecserélése magyarra**

- A hónap nevét követő szóköz pozíciójának meghatározása a szöveg.keres() függvénnyel: M18.
- A hónapnév hosszának meghatározása (M19).
- A hónap kivágása a sztringből (M20).
- Az angol hónapnevek pozíciójának a meghatározása a vektorban a hol.van() függvénnyel: M21. A segédtáblába mind a 12 hónap nevet felvettük, így bármilyen két nyelv között működik a konverzió.
- A magyar hónapnevek kiírása az index() függvénnyel (M22).
- Az angol hónapnevek lecserélése magyarra az eredeti sztringben a helyette() függvénnyel: M23. A helyette() függvény argumentumai a következők:
  - eredeti sztring
  - M20 képlettel kivágott angol hónapnév
  - M22 képlettel kiíratott magyar hónapnevek.
- Dátum sztringek számmá alakítása (M24).

M18 {=SZÖVEG.KERES(" ";M2:M51)}

M19 {=SZÖVEG.KERES(" ";M2:M51)-1}

M20 {=BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1)}

M21 {=HOL.VAN(BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);A53:A64;0)}

M22	M23	M24	cella for- mátum	M25	M26
#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!	IGAZ	1919.12.14
január	január 3, 1959	21553	1959.01.03	HAMIS	1959.01.03
#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!	#ÉRTÉK!	IGAZ	1946.12.28
január	január 29, 1961	22310	1961.01.29	HAMIS	1961.01.29
június	június 1, 1992	33756	1992.06.01	HAMIS	1992.06.01
április	április 30, 1912	4504	1912.04.30	HAMIS	1912.04.30
március	március 15, 1920	7380	1920.03.15	HAMIS	1920.03.15
március	március 4, 1991	33301	1991.03.04	HAMIS	1991.03.04
június	június 25, 1988	32319	1988.06.25	HAMIS	1988.06.25
május	május 29, 1948	17682	1948.05.29	HAMIS	1948.05.29
július	július 10, 1990	33064	1990.07.10	HAMIS	1990.07.10

**7. táblázat:**

*A dátum (Statehood) konvertálásához használt képletek  
visszaadott értéke*

M22 {=INDEX(B53:B64;  
HOL.VAN(BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);A53:A64;0))}  
M23 {=HELYETTE(M2:M51;BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);  
INDEX(B53:B64;HOL.VAN(BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);  
A53:A64;0)))}  
M24 {=HELYETTE(M2:M51;BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);  
INDEX(B53:B64;HOL.VAN(BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);  
A53:A64;0))) \*1 }

**Algoritmus: Dátum típusának ellenőrzése**

Annak ellenőrzésére, hogy a sikerült-e a dátum automatikus felismerése vagy sem használhatjuk az M18 képlet szóköz keresését.

- Rákeresünk a szóközre.
- Megnézzük, hogy hibával tért-e vissza a függvény vagy a szóköz pozíciójával.
- A hiba kezelése.

**Kódolás: Dátum típusának ellenőrzése**

- A hiba ellenőrzését a hibás() függvénnyel el tudjuk végezni. Ha az eredeti dátum sztring, akkor a szöveg.keres() megtalálja a szóközt és visszaadja a pozícióját, tehát ezekben az esetekben a hibás() függvény hamis értéket ad vissza. Ha már megtörtént az automatikus típus felismerés, akkor nincs szóköz a kifejezésben és a hibás() függvény igaz értéket ad vissza (M25).
- A hiba kezelésére a ha() függvényt használhatjuk (M26)
  - a függvény kérdése az M25 képlet,
  - az igaz ágon az eredeti dátumot adjuk vissza,
  - a hamis ágon az M24 képlet visszaadott értékét.

M25 {=HIBÁS(SZÖVEG.KERES(" ";M2:M51))}

M26 {=HA(HIBÁS(SZÖVEG.KERES(" ";M2:M51));  
M2:M51;  
HELYETTE(M2:M51;BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);  
INDEX(B53:B64;HOL.VAN(BAL(M2:M51;SZÖVEG.KERES(" ";M2:M51)-1);  
A53:A64;0))) \*1)}

**3.2.1.4. Államok területe****A területek jellemzői**

Az államok területe több sebből is vérzik:

- Egy cellában tárolták a mi2 és a km2 értékeket, amiket szét kell választani.
- Az oszlop tartalmaz egy 19 karakter hosszúságú nem nyomtató karakter sorozatot a mi2-től balra, amit szintén el kell távolítani.

**A km<sup>2</sup> jellemzői**

- Az eredeti sztring jobb oldalán helyezkedik el.
- Zárójelben vannak a számok.

	M27	M28	M29	M30	M31	M32	M33
700452420000000000052,420 (135,767)	35	27	8	135,767)	135,767	135767	135767
7005665384000000000665,384 (1,723,337)	38	28	10	1,723,337)	1,723,337	1723337	1723337
700456273000000000056,273 (145,746)	35	27	8	145,746)	145,746	145746	145746
700482278000000000082,278 (213,099)	35	27	8	213,099)	213,099	213099	213099
700440408000000000040,408 (104,656)	35	27	8	104,656)	104,656	104656	104656
700452378000000000052,378 (135,658)	35	27	8	135,658)	135,658	135658	135658
700435380000000000035,380 (91,634)	34	27	7	91,634)	91,634	91634	91634
70039616000000000009,616 (24,905)	33	26	7	24,905)	24,905	24905	24905
700442775000000000042,775 (110,787)	35	27	8	110,787)	110,787	110787	110787
700465496000000000065,496 (169,634)	35	27	8	169,634)	169,634	169634	169634
700497813000000000097,813 (253,335)	35	27	8	253,335)	253,335	253335	253335

**8. táblázat**

*A km<sup>2</sup> konvertálásához használt képletek visszaadott értéke*

**Algoritmus: A km<sup>2</sup> értékek kivágása**

- A számok különböző hosszúságúak, de
- valamennyi a nyitó zárójeltől jobbra helyezkedik el,
- az eredeti sztring jobb szélén,
- minden számot még követ egy záró zárójel, amit el kell távolítani.

**Kódolás: A km<sup>2</sup> értékek kivágása**

- A szám hosszának meghatározásához ismernünk kell az eredeti sztring hosszát (M27) és a nyitó zárójel pozícióját (M28), ezek különbsége fogja adni a szám hosszát (M29).
- Jobbról kivágunk ennyi karaktert (M30).
- Eltávolítjuk a záró zárójelet a sztring végéről egy bal() függvénnyel: M31. A bal függvény argumentumai a következők:

- a jobb() függvénnyel visszkapott sztring,
- jobb() függvénnyel visszkapott sztring hosszánál egy karakterrel rövidebb.
- A sztringből eltávolítjuk a vesszőket a helyette() függvénnyel, ahol a vesszőt cseréljük le az üres sztringre (M32).
- A bal függvénnyel visszaadott sztringet számmá alakítjuk egy szorzással (M33).

M27. {=HOSSZ(G2:G51)}

M28 {=SZÖVEG.KERES("";G2:G51)}

M29 {=HOSSZ(G2:G51)-SZÖVEG.KERES("";G2:G51)}

M30 {=JOBBSZÖVEG.KERES("";G2:G51)}

M31 {=BAL(JOBBSZÖVEG.KERES("";G2:G51));  
HOSSZ(JOBBSZÖVEG.KERES("";G2:G51))-1}

M32 {=HELYETTE(BAL(JOBBSZÖVEG.KERES("";G2:G51))-  
SZÖVEG.KERES("";G2:G51));  
HOSSZ(JOBBSZÖVEG.KERES("";G2:G51))-  
1);";";"}}

M33 {=HELYETTE(BAL(JOBBSZÖVEG.KERES("";G2:G51))-  
SZÖVEG.KERES("";G2:G51));  
HOSSZ(JOBBSZÖVEG.KERES("";G2:G51))-  
1);";";"}\*1}

### **A mi<sup>2</sup> jellemzői**

- Minden érték a 20. pozíción kezdődik,
- a nyitó zárójel előtt végződik 2 karakterrel.



	M34	M35	M36	M37	M38	M39
700452420000000000052,420 (135,767)	16	52,420 (135,767)	6	52,420	52420	52420
7005665384000000000665,384 (1,723,337)	19	665,384 (1,723,337)	8	665,384	665384	665384
700456273000000000056,273 (145,746)	16	56,273 (145,746)	6	56,273	56273	56273
700482278000000000082,278 (213,099)	16	82,278 (213,099)	6	82,278	82278	82278
700440408000000000040,408 (104,656)	16	40,408 (104,656)	6	40,408	40408	40408
700452378000000000052,378 (135,658)	16	52,378 (135,658)	6	52,378	52378	52378
700435380000000000035,380 (91,634)	15	35,380 (91,634)	5	35,38	3538	3538
70039616000000000009,616 (24,905)	14	9,616 (24,905)	5	9,616	9616	9616
700442775000000000042,775 (110,787)	16	42,775 (110,787)	6	42,775	42775	42775
700465496000000000065,496 (169,634)	16	65,496 (169,634)	6	65,496	65496	65496
700497813000000000097,813 (253,335)	16	97,813 (253,335)	6	97,813	97813	97813

9. táblázat

A  $mi^2$  konvertálásához használt képletek visszaadott értéke

**Algoritmus: A  $mi^2$  értékek kivágása**

- Az eredeti sztring jobb oldaláról a 20. karaktertől kezdődően kivágunk minden karaktert,  $mi^2$  és  $km^2$  értékeit is.
- Az így kapott sztring bal oldaláról kivágjuk a számot, a  $mi^2$  értéket.
- A vesszőket eltávolítjuk.
- Végül a sztringet számmá alakítjuk.

**Kódolás: A  $mi^2$  értékek kivágása**

- A jobb() függvénnyel kivágjuk az eredeti sztring végén található két számot: M35. A jobb() függvény argumentumai:
  - eredeti sztring,
  - az eredeti sztring hosszánál 19 karakterrel kevesebb.
- A bal() függvénnyel levágjuk a rövidített sztring bal széléről a  $mi^2$  értékeket (M37). A bal() függvény argumentumai:
  - rövid sztring,

- a szám hossza, amit megkapunk, ha kivonjuk a sztring hosszából a nyitó zárójel pozícióját és csökkentjük az értéket a szóköz és a nyitózároljel karakterek hosszával. A képlet rövidebb lesz, ha ennek az értéknek a meghatározásához az eredeti sztringet használjuk (M36).
- Vesszők eltávolítását a helyette függvénnyel végezzük el, ahol a vesszőt az üres sztringre cseréljük (M38).
- Sztring számmá alakítása egy szorzással (M39).

M34 {=HOSSZ(G2:G51)-19}

M35 {=JOB(B(G2:G51;HOSSZ(G2:G51)-19))}

M36 {=HOSSZ(G2:G51)-SZÖVEG.KERES("(";G2:G51)-2}

M37 {=BAL(JOB(B(G2:G51;HOSSZ(G2:G51)-19);  
HOSSZ(G2:G51)-SZÖVEG.KERES("(";G2:G51)-2))}

M38 {=HELYETTE(BAL(JOB(B(G2:G51;HOSSZ(G2:G51)-19);  
HOSSZ(G2:G51)-SZÖVEG.KERES("(";G2:G51)-2);",";""))}

M39 {=HELYETTE(BAL(JOB(B(G2:G51;HOSSZ(G2:G51)-19);  
HOSSZ(G2:G51)-SZÖVEG.KERES("(";G2:G51)-2);",";""))\*1}

### 3.2.2. Sprego feladatok: Ötöslottó (24. óra)

Az ötöslottó Excel táblázatának konvertálása elvégezhető táblázatkezelő függvényekkel is. Ennek algoritmusát és menetét mutatjuk be az alábbiakban. A fájl 1957. 10. hetétől tartalmazza a kihúzott számokat, 1998. 1. hetétől már a nyereményeket és a nyertes szelvények számát, valamint 2004. 45. hetétől a húzás dátumát is (7. ábra). A hiányzó dátumok celláit üresen hagyták, míg a nyereményekre vonatkozó hiányzó adatokat 0-val pótolták. Tehát a fájl óriási, az utolsó adat a 2892-es sorban található a minta szerinti letöltés alapján. A feldolgozáshoz mindenképpen érdemes a köztes sorokat elrejtetni, és csak az első és utolsó néhányat a képernyőn hagyni.

További jellegzetessége a fajlnak, hogy a dátumot és a nyereményre vonatkozó adatokat szöveges formában tárolták (7. ábra), ezért van szükség a konvertálásra (8. ábra).

7. ábra

Az ötösoltó állomány .xls verziója táblázatkezelőben megnyitva

8. ábra

Az ötösoltó .xls verziója konvertálás után

### 3.2.2.1. Konvertálás

A konvertálást egy új munkalapon végezzük el. Ennek a megoldásnak két előnye is van: egyrészt így az eredeti munkalap érintetlen marad, valamint figyelembe véve, hogy sok oszlopa van a táblázatnak, valamivel könnyebb a navigáció, ha megtartjuk az eredeti oszlopneveket (8. ábra).

#### A konvertálás algoritmusai:

- a dátumban a kötőjeleket lecseréljük pontra, majd dátummal alakítjuk a sztringet,
- a darabszámokat egész számmá alakítjuk,

- a nyeremények végéről eltávolítjuk □Ft karaktereket, majd egész számmá alakítjuk a sztringet.

Valamennyi konvertálási lépést tömbképlettel végezzük el, így minden oszlophoz egyetlen képletet kell létrehozni, amelyik bármikor módosítható másolgatások nélkül.

A kötőjel karakter pont karakterre cserélését a HELYETTE() függvénnyel végezzük (M43), a függvény kimenete egy sztring. Ezt követően a DÁTUMÉRTÉK() függvénnyel alakítjuk a sztringet dátummá (M44). A DÁTUMÉRTÉK() függvény kimenete egy egész szám vagy egy hibaüzenet, attól függően, hogy sikerült a konvertálás vagy sem. A hiba kezelésére megvizsgáljuk, hogy az eredeti táblázatban volt-e dátum megadva vagy nem (M45) Ahhoz, hogy ténylegesen dátum formátumban jelenjen meg a dátum át kell állítani a formátumot (*Kezdőlap* → *Szám* → *Cellák formázása* párbeszédablak *Szám* lap → *Kategória: Dátum* → Típus kiválasztása).

### 3.2.2.2. Konvertálás lap feltöltése az eredeti lap adatai alapján

Az *eredeti* lap adatait oszloponként átmásoljuk a *konvertálás* lapra. A feladatot érdemes tömbképlettel megoldani, mivel így elkerüljük a másolást. Az év, a hét és a dátum vektorok másolására használható tömbképletet az M40, M41 és M42 képletek mutatják.

M40 {=EREDETI!A2:A2892}

M41 {=EREDETI!B2:B2892}

M42 {=EREDETI!C2:C2892}

Azon oszlopok esetében, amelyekben az adatok másolása során további átalakításra is szükség van, a tömbképletek tovább bővíthetők.

### 3.2.2.3. Dátum-sztring dátummá konvertálása

#### Algoritmus

- A dátum-sztring dátummá konvertálása során a dátum formátummal nem kompatibilis kötőjelet lecseréljük (M43).
- A sztringet dátummá konvertáljuk (M44).
- Hibakezelés: ismert dátum esetén kiíratjuk a konvertálás eredményét, míg hiányzó dátum esetén az üres sztringet (M45).

**Kódolás**

M43 {=HELYETTE(EREDETI!C2:C2892;"-";".")}

M44 {=DÁTUMÉRTÉK(HELYETTE(EREDETI!C2:C2892;"-";"."))}

M45 {=HA(EREDETI!C2:C2892="";"";DÁTUMÉRTÉK(HELYETTE(EREDETI!C2:C2892;"-";".")))}

**3.2.2.4. Darabszámok-sztring számmá konvertálása**

A nyertes szelvények darabszámának számmá konvertálását legegyszerűbben egy eggyel való szorzással végezhetjük el (M46).

M46 {=EREDETI!D2:D2892\*1}

**3.2.2.5. Forint**

A Ft konvertálásának algoritmus a egyszerű, ám nagyon gyakran találkozunk ezzel a problémával.

**Ft konvertálás algoritmus:**

- a sztring végéről eltávolítjuk a □Ft karaktereket (szóköz, F és t)
- a kivágott sztring hossza 3 karakterrel lesz rövidebb, mint az eredeti sztring, tehát elsőként az eredeti sztring hosszát kell meghatározni (M47)
- kiszámítjuk a kivágandó karakterek számát (M48)
- kivágjuk az új sztringet az eredeti bal széléről (M49)
- a sztringet számmá alakítjuk (M50)
- érdemes a szám formátumát Ft formátumra állítani

M47 {=HOSSZ(EREDETI!E2:E2892)}

M48 {=HOSSZ(EREDETI!E2:E2892)-3}

M49 {=BAL(EREDETI!E2:E2892;HOSSZ(EREDETI!E2:E2892)-3)}

M50 {=BAL(EREDETI!E2:E2892;HOSSZ(EREDETI!E2:E2892)-3)\*1}

**3.2.2.6. Darab, Ft és számok**

A többi *darab* és *Ft* oszlopokba bemásoljuk a képletet, míg a nyertes számok átmásolását az M51 képlet mutatja.

M51 {=EREDETI!L2:P2892}

### 3.3. Weblapok megnyitása és kezelése táblázatkezelőben (28. óra)

#### 3.3.1. Megyék

A Magyarország megyéi weblapon gyakorolhatjuk, hogy hogyan lehet táblázatkezelő függvények segítségével konvertálást végezni. Természetesen igaz, hogy táblázatkezelő programokban is használhatóak azok a cserék, amelyeket az előző fejezetekben mutattunk be, de ennek a fejezetnek a célja bemutatni, hogy a konvertálások függvények használatával hogyan végezhetők el. Módszertani szempontból kiemelt jelentőségű a módszer, egyrészt azért, mert így az eredeti adatok megmaradnak és a képletekkel számolt értékek a munkalap, munkafüzet újabb oszlopaiban kerülnek kiíratásra, másrészt a függvényeket használva az adatok nem válnak statikussá, így az eredeti adatok módosítása maga után vonja a függvényeken keresztül a számolt értékek módosítását is.

#### 3.3.1.1. Fölösleges sorok és oszlopok törlése, fájl konverzió

	A	B	C	D	E	F	G	H	I
1									
2	<b>Címer</b>	<b>Megye neve</b>	<b>Megveszélőhely</b>	<b>Terület (km<sup>2</sup>)</b>	<b>Népesség (fő)</b>	<b>Népsűrűség (fő/km<sup>2</sup>)</b>	<b>Település</b>	<b>Kistérség</b>	<b>Járás</b>
3		<a href="#">Bács-Kiskun megye</a>		8445	524 841 fő (2011. jan 1.)(2) ->	64	113	10	11
4		<a href="#">Baranya megye</a>		4430	391 455 fő (2011. jan 1.)(9) ->	91	301	9	10
5		<a href="#">Békés megye</a>		5031	301 802 fő (2011. jan 1.)(4) ->	70	73	8	9
6		<a href="#">Borsod-Abaúj-Zemplén megye</a>		7247	684 793 fő (2011. jan 1.)(5) ->	102	358	15	16
7		<a href="#">Csongrád megye</a>		4269	421 827 fő (2011. jan 1.)(6) ->	100	69	7	7
8		<a href="#">Fejér megye</a>		4359	426 120 fő (2011. jan 1.)(7) ->	97	108	10	9
9		<a href="#">Győr-Ménfőcsanak-Sopron megye</a>		4208	439 967 fő (2011. jan 1.)(8) ->	105	183	7	7
10		<a href="#">Hajdú-Bihar megye</a>		8211	339 674 fő (2011. jan 1.)(3) ->	89	82	9	10
11		<a href="#">Heves megye</a>		3697	307 985 fő (2011. jan 1.)(10) ->	89	121	7	7
12		<a href="#">Jász-Nagykun-Szolnok megye</a>		5582	380 732 fő (2011. jan 1.)(11) ->	74	78	7	9
13		<a href="#">Komárom-Esztergom megye</a>		2285	311 411 fő (2011. jan 1.)(12) ->	139	76	7	6
14		<a href="#">Nógrád megye</a>		1596	201 939 fő (2011. jan 1.)(13) ->	86	181	6	6
15		<a href="#">Pest megye</a>		6999	1 237 581 fő (2011. jan 1.)(14) ->	176	187	16	18
16		<a href="#">Somogy megye</a>		6096	317 947 fő (2011. jan 1.)(15) ->	65	245	11	8
17		<a href="#">Szabolcs-Szatmár-Bereg megye</a>		3926	552 000 fő (2012. jan 1.)(16) ->	98	229	12	13
18		<a href="#">Tolna megye</a>		3703	291 183 fő (2011. jan 1.)(17) ->	67	109	5	6
19		<a href="#">Vas megye</a>		3896	257 688 fő (2011. jan 1.)(18) ->	80	216	9	7
20		<a href="#">Veszprém megye</a>		4493	353 068 fő (2011. nov 1.)(19) ->	82	217	9	10
21		<a href="#">Zala megye</a>		3784	287 043 fő (2011. jan 1.)(20) ->	78	257	9	6
22									
23									
24									
25									
26									

9. ábra

A Magyarország megyéi weblap megnyitása táblázatkezelőben, majd a táblázat előtti és utáni sorok kitörlése után visszamaradt szöveg és egyéb objektumok

	A	B	C	D	E	F	G	H
1	Megye neve	Megyeszékhely	Terület	Népesség	Népsűrűség	Település	Kistérség	Járás
2			(km <sup>2</sup> )	(fő)	(fő/km <sup>2</sup> )			
3	Bács-Kiskun megye	Kecskemét	8445	524 841 fő (2011. Jan 1.)[2] +/-	64	119	10	11
4	Baranya megye	Pécs	4480	391 455 fő (2011. Jan 1.)[8] +/-	81	801	9	10
5	Békés megye	Békéscsaba	5631	361 882 fő (2011. Jan 1.)[4] +/-	70	75	8	9
6	Borsod-Abaúj-Zemplén megye	Miskolc	7247	684 793 fő (2011. Jan 1.)[5] +/-	102	358	15	16
7	Csongrád megye	Szeged	4263	421 827 fő (2011. Jan 1.)[6] +/-	100	60	7	7
8	Fejér megye	Székesfehérvár	4339	426 120 fő (2011. Jan 1.)[7] +/-	97	108	10	9
9	Győr-Ménfőcsanak-Sopron megye	Győr	4208	449 967 fő (2011. Jan 1.)[8] +/-	105	183	7	7
10	Hajdú-Bihar megye	Debrecen	6211	539 674 fő (2011. Jan 1.)[9] +/-	89	82	9	10
11	Heves megye	Eger	3637	307 985 fő (2011. Jan 1.)[10] +/-	89	121	7	7
12	Jász-Nagykun-Szolnok megye	Szolnok	5582	836 752 fő (2011. Jan 1.)[11] +/-	74	78	7	9
13	Komárom-Esztergom megye	Tatabánya	2265	211 411 fő (2011. Jan 1.)[12] +/-	129	76	7	6
14	Nógrád megye	Salgótarján	2546	201 919 fő (2011. Jan 1.)[13] +/-	86	131	6	6
15	Pest megye	Budapest	6993	1 287 561 fő (2011. Jan 1.)[14] +/-	176	187	16	18
16	Somogy megye	Kaposvár	4036	317 947 fő (2011. Jan 1.)[15] +/-	55	245	11	8
17	Szabolcs-Szatmár-Bereg megye	Nyíregyháza	5936	552 080 fő (2012. Jan 1.)[16] +/-	98	229	12	13
18	Tolna megye	Szekszárd	3703	231 183 fő (2011. Jan 1.)[17] +/-	67	109	5	6
19	Vas megye	Szombathely	3336	237 688 fő (2011. Jan 1.)[18] +/-	80	216	9	7
20	Veszprém megye	Veszprém	4433	333 088 fő (2011. nov 1.)[19] +/-	82	217	9	10
21	Zala megye	Zalaegerszeg	3784	287 043 fő (2011. Jan 1.)[20] +/-	78	257	9	6

### 10. ábra

*A Magyarország megyéi weblap szövegfájlként mentése és újra megnyitása utáni állapot*

#### 3.3.1.2. „megye” eltávolítása a megyenevekből

Az átalakított táblázat A oszlopba tartalmazza a megyeneveket, és minden név után még a „megye” szó is szerepel (10. ábra A oszlop), ami fölösleges, redundáns, nincsen további információtartalma, tehát érdemes eltávolítani. A megoldást az M52 képlet mutatja.

M52 {=BAL(A3:A21;HOSSZ(A3:A21)-6)}

Az eredeti *Megye neve* oszlopot vagy elrejtjük, vagy kitöröljük. Arra azonban figyelni kell, hogy az oszlop csak akkor távolítható el, ha az újonnan előállított megyenevekből készítünk egy másolatot irányított beillesztéssel, ahol a képletet nem, csak a tartalmat másoljuk át (11. ábra A oszlop), majd ezt követően törölhetjük ki a fölösleges oszlopot.

A *Megye neve* oszlop eredeti formájában annyira fölösleges, hogy véleményem szerint nem történik adatvesztés, ha kitöröljük (11. ábra).

#### 3.3.1.3. Népesség konvertálása táblázatkezelő függvényekkel

A Népesség oszlop egynél több adatot is tartalmaz, ezért van szükség további konverzióra. Ebben az adathalmazban leválasztható a népesség és az adatgyűjtés időpontja.

A népesség leválasztásához szükséges konvertálás algoritmus a két fő lépésből áll:

- a szám utáni karaktersorozat eltávolítása,
- a számban található szóközők eltávolítása.

A karaktersorozat eltávolításához közös tulajdonságokat kell keresni, olyanokat, amelynek valamennyi népsűrűség megfelel. Ebben az oszlopban a legkézenfekvőbb ilyen tulajdonság, hogy a számokat egy szóközzel elválasztva a „fő” követ. Az eredeti sztringből tehát annak bal oldalát kell megtartani, mindent, ami a „fő”-t megelőző szóköző előtt van. A balról megtartandó karakterek számát meg tudjuk határozni, ha ismerjük a „fő” pozícióját, amit a SZÖVEG.KERES() függvény ad vissza (M53).

M53 {=SZÖVEG.KERES("FŐ";D3:D21)}

Az M53 képlet outputja egy olyan vektor, amelyik tartalmazza a népsűrűség sztringben található „fő” sztring pozícióját. A képletet kívülről bővítve a BAL() függvénnyel ki tudjuk vágni az eredeti, hosszú sztringből a számot, ami a népességet adja vissza. Nem szabad azonban megfelelkezünk arról, hogy a BAL() függvény visszaadott értéke egy sztring, tehát nem egy „igazi” szám.

M54 {=BAL(D3:D21;SZÖVEG.KERES("FŐ";D3:D21)-2)}

Az M54 képlet BAL() függvényének visszaadott értéke egy olyan vektor, amely tartalmazza a népsűrűségeket szöveg formátumban. A szöveg számmá alakítása előtt azonban el kell távolítani a számokból a szóközőket. Arra mindenképpen érdemes odafigyelni, hogy ezek nem-törhető szóközők, amelyeket be kell másolni a függvény argumentumába. A HELYETTE() függvénnyel tudjuk a nem-törhető szóközőt lecserélni a semmire, az üres sztringre (M55). Ez a művelet a Csere parancsnak felel meg, azzal a különbséggel, hogy a függvények használata esetén az eredeti adatokon történő bármiféle módosítás maga után vonja az újraszámolást.

M55 {=HELYETTE(BAL(D3:D21;SZÖVEG.KERES("FŐ";D3:D21)-2);" ";"")}

A konvertálás legutolsó lépése a sztring számmá alakítása, amit egy szorzással tudunk legegyszerűbben elvégezni M56 (11. ábra I oszlop).

M56 {=HELYETTE(BAL(D3:D21;SZÖVEG.KERES("FŐ";D3:D21)-2);" ";"")\*1}



Arra azért nehéz magyarázatot találni, hogy a népesség és a terület alapján kiszámolt népsűrűség (M57) miért tér el a táblázatban eredetileg megadott népsűrűség értékektől (9. ábra F oszlop, 10. ábra E oszlop, 11. ábra E és J oszlop).

M57 {=KEREKÍTÉS(13:I21/C3:C21;0)}

	A	B	C	E	F	G	H	I	J	K	L	M
1	Megye neve	Megyeszékhely	Terület (km <sup>2</sup> )	Népsűrűség (fő/km <sup>2</sup> )	Település	Kistérség	Járás	Népesség (fő)	Népsűrűség (fő/km <sup>2</sup> )	dátumsztring1	dátumsztring2	dátum
3	Bács-Kiskun	Kecskemét	8445	64	119	10	11	524841	62	2011. Jan 1.	1-Jan-2011	2011.01.01
4	Banyma	Pécs	4430	31	301	9	10	391483	88	2011. Jan 1.	1-Jan-2011	2011.01.01
5	Békés	Békéscsaba	3921	70	73	8	9	301802	84	2011. Jan 1.	1-Jan-2011	2011.01.01
6	Borsod-Abaúj-Zemplén	Miskolc	7247	102	338	15	16	684793	94	2011. Jan 1.	1-Jan-2011	2011.01.01
7	Csongrád	Szeged	4263	100	60	7	7	421827	99	2011. Jan 1.	1-Jan-2011	2011.01.01
8	Fejér	Székesfehérvár	4839	97	108	10	9	426120	98	2011. Jan 1.	1-Jan-2011	2011.01.01
9	Győr-Ménfőcsanak-Sopron	Győr	4208	105	183	7	7	499957	107	2011. Jan 1.	1-Jan-2011	2011.01.01
10	Hajdú-Bihar	Debrecen	6211	69	62	9	10	539074	87	2011. Jan 1.	1-Jan-2011	2011.01.01
11	Hajdú-Bihar	Eger	3927	89	121	7	7	307965	85	2011. Jan 1.	1-Jan-2011	2011.01.01
12	Jász-Nagykun-Szolnok	Szolnok	5932	74	78	7	9	386752	89	2011. Jan 1.	1-Jan-2011	2011.01.01
13	Komárom-Esztergom	Tatabánya	2265	139	76	7	6	311411	137	2011. Jan 1.	1-Jan-2011	2011.01.01
14	Nógrád	Salgótarján	2346	86	181	6	6	201919	79	2011. Jan 1.	1-Jan-2011	2011.01.01
15	Pest	Budapest	6883	176	187	16	18	1237561	184	2011. Jan 1.	1-Jan-2011	2011.01.01
16	Somogy	Kaposvár	6089	35	245	11	8	317947	33	2011. Jan 1.	1-Jan-2011	2011.01.01
17	Szabolcs-Szatmár-Bereg	Nyíregyháza	5935	98	229	12	13	552000	93	2012. Jan 1.	1-Jan-2012	2012.01.01
18	Tolna	Szekszárd	3703	67	109	5	6	231183	62	2011. Jan 1.	1-Jan-2011	2011.01.01
19	Vas	Szombathely	3885	80	216	9	7	257688	77	2011. Jan 1.	1-Jan-2011	2011.01.01
20	Veszprém	Veszprém	4493	82	217	9	10	393068	79	2011. nov 1.	1-nov-2011	2011.11.01
21	Zala	Zalaegerszeg	3794	78	237	9	6	287043	76	2011. Jan 1.	1-Jan-2011	2011.01.01

11. ábra

A Magyarország megyei táblázaton a képletekkel végzett konvertálás eredménye

### 3.3.1.4. Az adatgyűjtés dátumának konvertálása táblázatkezelő függvényekkel

Az eredeti weblapon az adatgyűjtés dátumát is a Népesség oszlop tartalmazza. Ennek leválasztása is elvégezhető táblázatkezelő függvényekkel. A probléma nem más, mint a nyitó és a záró zárójel közötti sztring kivágása az eredeti sztringből – dátumsztring1, majd ennek a sztringnek a dátummá konvertálása. Az algoritmus a következő:

- a dátumsztring1 kivágása
  - kivágjuk az eredeti sztring nyitó zárójeltől jobbra lévő karaktereit (M59),
  - kivágjuk az előző lépésben kapott sztring bal oldalán található karaktereket a záró zárójelig (M60).

Ennek a konvertálásnak végeredménye egy dátumsztring (dátumsztring1, 11. ábra K oszlop). Ez a sztring azonban, sajnos, nem alakítható át egyetlen lépésben dátum formátumra (11. ábra M oszlop). A dátumsztring1 szétbontását az alapján végezzük, hogy a dátummá konvertáló

DÁTUMÉRTÉK() függvény milyen argumentumokat fogad el. Ennek a függvénynek egyetlen argumentuma van, 1-jan-2011 alakban<sup>4</sup> (11. ábra L oszlop).

- a dátumsztring1 dátum formátumra konvertálás
  - leválasztjuk a napot (M61),
  - leválasztjuk az évet (M62),
  - leválasztjuk a hónapot (M63),
  - dátummá konvertálás függvénnyel (M65).

Az algoritmus kódolását a következő képletek tartalmazzák.

A kivágandó karakterek számát a nyitó zárójel pozíciója adja vissza, amit SZÖVEG.KERES() függvény ad vissza (M58).

M58 {=SZÖVEG.KERES("(";D3:D21)}

A nyitó zárójel pozícióját ismerve ki tudjuk vágni az zárójeltől jobbra lévő teljes sztringet (M59).

M59 {=JOBBS(D3:D21;HOSSZ(D3:D21)-SZÖVEG.KERES("(";D3:D21))}

A JOBS() függvénnyel visszkapott sztring bal oldali karaktereit kell megtartani a záró zárójelig (M60).

M60 {=BAL(JOBS(D3:D21;HOSSZ(D3:D21)-SZÖVEG.KERES("(";D3:D21)); SZÖVEG.KERES("(";JOBBS(D3:D21;HOSSZ(D3:D21)-SZÖVEG.KERES("(";D3:D21)))-1)}

Ezzel a két lépéssel visszakaptuk a dátumsztringet 2011. jan 1. formátumban (dátumsztring1). Ennek dátummá konvertálását végezzük majd a DÁTUMÉRTÉK() függvénnyel. A függvény egyetlen argumentuma három értéknek az összefűzése lesz: nap-hónap-év.

#### **Nap leválasztása:**

M61 {=BAL(JOBS(N3:N21;2))}

---

<sup>4</sup> Felhívnam a figyelmet, hogy az összes magyar súgó hibásan adja meg a DÁTUMÉRTÉK() függvény sztringjét (nem kell a pont a hónap neve után). <http://office.microsoft.com/hu-hu/excel-help/datumertek-fuggveny-HA102753065.aspx>. Letöltve: 2014. augusztus 7.

### Hónap leválasztása:

M62 {=BAL(JOBB(N3:N21;HOSSZ(N3:N21)-6);3)}

### Év leválasztása:

M63 {=BAL(N3:N21;4)}

Az átalakított dátumsztring (dátumsztring2) létrehozását az M64 képlet mutatja. A képletben felhasználtuk az M61, M62 és M63 képleteket.



M64 {=BAL(JOBB(N3:N21;2))&"-"&  
BAL(JOBB(N3:N21;HOSSZ(N3:N21)-6);3)&"-"&  
BAL(N3:N21;4)}

Az 0 képlet által visszaadott sztring lesz a DÁTUMÉRTÉK() függvény argumentuma M65.

M65 {=DÁTUMÉRTÉK(BAL(JOBB(N3:N21;2))&"-"&  
BAL(JOBB(N3:N21;HOSSZ(N3:N21)-6);3)&"-"&  
BAL(N3:N21;4))}

A DÁTUMÉRTÉK() függvény visszaadott értéke egy egész szám, ami a 2008. január 1-jei dátum sorszámmegfelelője<sup>5</sup>. A sorszám dátum formátumban történő megjelenítését a formátum átállításával tudjuk elérni (*Kezdőlap* → *Szám* (Cellák formázás párbeszédablak Szám lapja) → *Kategória* → *Dátum*).

### 3.3.2. Best Books Ever

	A	B	C	D
214	15		<a href="#">Animal Farm</a>	
215			by George Orwell	
216			3.79 of 5 stars 3.79 avg rating — 1,314,368 ratings	Want to Read saving..
217			score: 709,723, and 7,428 people voted	
218				
219				
220				Error rating book. Refresh and try again.
221				Rate this book
222				Clear rating
223				1 of 5 stars 2 of 5 stars 3 of 5 stars 4 of 5 stars 5 of 5 stars

#### 12. ábra

*Best Books Ever* weblap megnyitása táblázatkezelő programban

<sup>5</sup> DÁTUMÉRTÉK függvény. <http://office.microsoft.com/hu-hu/excel-help/datumerterek-fuggveny-HA102753065.aspx>. Letöltve: 2014. augusztus 7.

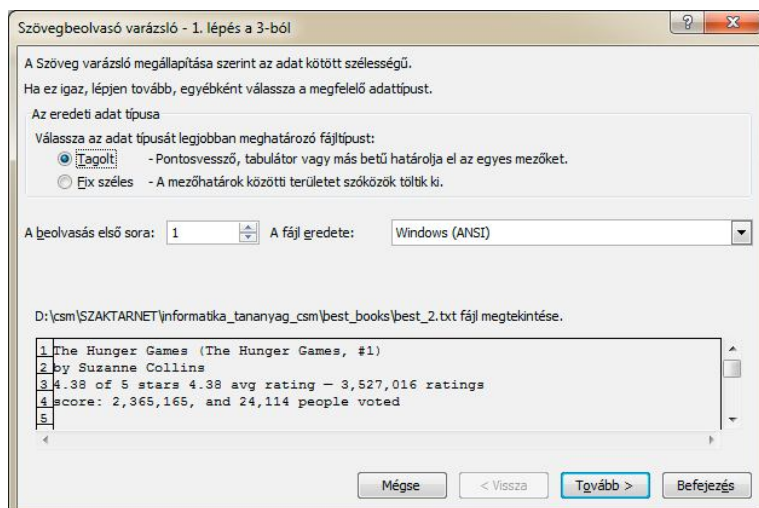
Bármennyire is ijesztőnek tűnik, a Best Books Ever weblapról letöltött táblázat<sup>6</sup> egy függvényekkel nagyon jól konvertálható fájl. A fölösleges sorok és oszlopok törlése után ez még inkább egyértelművé válik. Az egy könyvre vonatkozó adatok szerkezetét a 12 ábra – 16. ábra és a 10. táblázat mutatja. Minden könyv egy tíz soros adathalmaz, kivéve az első könyvet, de ezt az egyetlen egyet hozzá tudjuk igazítani a többihez.

Book	Row Range	Author	Avg Rating	Score	People Voted
Hunger Games	1-10	Katniss Everdeen	4.41	1,986,694	20,261
	11-20	Prince James	4.41	1,986,694	20,261
	21-30	Princess Snow	4.41	1,986,694	20,261
	31-40	Princess Snow	4.41	1,986,694	20,261
	41-50	Princess Snow	4.41	1,986,694	20,261
	51-60	Princess Snow	4.41	1,986,694	20,261
	61-70	Princess Snow	4.41	1,986,694	20,261
	71-80	Princess Snow	4.41	1,986,694	20,261
	81-90	Princess Snow	4.41	1,986,694	20,261
	91-100	Princess Snow	4.41	1,986,694	20,261
Harry Potter	11-20	J.K. Rowling	4.40	1,440,199	14,808
	21-30	J.K. Rowling	4.40	1,440,199	14,808
	31-40	J.K. Rowling	4.40	1,440,199	14,808
	41-50	J.K. Rowling	4.40	1,440,199	14,808
	51-60	J.K. Rowling	4.40	1,440,199	14,808
	61-70	J.K. Rowling	4.40	1,440,199	14,808
	71-80	J.K. Rowling	4.40	1,440,199	14,808
	81-90	J.K. Rowling	4.40	1,440,199	14,808
	91-100	J.K. Rowling	4.40	1,440,199	14,808
	101-110	J.K. Rowling	4.40	1,440,199	14,808
Divergent	993-1002	Randall	3.66	184,407	2,035
	1003-1012	Randall	3.66	184,407	2,035
	1013-1022	Randall	3.66	184,407	2,035
	1023-1032	Randall	3.66	184,407	2,035
	1033-1042	Randall	3.66	184,407	2,035
	1043-1052	Randall	3.66	184,407	2,035
	1053-1062	Randall	3.66	184,407	2,035
	1063-1072	Randall	3.66	184,407	2,035
	1073-1082	Randall	3.66	184,407	2,035
	1083-1092	Randall	3.66	184,407	2,035

### 13. ábra

*A Best Books Ever könyvei a munkalapon, a táblázat előtt és után elhelyezkedő fölösleges sorok és oszlopok törlése után*

<sup>6</sup> goodreads. Best Books Ever. [http://www.goodreads.com/list/show/1.Best\\_Books\\_Ever](http://www.goodreads.com/list/show/1.Best_Books_Ever).  
Letöltve: 2014. augusztus 7. Letöltve: 2015. június 29.



**14. ábra**

*A Best Books Ever szövegfájl megnyitása táblázatkezelő programban. A táblázatkezelő felajánlja a fix szélességű oszlopok létrehozását. Ezt azonban nem fogadhatjuk el, mert nincs szükségünk tagolásra, mivel minden adat egyetlen oszlopban van.*

A		A	
1	The Hunger Games (The Hunger Games, #1)	1	The Hunger Games (The Hunger Games, #1)
2	by Suzanne Collins	2	by Suzanne Collins
3	4.41 of 5 stars 4.41 avg rating — 2,779,503 ratings	3	4.38 of 5 stars 4.38 avg rating — 3,527,016 ratings
4	score: 1,986,694, and 20,261 people voted	4	score: 2,365,165, and 24,114 people voted
5		5	
6		6	
7		7	
8		8	
9		9	
10		10	
11	Harry Potter and the Order of the Phoenix (Harry Potter, #5)	11	Harry Potter and the Order of the Phoenix (Harry Potter, #5)
12	by J.K. Rowling	12	by J.K. Rowling
13	4.4 of 5 stars 4.40 avg rating — 1,205,636 ratings	13	4.42 of 5 stars 4.42 avg rating — 1,419,740 ratings
14	score: 1,440,199, and 14,808 people voted	14	score: 1,787,314, and 18,341 people voted
15		15	
16		16	
17		17	
18		18	
19		19	
20		20	

**15. ábra**

*A Best Books Ever könyvei a szövegfájl alakítás után (Letöltés 2014. augusztus 7. és 2015. június 29.)*

1. sor	cím	B
2. sor	szerző	C
3. sor	rating	D
	ratings	E
4. sor	score	F
	people	G
5–10. sor	üres	

**10. táblázat***A Best Books Ever táblázat szerkezete*

A Best Books Ever táblázat a törlések után minden adatot az A oszlopban tárol, egymás alatt négy cellában. Ezt az egyetlen oszlopot kell szét-szedjük hat oszlopra, amit két nagyobb lépésben tudunk megtenni:

- az egymás alatt található négy cellát átkonvertáljuk négy oszlopra,
- azokat az oszlopokat, amelyek két adatot tartalmaznak szétbontjuk két-két további oszlopra.

**3.3.2.1. Sorok konvertálása oszlopokká**

A művelet végeredménye egy mátrix, amit a legegyszerűbben egy többértékű tömbképlettel tudunk előállítani.

Az algoritmus a következő:

- a könyvek minden első sorát tesszük az új vektorba közvetlenül egymás alá, tehát minden tízedik elemet az eredeti oszlop vektorból M66.
- az előző lépésben előállított vektort bővítjük mátrixszá, tehát a második elemek kerülnek a második oszlop és így tovább M67.

A mátrix előállításához az INDEX() függvény több értéket visszaadó verzióját érdemes használni. Mivel célunk egy oszlop vektor előállítása, ezért az INDEX() függvény oszlop indexét 0-ra állítjuk. A sor indexeket pedig a címek eredeti sor indexéből számoljuk ki, tudva, hogy az eredeti vektor minden tízedik sorában egy könyvcím található, és az elő cím az első sorban van. A könyvcímek vektorának előállítását az M66 képlet mutatja.

M66 {=INDEX(A1:A999;(SOR()-1)\*10+1;0)}

	A	B
1	The Hunger Games (The Hunger Games, #1)	The Hunger Games (The Hunger Games, #1)
2	by Suzanne Collins	Harry Potter and the Order of the Phoenix (Harry Potter, #5)
3	4.38 of 5 stars 4.38 avg rating — 3,527,016 ratings	To Kill a Mockingbird
4	score: 2,365,165, and 24,114 people voted	Twilight (Twilight, #1)
5		Pride and Prejudice
6		The Chronicles of Narnia (Chronicles of Narnia, #1-7)
7		Gone with the Wind
97		The Complete Stories and Poems
98		The Secret Life of Bees
99		Winnie-the-Pooh (Winnie-the-Pooh, #1)
100		Middlesex
101	The Book Thief	
102	by Markus Zusak (Goodreads Author)	
103	4.35 of 5 stars 4.35 avg rating — 841,647 ratings	
104	score: 765,258, and 7,976 people voted	

**16. ábra**

*Címek egymás alá gyűjtése a B oszlopba (M66)*

A teljes mátrix előállításához a sor index értékét kell módosítani, annak megfelelően, hogy hanyadik sorból kérjük az adatot. Mivel az adatok és az új oszlopok sorszáma között egy-egy megfeleltetés van, ezért az összeadással módosított érték nem egy konstans lesz a továbbiakban, hanem az oszlop sorszámából tudjuk kiszámolni (M67).

M67 {=INDEX(A1:A999;(SOR()-1)\*10+OSZLOP()-1;0)}

	B	C
1	The Hunger Games (The Hunger Games, #1)	by Suzanne Collins
2	Harry Potter and the Order of the Phoenix (Harry Potter, #5)	by J.K. Rowling
3	To Kill a Mockingbird	by Harper Lee
4	Twilight (Twilight, #1)	by Stephenie Meyer
5	Pride and Prejudice	by Jane Austen
6	The Chronicles of Narnia (Chronicles of Narnia, #1-7)	by C.S. Lewis
7	Gone with the Wind	by Margaret Mitchell
97	The Complete Stories and Poems	by Edgar Allan Poe
98	The Secret Life of Bees	by Sue Monk Kidd (Goodreads Author)
99	Winnie-the-Pooh (Winnie-the-Pooh, #1)	by A.A. Milne
100	Middlesex	by Jeffrey Eugenides

**17. ábra**

*A könyv címek (B oszlop) és a szerzők (C oszlop) egymás melletti oszlopokban (M67), egymás alá rendezve*

	C	D	E
1	by Suzanne Collins	4.38 of 5 stars 4.38 avg rating — 3,527,016 ratings	score: 2,365,165, and 24,114 people voted
2	by J.K. Rowling	4.42 of 5 stars 4.42 avg rating — 1,419,740 ratings	score: 1,787,314, and 18,341 people voted
3	by Harper Lee	4.23 of 5 stars 4.23 avg rating — 2,371,811 ratings	score: 1,461,528, and 15,115 people voted
4	by Stephenie Meyer	3.56 of 5 stars 3.56 avg rating — 2,974,649 ratings	score: 1,304,412, and 13,279 people voted
5	by Jane Austen	4.23 of 5 stars 4.23 avg rating — 1,597,491 ratings	score: 1,243,610, and 12,926 people voted
6	by C.S. Lewis	4.22 of 5 stars 4.22 avg rating — 320,550 ratings	score: 907,277, and 9,549 people voted
7	by Margaret Mitchell	4.26 of 5 stars 4.26 avg rating — 722,969 ratings	score: 890,265, and 9,174 people voted
97	by Edgar Allan Poe	4.37 of 5 stars 4.37 avg rating — 162,311 ratings	score: 142,483, and 1,681 people voted
98	by Sue Monk Kidd (Goodreads Author)	3.98 of 5 stars 3.98 avg rating — 789,528 ratings	score: 138,672, and 1,590 people voted
99	by A.A. Milne	4.32 of 5 stars 4.32 avg rating — 176,989 ratings	score: 133,516, and 1,616 people voted
100	by Jeffrey Eugenides	3.95 of 5 stars 3.95 avg rating — 426,055 ratings	score: 132,034, and 1,475 people voted

### 18. ábra

*Szerző (C oszlop), rating (D oszlop) és score (E oszlop) egymás melletti oszlopokban (M67)*

Arról nem szabad megfeledkezni, hogy a mátrix területét ki kell jelölni, hogy valamennyi érték megjelenjen a munkalapon. Jelen példában ez egy  $100 \times 4$ -es mátrix.

#### 3.3.2.2. Adatok szétbontása két oszlopra

Két olyan oszlop van, amelyeket tovább kell bontanunk további oszlopokra:

- a rating és ratings adatai, valamint a
- a score és a people voted adatai.

Mindkét esetben egymásba ágyazott szöveg függvényekkel tudjuk a problémát megoldani.

A rating és ratings szétválasztásának algoritmusai (M68–M76):

- rating értékek (M68–M71)
  - az oszlop kétszer is tartalmazza ugyanazokat a rating értékeket, amelyek közül csak az egyiket kell megtartani, az első egyszerűbb, tehát azt fogjuk kivágni (M69),
  - ez a szám az oszlop bal szélén helyezkedik el, az első szóköztől balra,
  - megkeressük a szóköz pozícióját (M68), és az így kapott egész számnál 1-gyel kisebb érték lesz a kivágandó karakterek száma,
  - tizedes pont cseréje tizedes vesszőre (nem kötelező, a beállításoktól függ) (M70),
  - szöveg számmá konvertálása (M71).
- ratings értékek (M72–M76)
  - a ratings értékek a cella jobb oldalán helyezkednek el, és minden esetben ugyanazt „—□” sztringet követik



- „—” pozíciójának megkeresése (M72)
- a sztring kivágása az eredeti sztringből, tudva, hogy az két karakterrel jobbra helyezkedik el, mint a „—” karakter (M73)
- a „ratings” sztring eltávolítása a végéről: kivágjuk a sztring bal oldalát, ismerve, hogy hány karakter fölösleges a sztring végén (M74)
- ezreselválasztó karakterek cseréje az üres sztringre (nem kötelező, a beállításoktól függ) (M75)
- szöveg számmá konvertálása (M76)

A rating tizedes tört előállításához szükséges képletek (M68–M71).

M68 {=SZÖVEG.KERES(" ";D1:D100)}  
 M69 {=BAL(D1:D100;SZÖVEG.KERES(" ";D1:D100)-1)}  
 M70 {=HELYETTE(BAL(D1:D100;SZÖVEG.KERES(" ";D1:D100)-1);".";")}  
 M71 {=HELYETTE(BAL(D1:D100;SZÖVEG.KERES(" ";D1:D100)-1);".";")\*1}

A ratings szám előállításához szükséges képletek (M72–M76)

M72 {=SZÖVEG.KERES("—";D1:D100)}  
 M73 {=JOBBD1:D100;HOSSZ(D1:D100)-SZÖVEG.KERES("—";D1:D100)-1)}

A ratings egész szám kivágásánál figyelni kell arra, hogy most már az M73 képlettel előállított csonkított sztringnek kell a hosszát meghatározni, majd ebből kivonni a fölösleges 8 karaktert (a sztringet előállított képletet csak másolni kell) (M74).

M74 {=BAL(JOBB(D1:D100;HOSSZ(D1:D100)-SZÖVEG.KERES("—";D1:D100)-1);HOSSZ(JOBB(D1:D100;HOSSZ(D1:D100)-SZÖVEG.KERES("—";D1:D100)-1))-8)}  
 M75 {=HELYETTE(BAL(JOBB(D1:D100;HOSSZ(D1:D100)-SZÖVEG.KERES("—";D1:D100)-1);HOSSZ(JOBB(D1:D100;HOSSZ(D1:D100)-SZÖVEG.KERES("—";D1:D100)-1))-8);".";")}  
 M76 {=HELYETTE(BAL(JOBB(D1:D100;HOSSZ(D1:D100)-SZÖVEG.KERES("—";D1:D100)-1);HOSSZ(JOBB(D1:D100;HOSSZ(D1:D100)-SZÖVEG.KERES("—";D1:D100)-1))-8);".";")\*1}

A score és people voted számok szétválasztásának algoritmus (M77–M86):

- a score számai az „and” sztringtől balra helyezkednek el (M78)
  - „and” sztring pozíciójának meghatározása (M77)
  - a score számai a „score” sztringtől jobbra helyezkednek el (M79)
  - ezreselválasztó karakterek eltávolítása (M80)
  - szöveg számmá konvertálása (M81)
- a people voted számai az „and” sztringtől jobbra helyezkednek el (M83)
  - „and” pozíciójának meghatározása (M82)
- a people voted számai a „people voted” sztringtől balra helyezkednek el (M84)
  - ezreselválasztó karakterek eltávolítása (M85)
  - szöveg számmá konvertálása (M86)

A score számának előállítás (M77–M81)

```
M77 {=SZÖVEG.KERES("AND";E1:E100)}
M78 {=BAL(E1:E100;SZÖVEG.KERES("AND";E1:E100)-3)}
M79 {=JOB(BAL(E1:E100;SZÖVEG.KERES("AND";E1:E100)-3);
      HOSSZ(BAL(E1:E100;SZÖVEG.KERES("AND";E1:E100)-3))-7)}
M80 {=HELYETTE(JOB(BAL(E1:E100;SZÖVEG.KERES("AND";E1:E100)-3);
      HOSSZ(BAL(E1:E100;SZÖVEG.KERES("AND";E1:E100)-3))-7);";"")}
M81 {=HELYETTE(JOB(BAL(E1:E100;SZÖVEG.KERES("AND";E1:E100)-3);
      HOSSZ(BAL(E1:E100;SZÖVEG.KERES("AND";E1:E100)-3))-7);";"")*1}
```

A people voted számainak előállítás (M82–M86)

```
M82 {=SZÖVEG.KERES("AND";E1:E100)}
M83 {=JOB(E1:E100;HOSSZ(E1:E100)-SZÖVEG.KERES("AND";E1:E100)-3)}
M84 {=BAL(JOB(E1:E100;HOSSZ(E1:E100)-SZÖVEG.KERES("AND";E1:E100)-3);
      HOSSZ(JOB(E1:E100;HOSSZ(E1:E100)-SZÖVEG.KERES("AND";E1:E100)-3))-13)}
```

M85 {=HELYETTE(BAL(JOBB(E1:E100;HOSSZ(E1:E100)-SZÖVEG.KERES("AND";E1:E100)-3);HOSSZ(JOBB(E1:E100;HOSSZ(E1:E100)-SZÖVEG.KERES("AND";E1:E100)-3))-13);";";"")}

M86 {=HELYETTE(BAL(JOBB(E1:E100;HOSSZ(E1:E100)-SZÖVEG.KERES("AND";E1:E100)-3);HOSSZ(JOBB(E1:E100;HOSSZ(E1:E100)-SZÖVEG.KERES("AND";E1:E100)-3))-13);";";"")\*1}

Szerző nevétől balra a fölösleges karakterek törlése (F oszlop) (M87)

M87 {=JOB(C1:C100;HOSSZ(C1:C100)-3)}

A konvertálások elvégzése után érdemes a 6 oszlopot egy új lapra másolni (M88–M89).

A B oszlopban a címeket nem kellett módosítani, ezért azok maradtak az eredeti helyükön M88, M89.

M88 {=BEST\_BOOKS!B1:B100}

A másik öt egymás mellett lévő oszlop konvertálás során keletkezett (F–J oszlopok), így azok egy további képlettel átmásolhatók az új munkalapra.

M89 {=BEST\_BOOKS!F1:J100}



## *Irodalomjegyzék*

- A magyar helyesírás szabályai (AkH) (2005) Tizenegyedik kiadás. Tizenkettedik lenyomat: 2000. Változatlan lenyomat: 2005. Akadémiai Kiadó, Budapest. <http://mek.oszk.hu/01500/01547/index.phtml> (2011. január 3.)
- Abraham, R. és Erwig, M. (2009). Mutation Operators for Spreadsheets. Software Engineering, IEEE Transactions on Volume: 35, Issue: 1.
- Anführungszeichen. <http://www.duden.de/sprachwissen/rechtschreibregeln/anfuhrungszeichen>
- Antalné Szabó Ágnes: Az anyanyelvi nevelés új stratégiái, Magyar Nyelvőr 127.: (4.) pp. 407–427.
- Bakk Gyuláné (1998). Számítógéppel az iratok birodalmában. Nemzeti Tankönyvkiadó. Budapest.
- Balázs Géza (2007). Az informatika hatása a nyelvre. 11–45. In: Hódi Éva szerk.: Szarvas Gábor Nyelvművelő Napok, Ada, 2007. okt. 11–13. Szarvas Gábor Nyelvművelő Egyesület, Ada, 2008. <http://szgnye.vmmi.org/balazs2007.htm>.
- Baum, J. (é. n.): Learning Computer Keyboard Skills. Letöltés: [http://www.ehow.com/how-does\\_4811965\\_learning-computer-keyboard-skills.html](http://www.ehow.com/how-does_4811965_learning-computer-keyboard-skills.html). (2014.04.04.)
- Ben-Ari, M. és Yeshno, T. (2006). Conceptual models of software artifacts. Interacting with Computers. 18. pp. 1336–1350.
- Ben-Ari, M. (1999). Bricolage Forever! PPIG 1999. 11<sup>th</sup> Annual Workshop. 5–7 January 1999. Computer-Based Learning Unit, University of Leeds, UK. Retrieved April 12, 2014 from <http://www.ppig.org/papers/11th-benari.pdf>. Letöltve: 2014. július 15.
- Biró Piroska és Csernoch Mária (2013a): Deep and surface structural metacognitive abilities of the first year students of Informatics. 4th IEEE International Conference on Cognitive Infocommunicaitons, Proceedings, Budapest, 521–526.
- Biró Piroska és Csernoch Mária (2013b): Elsőéves informatikushallgatók algoritmizáló készségei. XXIII. Nemzetközi Számítástechnika és Oktatás Konferencia – SzámOkt 2013, EMT, 154–159.
- Biró Piroska és Csernoch Mária (2014): Táblázatkezelés algoritmikus megközelítése. Interdiszciplináris pedagógia és a fenntartható fejlődés.

- VIII. Kiss Árpád Emlékkonferencia tanulmánykötete. Közlésre elfogadva.
- Bódi Zoltán (2004a). A világháló nyelve. Gondolat Kiadó, Budapest
- Bódi Zoltán (2004b). A világháló nyelve. Internetezők és internetes nyelvhasználat a magyar társadalomban. Gondolat Kiadó, Budapest. ISBN: 9639500518
- Borbély Mária (2013). Digitális írástudás, digitális kompetenciák a statisztikai adatok tükrében. TMT. 60: 3, 120–127.
- Bujdosó Gyöngyi és Csernoch Mária (2008) Problémák a szövegszerkesztés verseny- és ECDL-vizsgafeladataival kapcsolatban. Problems in Tests Papers on Word Processing of ECDL Exams and Competitions. XVIII. SzámOkt 2008. 122–128.
- Bujdosó Gyöngyi és Csernoch Mária (2009) Mondanivaló kiemelésének formai eszközei. Networkshop 2009 (2009. április 14–17., Szeged) kiadványa.
- Bujdosó Gyöngyi és Csernoch Mária (2014). Digitális írástudás, digitális nyelvhelyesség. Tudományos és Műszaki Tájékoztatás, 61. évfolyam (2014) 10. szám.
- Carroll M. J. és Mack L. R. (1984). Learning to use a word processor: by doing, by thinking and by knowing in Thomas, J.C and Schneider, M.L. (eds) Human Factors in Computer Systems Ablex, 13–51.
- Case, J., Gunstone, R. és Lewis, A. (2001). Students' metacognitive development in an innovative second year chemical engineering course, *Research in Science Education*, 31(3), pp. 331–355.
- Case, J. és Gunstone, R. (2002). Metacognitive development as a shift in approach to learning: an in-depth study. *Studies in Higher Education*, 27(4), 459–470.
- Case, J. és Gunstone, R. (2003). Approaches to learning in a second year chemical engineering course. *International Journal of Science Education*, 25(7), 801–819.
- Case, J. (2000). Students' perceptions of context, approaches to learning and metacognitive development in a second year chemical engineering course. Unpublished PhD, Monash University, Melbourne.
- Cox, M. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence*, 169 (2), 104–141.

- Csernoch, M. (1997). Methodological Questions of Teaching Word Processing. 3rd International Conference on Applied Informatics: Eger-Noszvaj, Hungary, August 25–28, 1997. pp. 375–382.
- Csernoch Mária (2009) Teaching word processing – the theory behind. *Teaching Mathematics and Computer Science*. 2009/1. pp. 119–137.
- Csernoch Mária (2010) Teaching word processing – the practice. *Teaching Mathematics and Computer Science*. 8/2 (2010). pp. 247–262.
- Csernoch Mária (2012). Introducing Conditional Array Formulas in Spreadsheet Classes. EDULEARN12 Proceedings. Barcelona, Spain. 2-4 July, 2012. Publisher: IATED, 7270–7279.
- Csernoch Mária (2014a). Az informatikai terminológia használata a tan-  
könyvekben. *Új Pedagógiai Szemle*.
- Csernoch Mária (2014b). Programozás táblázatkezelő függvényekkel. Sprego. Műszaki Könyvkiadó, Budapest.
- Csernoch Mária és Balogh László (2010). Algoritmusok és táblázatkezelés – Tehetséggondozás a közoktatásban az informatika területén. Magyar Tehetségsegítő Szervezetek Szövetsége, Budapest. [http://geniuszportal.hu/sites/default/files/16\\_kotet\\_net\\_color.pdf](http://geniuszportal.hu/sites/default/files/16_kotet_net_color.pdf). Letöltve: 2011. október 25.
- Csernoch Mária és Biró Piroska (2013a). Button-up technikák hatékonyságának vizsgálata informatika szakos hallgatók táblázatkezelés-oktatásában. Szerk: Kozma Tamás és Perjés István, *Új kutatások a neveléstudományokban 2012*, ELTE Eötvös Kiadó, 369–392.
- Csernoch Mária és Biró Piroska (2013b). Teachers’ Assessment and Students’ Self-Assessment on The Students’ Spreadsheet Knowledge. EDULEARN13 Proceedings July 1st-3rd, 2013 – Barcelona, Spain. Publisher: IATED. ISBN: 978-84-616-3822-2. pp. 949–956.
- Csernoch Mária és Biró Piroska (2013c). Spreadsheet misconceptions, spreadsheet errors. Hungarian Conference on Educational Research, Debrecen.
- Csernoch Mária és Biró Piroska (2015a): Sprego programming. *Spreadsheets in Education (eJSiE) Vol. 8: Iss. 1*. <http://epublications.bond.edu.au/cgi/viewcontent.cgi?article=1175&context=ejsie>, 14/03/2015.

- Csernoch Mária és Biró Piroska (2015b). Számítógépes problémamegoldás. *Tudományos és Műszaki Tájékoztatás*, 62. évfolyam (2015) 3. szám.
- Csernoch Mária és Bujdosó Gyöngyi (2009a). Quality text editing. *Journal of Computer Science and Control Systems*. 2/2 5–10.
- Csernoch Mária és Bujdosó Gyöngyi (2009b). Vizsga- és versenyfeladatok szövegbeviteli hibái és ezek következményei. *Új Pedagógiai Szemle*. 2009/1.
- Csíkos Csaba (2006). *Metakogníció. A tudásra vonatkozó tudás pedagógiája*. Műszaki Kiadó. Budapest.
- Database functions. [http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx#BMdatabase\\_functions](http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx#BMdatabase_functions). Letöltve: 2013. június 23.
- Digital literacy. The Guardian's campaign to upgrade computer science and IT in schools. <http://www.theguardian.com/education/series/digital-literacy-campaign>
- Digitalliteracy.gov Your destination for digital literacy resources and collaboration. <http://www.digitalliteracy.gov/>
- Dormann, T. & Frese, M. (1994). Error Training: Replication and the Function of Exploratory Behavior. *International Journal of Human-Computer Interaction* 6(4) pp. 365–372.
- Excel functions (alphabetical list) <http://office.microsoft.com/en-us/excel-help/excel-functions-alphabetical-list-HA010277524.aspx?CTT=3>. Letöltve: 2012. augusztus 8.
- Excel functions (by category) <http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx>. Letöltve: 2012. augusztus 8.
- Figueredo, L. és Varnhagen, C. K. (2006). Spelling and Grammar checkers: are they intrusive? *British Journal of Educational Technology*. Vol 37 N5 2006 DOI: 10.1111/j.1467-8535.2006.00562.x pp. 721–732.
- Frese, M. & Altmann, A. (1989). The Treatment of Errors in Learning and Training. in *Developing skills with Information Technology*. (eds.) Baibbridge, L. & Quintanilla S. A. R.. John Wiley & Sons Ltd.
- Frese, M., Brodbreck, F., Heinbokel, T., Mooser, C., Schleiffenbaum, E. & Thiemann, P. (1991). Errors in Training Computer Skills: On the Positive Function of Errors. *Human-Computer Interaction*, Volume 6, pp.77–93.



- German Punctuation. <http://german.about.com/library/weekly/aa031901b.htm>
- Glister, P. (1997). *Digital Literacy*. John Wiley & Sons, Inc.
- Gove, M. (2012). Michael Gove speech at the BETT Show 2012. Published 13 January 2012. <https://www.gov.uk/government/speeches/michael-gove-speech-at-the-bett-show-2012>. Letöltve: 2014. július 15.
- Gove, M. (2014). Michael Gove speaks about computing and education technology. Published 22 January 2014. <https://www.gov.uk/government/speeches/michael-gove-speaks-about-computing-and-education-technology>. Letöltve: 2014. július 15.
- Graf, D. SZEMLE Az élő nyelvnek írásban való megjelenítéséről. *Rendszer és stilisztika helyesírásban*. Gyula, 1992. 117 l. – *Helyesírás és nyelvtudomány*. Gyula, 1993. 118 l. – *Anyanyelvi nevelés és helyesírás*. Gyula, 1994–95. 134 l. – *Helyesírás és nyelvpolitika*. Gyula, 1996–97. 128 l.
- IEEE&ACM Report 2013. *Computer Science Curricula 2013 (2013): The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society*. Letöltés: <http://www.acm.org/education/CS2013-final-report.pdf>. (2014.04.04.)
- Informatics education: Europe cannot afford to miss the boat. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education April 2013. Letöltés: <http://germany.acm.org/upload/pdf/ACMandIEreport.pdf>. (2014.02.02.)
- Javaslat a magyar nyelv alkotmányi szintű védelmével kapcsolatban, Javaslatok – A készülő új Alkotmány kodifikációs folyamatához. Köztársasági Elnöki Hivatal, 2010. 17. oldal (2010. november 12.) [http://www.parlament.hu/biz/aeb/info/kozt\\_elnok.pdf](http://www.parlament.hu/biz/aeb/info/kozt_elnok.pdf)
- Jorgensen, H. (2013). How not to Excel in economics Retrieved June 7, 2013 from <http://www.lowyinterpreter.org/post/2013/04/18/How-not-to-Excel-in-economics.aspx>
- Jury, D. (2004). *About Face: Reviving the Rules of Typography*. RotoVision. ISBN-10: 2880467985. ISBN-13: 978-2880467982.
- Jury, D. (2006). *What is Typography?* RotoVision. ISBN-10: 2888931036. ISBN-13: 978-2888931034.
- Jury, D. (2007). *Mi az a tipográfia?* SCOLAR KFT. ISBN: 9789639534773.

- Kadijevich, D. (2009). Simple spreadsheet modeling by first-year business undergraduate students: Difficulties in the transition from real world problem statement to mathematical model. In M. Blomhøj and S. Carreira (Eds.), *Mathematical applications and modeling in the teaching and learning of mathematics: Proceedings of the 11th International Congress on Mathematical Education, Mexico*, 241–248.
- Kadijevich, D. (2013). Learning about spreadsheet. In Kadijevich Djordje, Angeli Charoula and Schulte Carsten (Eds.). (2013). *Improving Computer Science Education*. New York and London: Routledge; p. 19–33.
- Kerettanterv (2008). *Magyar Közlöny*, 20. sz. II. kötet, 2008. február 8.
- Kerettanterv (2013). Letöltés: <http://kerettanterv.ofi.hu/index.html>. (2013.08.30.)
- Keszler Borbála (2003). A magyar írásjelhasználat és Európa. *Magyar nyelvőr.* 127/1. szám 24–36. <http://www.c3.hu/~nyelvor/period/1271/127104.pdf> (2011. január 3.)
- Kis Balázs és Prószéky Gábor (1999). *Számítógéppel emberi nyelven*. Szak Kiadó.
- Koriat, A. és Levy-Sadot, R. (2000). Conscious and Unconscious Metacognition: A Rejoinder. *Consciousness and Cognition*. (9). 193–202.
- Kritikai érzékkel és társadalmi felelősséggel a PISA-eredményekről. MTA 2014. március 20. Akadémia Közoktatási Elnöki Bizottságának Konferenciája. [http://mta.hu/mta\\_hirei/kritikai-erzekkel-es-tarsadalmi-felelosseggel-a-pisa-eredmenyekrol-133797](http://mta.hu/mta_hirei/kritikai-erzekkel-es-tarsadalmi-felelosseggel-a-pisa-eredmenyekrol-133797). (2014.07.03.)
- Kruger, J. és Dunning, D. (1999). Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments. *Journal of Personality and Social Psychology* 77 (6): 1121–34.
- Kwak, J. (2013). The Importance of Excel. <http://baselinescenario.com/2013/02/09/the-importance-of-excel>. Letöltve: 2013. május 17.
- Language Specific Quoting and Quotation Marks. <http://www.witch.westfalen.de/csstest/quotes/quotes.html>
- Layton, L. (2013.10.14.): Elementary students learn keyboard typing ahead of new Common Core tests. *The Washington Post*. Letöltés: Elementary students learn keyboard typing ahead of new Common Core tests (2014.04.04.)

- Lazar, J. és Norcio, A. (2003). Training Novice Users in Developing Strategies for Responding to Errors When Browsing the Web. *International Journal of Human–Computer Interaction*, 15(3), pp. 361–377.
- Les guillemets.0 <http://www.la-ponctuation.com/guillemets.html>
- Math and trigonometry functions. [http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx#BMmath\\_and\\_trigonometry\\_functions](http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx#BMmath_and_trigonometry_functions). Letöltve: 2013. június 23.
- Meet the spreadsheet. (2013). <http://office.microsoft.com/en-001/excel-help/meet-the-spreadsheet-RZ101773335.aspx?section=2>. Letöltve: 2013. augusztus 2.
- Microsoft Digital Literacy. <http://www.microsoft.com/about/corporate-citizenship/citizenship/giving/programs/up/digitalliteracy/default.msp>
- Microsoft Excel 2010 function translations <http://wwwhome.ewi.utwente.nl/~trieschn/excel/excel.html>. Letöltve: 2012. augusztus 8.
- Nielsen, J. (1993). *Usability Engineering*, Academic Press, Boston, MA.
- Northstar Digital Literacy Project. <http://www.digitalliteracyassessment.org/index.php>
- OECD (2011). *PISA 2009 Results: Students on Line: Digital Technologies and Performance (Volume VI)*. <http://browse.oecdbookshop.org/oecd/pdfs/free/9811031e.pdf>. <http://dx.doi.org/10.1787/9789264112995-en>. Letöltve: 2014. július 15.
- Panko, R. és Aurigemma, S. (2010). Revising the Panko-Halverson taxonomy of spreadsheet errors. *Decis. Support Syst.* 49, 2 (2010): 235–244.
- Panko, R. (2008). What We Know About Spreadsheet Errors. *Journal of End User Computing's Special issue on Scaling Up End User Development.* (10)2, 15–21.
- Pearson, C. H. (2007). Array Formulas. <http://www.cpearson.com/excel/ArrayFormulas.aspx>. Letöltve: 2012. augusztus 8.
- Powell, S., Baker, K. és Lawson, B. (2008). A critical review of the literature on spreadsheet errors. *Decision Support Systems*, 46(1), 128–138.
- Powell, S., Baker, K. és Lawson, B. (2009a): Errors in operational spreadsheets. *Journal of Organizational and End-User Computing*, 1(3), 4–36.

- Powell, S., Baker, K. és Lawson, B. (2009b): Impact of errors in operational spreadsheets. *Decision Support Systems*, 47(2), 126–132.
- Quotation Marks and their Unicode numbers. <http://www.witch.westfalen.de/csstest/quotes/quotes-large.html>
- Quotation Marks. <http://grammar.ccc.commnet.edu/grammar/marks/quotation.htm>
- Quotation Marks. <http://www.grammarly.com/handbook/punctuation/quotation-marks/>
- Report of JPMorgan Chase & Co. (2012). Management Task Force. Regarding: CIO Losses. [http://files.shareholder.com/downloads/ONE/2272984969x0x628656/4cb574a0-0bf5-4728-9582-625e4519b5ab/Task\\_Force\\_Report.pdf](http://files.shareholder.com/downloads/ONE/2272984969x0x628656/4cb574a0-0bf5-4728-9582-625e4519b5ab/Task_Force_Report.pdf). Letöltve: 2013. május 17.
- W3 (2012): Az olasz, magyar, görög után most holland tanulmány a tudatlanság áráról. Feltöltve: 2012. március 12. <http://njszt.hu/ecdl/hir/20120312/az-olasz-magyar-gorog-utan-most-holland-tanulmany-a-tudatlansag-ararol>. Letöltve: 2013. június 7.
- Scaffidi, C., Shaw, M. és Myers, B. (2005). Estimating the Numbers of End Users and End User Programmers. In *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 207–214.
- Sestoft, P. (2010). Spreadsheet technology. Version 0.12 of 2012-01-31. IT University Technical Report ITU-TR-2011-142. IT University of Copenhagen, December 2011.
- Starr, L. (2005). Teaching Keyboarding: More Than Just Typing. *Education World*. Updated 06/14/2011. Letöltés: [http://www.education-world.com/a\\_tech/tech/tech072.shtml](http://www.education-world.com/a_tech/tech/tech072.shtml). (2013.08.09.)
- Statistical functions. [http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx#BMstatistical\\_functions](http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx#BMstatistical_functions). Letöltve: 2013. június 23.
- Szabó József (2005). A nyelvjárás és az irodalmi nyelv helye és szerepe német Svájcban, különös tekintettel az iskolai oktatásra. *Magyar Nyelvőr*. 129/1. szám 59–70. <http://www.c3.hu/~nyelvor/period/1291/129104.pdf> (2011. január 3.)
- Szikszainé Nagy Irma (2007) *Magyar stilisztika*. Osiris Kiadó Kft.
- Teo, T. és Tan, M. (1999). Spreadsheet Development and “What-if” Analysis: Quantitative versus Qualitative Errors, *Accounting Management and Information Technologies*, vol. 9, pp. 141–160.

- Tort, F. Blondel, F. és Bruillard, É. (2008). Spreadsheet Knowledge and Skills of French Secondary School Students. R. T. Mittermeir and M. M. Sysło (Eds.): ISSEP 2008, LNCS 5090, 305–316, 2008. Springer-Verlag Berlin Heidelberg.
- Tort, F. (2010). Teaching Spreadsheets: Curriculum Design Principles. In S. Thorne (Ed.), Proceedings of the EuSpRIG 2010 conference: Practical steps to protect organisations from out-of-control spreadsheets, pp. 99–110.
- Tóth Tamás (2001). A szövegszerkesztés alapjai. Nemzeti Tankönyvkiadó.
- Van Deursen A. és Van Dijk J. (2012). CTRL ALT DELETE. Lost productivity due to IT problems and inadequate computer skills in the workplace. Enschede: Universiteit Twente. Retrieved from: [http://www.ecdl.org/media/ControlAltDelete\\_LostProductivityLackofICTSkills\\_UniverstiyofTwente1.pdf](http://www.ecdl.org/media/ControlAltDelete_LostProductivityLackofICTSkills_UniverstiyofTwente1.pdf). 02. 01. 2014.
- Virágvölgyi Péter (2004). A tipográfia mestersége – számítógéppel. Osiris Kiadó, Budapest ISBN: 9633894778
- Walkenbach, J. és Wilcox, C. (2003). Putting basic array formulas to work <http://office.microsoft.com/en-us/excel-help/putting-basic-array-formulas-to-work-HA001087292.aspx?CTT=5&origin=HA001087290>. Letöltve: 2012. augusztus 8.
- Warren, P. (2004). Learning to program: spreadsheets, scripting and HCI, in Proceedings of the Sixth Australasian Conference on Computing Education – vol. 30, Darlinghurst, Australia, pp. 327–333.
- Welcome to Digital Literacy. <http://digitalliteracy.rosendigital.com/>
- Wilcox, C. és Walkenbach, J. (2003). Introducing array formulas in Excel. <http://office.microsoft.com/en-us/excel-help/introducing-array-formulas-in-excel-HA001087290.aspx>. Letöltve: 2013. február 2.
- Wing, J. M. (2006). Computational Thinking. March 2006/Vol. 49, No. 3 Communications of the ACM.
- Z. Karvalics László (2002). Két világító torony. 2002. március <http://www.varga.hu/VARGA%20CSABA/UJSAGCIKKEK/NEPSZAVA.htm> (2011. január 3.)



## Mellékletek

Sorolja fel, milyen hibákat tartalmaz az alábbi szövegrészlet! A táblázat első oszlopában nevezze meg hibát, a másodikban adja meg a hiba típusát, a harmadikban pedig a helyes megoldást! Az ábrán jelölje a hibákat az ábécé betűivel!

**2011--2012**

...GODDARD SKI CLUB MEMBERSHIP APPLICATION

Make check payable to "GODDARD SKI CLUB" → Tom Hartmann/ Code 553

and return SIGNED form and check to: → Goddard Space Flight Center  
Greenbelt, MD 20771

LAST-NAME \_\_\_\_\_ FIRST \_\_\_\_\_

MEMBER TYPE (check only one)

Current Member Renewal before 9/30/11 →
  New Member or Renewal after 9/30/11 →
  U.S. Citizenship (for on-site events) Yes No

Individual--\$15 →
  Individual--\$25
  Family--\$20 →
  Family--\$30
 Family Member Names \_\_\_\_\_

(Individuals who have legal guardianship in same household, children under 18, or spouse)

Hiba	Hiba típusa	Helyes megoldás	pont
a)			
b)			
c)			

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

**A verseny új időpontja: 2011. március 31. - április 01-02. (csütörtök - péntek - szombat)**

**Helyszín:** Harruckem János Közoktatási Intézmény  
 .....5700 Gyula, Szent-István u. 38.

**Jelentkezés új határideje: 2011. március 09. (szerda)**

**Részvételi díj:** 24.000 Ft/csapat (csapatonként: 5 fő + 1 fő csapatvezető),  
 .....6.000 Ft/fő minden további kísérő.

A részvételi díj befizetését a Harruckem János Közoktatási Intézmény címére, a következő számlaszámra kérjük teljesíteni: "VII. Országos Rendészeti Csapatverseny" megjelöléssel.

Harruckem János Közoktatási Intézmény (Gyula)  
 11733113-15347079

A versenyés megkezdésének kizárólagos feltétele a nevezési díj befizetése megtörténtének igazolása előzetesen, de legkésőbb a helyszínen. (pl.: feladóvevény, átutalásról fénymásolat stb.)  
 A befizetési bizonylatot a csapatvezetők érkezéskor a regisztráción adják le.  
 A részvételi díj tartalmazza a verseny idejére a szállás- és az étkezés költségét.

3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Release of Liability

PLEASE READ AND SIGN the following: "Winter sports such as Nordic skiing contain inherent risks and participation in those sports can cause injury or death. As a purchaser/user of this season pass"

Phone number: → → → → \_\_\_\_\_

Signature of legal guardian if under 18: → → → → → → → → \_\_\_\_\_

Pass fees	→	→	→	→	before Dec. 15	→	→	Starting Dec. 15
Adult pass	→	→	→	→	\$100	→	→	\$130
Junior (6-18)/team member	→	→	→	→	\$65	→	→	\$90
Dartmouth Student	→	→	→	→	\$65	→	→	\$90
Family (2 adults and children living at home)	→	→	→	→	\$200	→	→	\$235

10% Discount off of regular Adult Pass fee for Dartmouth College (and Professional Schools) employees, senior citizens (65 and older) and recognized ski team or club\*.  
 (This does not include family season passes).

\* NENSA - New England Nordic Ski Association recognized ski team or club, Ford Sayre, high



**Versenykiírás**

**A verseny megnevezése:**

Hernád völgye Amatőr Rallysprint verseny.

Szlovák–Magyar Rallysprint Bajnokság 2011.

**A verseny helye:** Pere település, Rallysprint Pálya (Hernád hídnál)–Rancs Moldava (Szepsi)

**Különböző pályákon lesz megtartva a verseny sorozat.**

A verseny sorozat 8 futamból áll, 4 Magyar pályán, és 4 Szlovák pályán.

A verseny távja: Pere, 4x2,5 km

A pálya talaja: Jó minőségű kavicsozott földút.

Nevezési díj: 7000,- HUF (gulyás + üdítő)

**Nevezések:** Levélben: 3860, Encs: Arany J. Út.10. vagy email-ben: [modern@skylan.hu](mailto:modern@skylan.hu)

Tel: 06/202-921-687 ..... 06/203-548-648

Belépő: Nincs

**Eső és sár esetén, Encs-Füged 2-számú pályán lesz a verseny megtartva!**

**Kategóriák**

.....I. Kategória: 0 – 1300 ccm-ig


.....II. Kategória: 1301 – 1600 ccm-ig

.....III. Kategória: 1601 – 2000 ccm-ig

**BOOK LIST - WORLD GEOGRAPHY**


**PLEASE BE ADVISED:** There are several controversial topics and authors included on this list. Most of these books are fictionalizations of historic happenings and are open to interpretation. They are NOT to be taken as "fact" but may be representations of the events the author is dealing with. The purpose of this book review is to introduce you to other ideas and viewpoints.

TITLE	AUTHOR	SUBJECT
Sarum	Rutherford	English history in fictional setting
Poland	Michener	Polish history – follows several fictional families
Centennial	Michener	Colorado territory – fact & fiction
The Source	Michener	Israel settlement – by archeological "dig"
The Covenant	Michener	South African history – through various groups





**Magyar Tehetségsegítő Szervezetek Szövetsége • Magyar Géniusz Projektiroda**

1111 Budapest, Karinthy Frigyes út 4-6. II/1.  
 Telefon: (06-1) 361-4094, 787-0596; Fax: (06-1) 688-4880  
 Honlap: [www.tehetssegpont.hu](http://www.tehetssegpont.hu) • Email: [info@tehetssegpont.hu](mailto:info@tehetssegpont.hu)





Magyar Géniusz Integrált Tehetségsegítő Program - Országos Tehetségsegítő Hálózat kialakítása TÁMOP-3.4.4-A/08/1-2009-0001

**Kedves Kollégák!**

## Kecskeméti TE Súlyemelő Szakosztály









**XIII. Messzi István Emlékverseny**  
**Versenyhelye: Messzi István Sportcsarnok, Kecskemét, Olimpia-u. 1/a**  
 Megközelítés: 1-es, 11-es és 22-es busz

**Tervezett versenyprogram**  
**11.00-óra férfi, serdülő, ifjúsági korcsoport**  
 Mérlegelés: 9.00-10.00-óráig

**Szállás-lehetőség:** Sport Hotel (Kecskemét, Izsáki út 15. Tel: 76/323-090, 496-822)  
 → → → Szauna Hotel (Kecskemét, Csabay-G. krt. 2. Tel: 76/329-633)  
 Korall Szálló (Kecskemét, Jász u. 32. Dobos Mihály 06/20/986-1619)

→ → → → → → → → Bálint Csaba  
 → → → → → → → → edző

**U-14 és Felnőtt női, férfi Országos Bajnokság**

**A verseny támogatója:** Fire Systems Tűzvédelem Kft., Hungaro-Lux Light Kft.,  
 ...Nagysáp Község Önkormányzata.

**A verseny ideje:** 2013. február 17. / vasárnap  
**Felnőtt korcsoport mérlegelés:** 12.00-12.30 óráig  
**Verseny kezdete:** 13.30 óra

Felnőtt: 2.000.- Ft/fő / kategória → 1.000.- Ft éves versenyzői díj / kártyadíj  
**Dorog, 2013. január 17.**

→ Tar Mihály → → → Puksa Ferenc → → → → Klányi János  
 ...MSSZ társelnök → ...DNC elnök → → → Sumo Szakosztály vezető

**Verseny-rendezője:**  
 .....MÁVTK-Triathlon-Club

**Verseny-helyszíne:**  
 .....Kaposvár – Toponár-Deseda tó

**Verseny-ideje:**  
 .....2013. 09. 22

**Verseny-központ:**  
 .....Deseda tó - alsó parkoló

**Általános tudnivalók:**

- Versenyen mindenki saját felelősségére indul!
- Kerekpáros fejevődi használat kötelező!
- Minden versenyző csak a saját korcsoportjában indulhat!
- A KRESZ szabályait kötelező betartani!
- Versenyen az MTSZ verseny szabályai érvényesek!

**Diakolimpia-Megyei-Döntő az Országos Szövetség szabályai szerint: részletes versenykiírás a [www.smdsz.hu/Forgatokonyvek/fejzet-alatt](http://www.smdsz.hu/Forgatokonyvek/fejzet-alatt).**

**Megyei Bajnokság:**  
 - felnőtt, összevont senior, összevont veterán korcsoportban, 65 év alatti verseny.

**Pedagógus-Országos Bajnokság:**  
 - nevezési érvényes pedagógus igazolvánnyal.

**Mindenki csak egy korcsoportban indulhat!**

**Várunk szeretettel minden indulót!**

**Nevezés:**

A helyszínen 8.30 órától folyamatosan!

**Korcsoportok:** ..... **Nevezési díj:**

- Nevezési díjat az „A” kategóriában indulóknak kell fizetni.
- I. kcs. .... 900 Ft
- II. kcs. .... 1000 Ft
- III. kcs. .... 1000 Ft
- IV. kcs. .... 1000 Ft
- V. kcs. .... 1200 Ft
- VI. kcs. .... 1200 Ft
- felnőtt → 74.92 → ..... 2000 Ft
- senior → 54.72 → ..... 2000 Ft
- veterán → 53 és előtte születettek → ..... 2000 Ft
- pedagógus I. + 1973 és előtte születettek → ..... 2000 Ft
- pedagógus II. + 1974 és utána születettek → ..... 2000 Ft

**Korcsoportok: Verseny táj: Rajt:**

- I. kcs. → 300 m - 2 km - 230 m → ..... 09.30 óra
- II. kcs. → 1 km - 4 km - 300 m → ..... 10.03 óra
- III. kcs. → 1 km - 4 km - 300 m → ..... 10.03 óra
- IV. kcs. → 2 km - 8 km - 1 km → ..... 10.50 óra
- V. kcs. → 2.5 km - 10 km - 1.5 km → ..... 11.45 óra
- VI. kcs. → 2.5 km - 10 km - 1.5 km → ..... 11.45 óra
- felnőtt → 5 km - 20 km - 2.5 km → ..... 12.45 óra
- senior → 5 km - 20 km - 2.5 km → ..... 12.45 óra
- veterán → 5 km - 20 km - 2.5 km → ..... 12.45 óra
- pedagógusok → 5 km - 20 km - 2.5 km → ..... 12.45 óra

**Díjazás:**  
 Korcsoportonként I-III. helyezettek érem jutalomában részesülnek.

**La nourriture, c'est for-miam-dable!**


En plus les vedettes tirent souvent avantage de leurs noms. Paul Newman a sa propre marque de produits alimentaires et tous les profits vont aux organismes charitables. Wayne Gretzky, Don Cherry et Doug Gilmour ont leurs propres restaurants. ...

D'autres vedettes font de la publicité pour des ...


↓ Casse-croûte: Ringo Starr des Beatles est dans → les annonces de Domino's Pizza, Joe Carter quitte un match de baseball pour → aller chez McDonalds pour un hamburger et Kristi Yamaguchi mangé des frites avec Dave chez Wendy's. Ah! La pizza, les hamburgers, et les frites! J'ai une faim du loup! Au revoir!

**La nourriture, c'est for-miam-diable!** ..... Nom: .....

Répondez aux questions en phrases complètes.

1. → Quels sont les trois repas du jour? 


.....

2. → Quel est un goûter populaire quand on regarde la télé? 

.....

**Vrai ou Faux? Encermez la bonne réponse.**

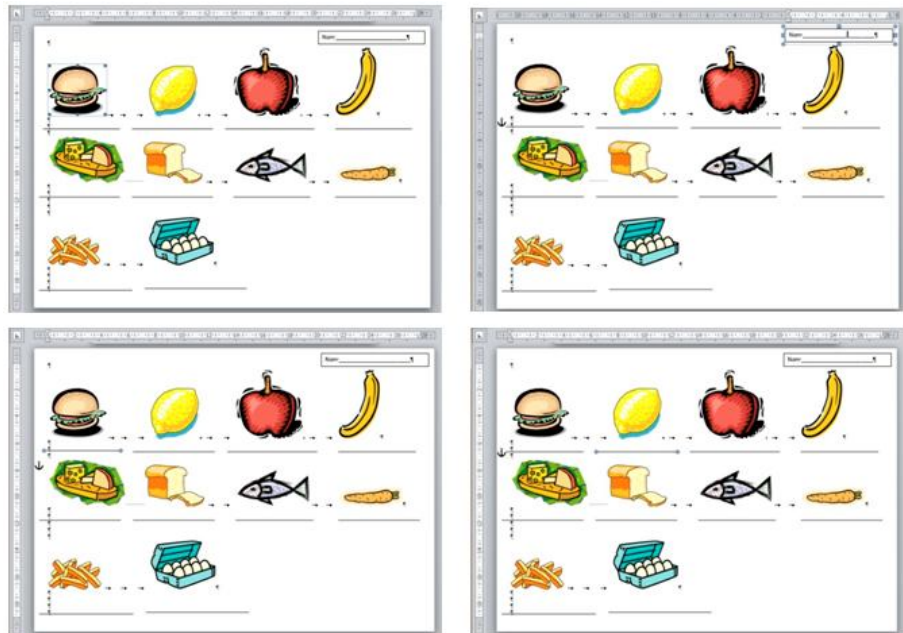
1. → On mange des chips quand on fait les devoirs → → → v → → f

2. → On mange de la pizza quand le prof ne regarde pas? → → → v → → f 

---

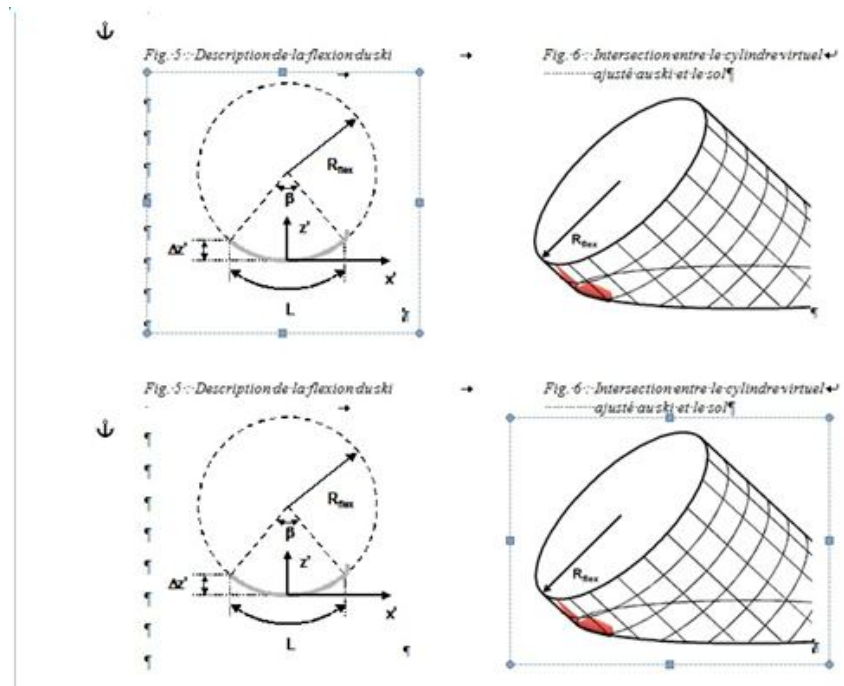
hiba megnevezése	hiba javítása	
Milyen formázási hibákat talál a képen?		
a)		
b)		
Milyen tördelési hibákat talál a képen?		
c)		
d)		
e)		
f)		
Milyen tipográfiai hibákat talál a képen?		
g)		
h)		
Milyen a kijelölt kép elrendezése és melyik bekezdéshez tartozik a kép?		
Melyik bekezdés formátumát mutatja a vonalzó? Indokolja választását!		

Válaszoljon a kérdésekre az ábra alapján!



- *Bal felső:* Nevezze meg a kijelölt objektum elrendezését és mondja meg, hogy helyes-e?
- *Jobb felső:* Melyik bekezdéshez tartozik a kijelölt objektum és rögzítve van-e? Karikázza be a kiválasztott bekezdés jelét!
- *Bal alsó:* Melyik bekezdéshez tartozik a kijelölt objektum és milyen az elrendezése? Karikázza be a kiválasztott bekezdés jelét!
- *Jobb alsó:* Melyik bekezdéshez tartozik a kijelölt objektum és rögzítve van-e? Karikázza be a kiválasztott bekezdés jelét!
- Mi lett volna a helyes megoldás a képek és a vonalak megjelenítésére?
- Mi lett volna a helyes megoldás a név megjelenítésére?

Válaszoljon az alábbi kérdésekre!



- Helyes-e a bal oldali kép elrendezése? Indokolja válaszát!
- Helyes-e a jobb oldali kép elrendezése? Indokolja válaszát!
- Milyen a bal oldali kép elrendezése? Indokolja válaszát!
- Melyik bekezdéshez tartozik a jobb oldali kép?
- Melyik bekezdéshez tartozik a bal oldali kép?
- Hogyan tudja eldönteni, hogy melyik bekezdéshez tartozik a kép?
- A képek hibás elrendezése milyen típusú hiba?
- Milyen tördelési hibát talál a mintán? Hogyan javítaná?
- Hogyan oldaná meg a feladatot helyesen?

Rajzolja meg a helyes megoldás vonalzóelrendezését! A megoldást a felső, felnagyított vonalzón készítse el!



– Írja le a vonalzón elvégzett beállításokat!

5002	ARTEMIA 100 ML BLISTER	2.00	1.40	0.88	0.74	0.70	0.66
5502	ARTEMIA 500 GR PLAAT/PLAQUE	6.30		3.96	3.34		-I
5003	BEEF HEART 100 ML BLISTER	2.00	1.40	0.88	0.74	0.70	0.66
5503	BEEF HEART 500 GR PLAAT/PLAQUE	6.30		3.96	3.34		-I
N555	BOSMIDEN 100 ML BLISTER	2.00	1.40	0.88	0.74	0.70	0.66

Rajzolja be a szükséges tabulátorokat a vonalzóra! Figyeljen arra, hogy egyértelmű legyen a tabulátor típusa!

- Hány oszlopos lesz a táblázat?
- Hány tabulátor elhelyezésére van szükség? Indokolja választát!

Válaszoljon a következő kérdésekre az ábra alapján! Indokolja válaszait!



- Helyes a kép elrendezése?
- Középre van igazítva a kép?
- Élőfejben van a kép?
- Melyik bekezdéshez tartozik a kép?
- Ezt az elrendezést használva lehetne a kép mellé szöveget írni?
- Ebben az esetben szükség van arra, hogy a kép mellé szöveget írjunk?
- Melyik elrendezést tartaná a legjobb megoldásnak?
- Megfelelőnek tartja a margóbeállításokat?
- Tördelési hibát lát-e a mintán? Ha igen, akkor mi ez és hogyan javítaná?
- Formázási hibát lát-e a mintán? Ha igen, akkor mi ez és hogyan javítaná?
- Tipográfiai hibát lát-e a mintán? Ha igen, akkor mi ez és hogyan javítaná?