



Tudástranszferelemek szerepe a hatékony számítógépes problémamegoldásban

Egyetemi doktori (PhD) értekezés

szerző neve: Horváthné Nagy Tímea
témavezető neve: Dr. Csernoch Mária

DEBRECENI EGYETEM
Természettudományi és Informatikai Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2024.

Ezen értekezést a Debreceni Egyetem Természettudományi és Informatikai Doktori Tanács Informatikai Tudományok Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programja keretében készítettem a Debreceni Egyetem műszaki doktori (PhD) fokozatának elnyerése céljából.

Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Debrecen, 2024. 03. 06

Horváthné Nagy Tímea
jelölt

Tanúsítom, hogy Horváthné Nagy Tímea doktorjelölt 2020–2024 között a fent megnevezett Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Az értekezés elfogadását javasolom.

Debrecen, 2024. 03. 06.

Csernoch Mária
témavezető

TUDÁSTRANSZFERELEMEK SZEREPE A HATÉKONY SZÁMÍTÓGÉPES PROBLÉMAMEGOLDÁSBAN

Értekezés a doktori (PhD.) fokozat megszerzése érdekében
Az Informatika didaktika tudományágban

Írta: Horváthné Nagy Tímea
okleveles középiskolai informatika–matematika tanár

Készült a Debreceni Egyetem Informatikai Tudományok doktori iskolája
(Elméleti számítástudomány, adatvédelem és kriptográfia programja)
keretében

Témavezető: Dr. Csernoch Mária

Az értekezés bírálói:

Dr.
Dr.
Dr.

A bírálóbizottság:

elnök: Dr.
tagok: Dr.
Dr.
Dr.
Dr.

Az értekezés védésének időpontja:

Tartalomjegyzék

1. Bevezetés	1
1.1. Célkitűzés	1
2. Elméleti megalapozás	3
2.1. Informatikaoktatás helyzete Magyarországon.....	3
2.1.1. Tartalmi-szabályozó dokumentumok	3
2.1.2. Informatikaoktatás a gyakorlatban	3
2.2. Módszertani eszközök	7
2.2.1. Táblázatkezelés.....	7
2.2.2. Szövegkezelés.....	9
2.3. Digitális fenntarthatóság.....	15
3. Mini-kompetenciateszt	18
3.1. A teszt felépítése.....	18
3.2. Táblázatkezelés.....	20
3.2.1. Képletkiegészítés	20
3.2.2. Műveleti sorrend.....	21
3.2.3. Adattípusok, adatelemzés	23
3.2.4. Tömbképlet.....	26
3.3. Szövegkezelés.....	27
3.4. Minta.....	28
4. Tézisek.....	29
5. Táblázatkezelés és önértékelések	30
6. Digitális eszközökre fordított idő	41
7. A grafikus információk értelmezése a szövegszerkesztésben	48
7.1. A mérés bemutatása.....	48
7.2. A vizsgálat eredményei	50
7.3. Az automatikus számozás entrópiája.....	53
8. A digitális szövegek entrópiája.....	57
8.1. Digitális szövegek entrópiájának elméleti háttere	57
8.1.1. Kapcsolat a kommunikációs entrópiával	57
8.1.2. Digitális szövegek entrópiájának kiszámítása	57

8.2. A minta	59
8.2.1. Gyógyszerek dokumentum.....	59
8.2.2. Frenchfood dokumentum.....	60
8.3. A módszer.....	61
8.4. Eredmények.....	64
8.4.1. Gyógyszerek dokumentum javítása.....	64
8.4.2. Gyógyszerek dokumentum formázása.....	66
8.4.3. Gyógyszerek dokumentum módosítása	66
8.4.4. Frenchfood dokumentum javítása	69
8.4.5. Frenchfood dokumentum formázása	70
8.4.6. Frenchfood dokumentum módosítása.....	71
9. Dokumentumok fenntarthatósági aránya.....	74
9.1. Elméleti háttér.....	74
9.2. A minta	76
9.3. A módszer.....	76
9.4. A fenntarthatósági arány alkalmazása	78
9.4.1. Fejezetek számozásának megváltoztatása	78
9.4.2. Számozott bekezdések beillesztése	80
10. Összefoglaló	83
11. Hosszútávú tervek.....	87
12. Köszönetnyilvánítás.....	89
13. Summary.....	90
14. Irodalomjegyzék.....	95
15. Függelék	111
16. Publikációs jegyzék	117

1. Bevezetés

Egy hatalmas iparág épül a végfelhasználókra (end-users), de aligha foglalkozik valaki azzal, hogy mennyire alkalmazzák hatékonyan és eredményesen mind a módszertanokat, mind a számukra kifejlesztett eszközöket, és mennyire hatékonyan használják ki az ezekre a tevékenységekre alkalmas erőforrásokat. Elsősorban a vállalatok – mind a hardvergyártók, mind a szoftvergyártók – és még sok más kísérő iparág profitál a végfelhasználókból, a digitális kompetenciák hiányából és az alacsony szintű számítógépes gondolkodási készségekből, képességekből. A kérdés azonban nem az, hogy egyes csoportok mekkora profitot termelnek, hanem az, hogy mennyi veszteség keletkezik, szinte észrevétlenül.

Az oktatás a végfelhasználói tevékenységek egyik elsődleges platformja. A cégek ugyan elárastották a végfelhasználókat eszközeikkel és felhasználóbarát szlogenjeikkel, de képtelenek voltak valódi felhasználóbarát módszereket létrehozni, támogatva ezzel a tanulás-tanítás folyamatát. Általánosságban elmondható, hogy a számítógépes oktatás egy toló (push) termelési rendszer, amely definíció szerint veszteséget termel [1]–[5]. Látszatra az oktatás sokat nyer, hiszen a köz-, és a tanfolyami oktatás, valamint az oktatási segédanyagok biztosítása egy hatalmas iparágat működtet. Másrészt az oktatás a legnagyobb vesztese ennek a helyzetnek, mivel a végfelhasználói tevékenységben és a termelésben nem mutatható ki jelentős előrehaladás.

Általában a végfelhasználók nem látják tisztán az újonnan megjelenő eszközök jelentőségét, alkalmazhatóságát, holott bizonyításra került, hogy a hatékony számítógépes gondolkodás nélküli oktatás miatt a legtöbb ember könnyen félrevezethető, és félre is vezetik őket [6].

A Toyota azt állítja, hogy jobb autókat akarnak építeni több ember számára [1]. Ennek mintájára, kutatócsoportunk azt a célt tűzte ki, hogy jobb (hatékonyabb) számítógépes oktatást nyújtunk több ember (tanuló, tanár, végfelhasználó) számára, amelyek a jövő munkahelyeihez szükségesek [7][8], nem csak a kiválasztott csoportok (pl. informatikusok) számára.

1.1. Célkitűzés

Tudományos kutatómunkámat 2018-ban kezdtem meg, az akkor érvényes Nemzeti Alaptanterv és a Kerettantervek elemzésével [9]. A magyarországi informatikaoktatás problémáinak és az elsőéves informatikus hallgatók teszt-eredményeinek [10]–[14] ismeretében mini-kompetenciatesztet terveztünk az

általános és középiskolások (7–10. évfolyam) informatikai képességeinek mérésére. A teszt célja az informatikaoktatás hatékonyságának vizsgálata, a konkrét tudáselemek azonosítása, valamint a tudástranszferelemek aktiválásának mérése.

Az eredményekkel rávilágíthatunk arra, hogy a számítógépes és számítógéppel támogatott oktatásban alkalmazott és széles körben elfogadott, eszközorientált, tudástranszferelemeket nélkülöző megközelítések alacsony hatékonyságú számítógépes problémamegoldáshoz vezetnek, melynek további következménye az informatikai felsőoktatásban az óriási hallgatói lemorzsolódás, az iparban az informatikus hiány, valamint a nem-professzionális informatikában keletkező óriási anyagi veszteségek, amelyek a hanyag adatkezelés következményei.

2. Elméleti megalapozás

2.1. Informatikaoktatás helyzete Magyarországon

2.1.1. Tartalmi-szabályozó dokumentumok

Magyarország azon országok közé tartozik a világon, ahol az informatikát, mint iskolai tantárgyat már a 90-es évek közepén bevezették [15]–[17] az első Nemzeti Alaptantervben (NAT) [18]. A tantárgyhoz időközben (2009-től) különböző elnevezések és óraszámok kerültek hozzárendelésre (1. táblázat) [18]–[21], az informatika kerettantervekben részletezett tartalmakban azonban csak kisebb változtatások történtek.

Korábbi mérések [9], tantermi tapasztalatok és tanári interjúk azt mutatták, hogy mind a jelenlegi (NAT 2020) [21], mind az előző (NAT 2012) [20] Nemzeti Alaptanterv és a magyar informatika kerettantervek [22]–[25] tartalma túlméretezett (még a legnagyobb óraszám esetén is), kétértelmű kifejezésekkel terhelt, eszközközpontú, szigorúan elkülönülő altémákra tagolt, továbbá a problémamegoldás kizárólag a programozásra korlátozódik.

	Évfolyamok												Össz.
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	
1995	nincsenek óraszámok megadva, csak intervallumok a többi tantárgyhoz viszonyítva												
2009	–	–	0,5	1	0,5	1	1	1	1,5	1	1,5	1	10
2012	–	–	–	–	–	1	1	1	1	1	–	–	5
2020	–	–	1	1	1	1	1	1	2	1	2	–	11

1. táblázat: Az informatikához rendelt heti órák száma a Nemzeti Alaptantervek szerint [18]–[21].

Wing javaslata alapján a számítógépes gondolkodásnak negyedik alapkészségként kellene megjelennie a magyar oktatási rendszerben is, az olvasás, az írás és a számolás mellett [26][27]. Ezekből a dokumentumokból nem derül ki azonban egyértelműen, hogy az informatikaoktatás hogyan támogatja az algoritmikus szemlélet kialakítását, illetve a tantárgyközi és az informatikán belüli tudástranszfert [27], tantárgyi integrációt.

2.1.2. Informatikaoktatás a gyakorlatban

A legtöbb informatikatanár a problémamegoldás helyett az eszközök és a felületek tanítására összpontosít [28]–[30]. Ennek a dekontextualizált [31], ala-

csony mathability megközelítésnek [28] a következménye, hogy a tanulók hibásan, lassú gondolkodású tevékenységeket végeznek [32] a magas mathability gondolkodás helyett [28], mely a fogalomalapú valós problémamegoldást jelenti [29] a gyors és lassú gondolkodás megfelelő arányú és megfelelő helyen történő használatával [32]–[34].

Az alacsony óraszám és a túltervezett tartalom egyik következménye, hogy az informatikaoktatás egy meglehetősen eszközközpontú, unalmas, semmire sem jó tantárgy lett [35]–[38], amely nem támogatja a számítógépes gondolkodás mint alapvető készség [27] fejlesztését. A széles körben elfogadott és népszerű alacsony mathability [28][39] tanítási megközelítések elkövetik mindazokat a hibákat, amelyeket Soloway már 1993-ban felvázolt [40]. A szerző és kutatótársai azt állították, hogy az iskolák nem hatékonyak a programozás tanítása és a tanulók algoritmikus képességének fejlesztése szempontjából, ami az informatika oktatásának egyik végső célja lenne. Megállapították, hogy az iskolák elszigetelték és elválasztották a programozást más tevékenységektől, és az iskoláknak „teljes programozási” megközelítést kellene alkalmazniuk. Emellett bemutatták elképzeléseiket arról, hogyan lehet a programozást integrálni az oktatásba. Megfogalmazták, hogy a programozás mindenütt jelen van, és ezért ki kell terjeszteni a végfelhasználói számítástechnikára, mindenki számára elérhetővé kell tenni és kognitív kontextusba kell ágyazni (real situation) [41]. Azt állították továbbá, hogy egy valós adatokat feldolgozó számítógépes közeg létrehozása megköveteli a helyes kifejezőképességet, a hasznosságot, továbbá lehetővé teszi, hogy a programozás könnyebben tanulható és kivitelezhető legyen.

Akkor, 1993-ban ezek a kijelentések a kutatók kísérletén alapultak, azóta azonban ezek az eredmények többszörösen bebizonyosodtak [42]–[49]. A hangsúlyt a számítógépes gondolkodás fejlesztésére [27][50]–[56] kell helyezni, beleértve az algoritmikus és számítógépes problémamegoldó készségek fejlesztését [10][30][33][57][58], kiterjesztve a végfelhasználói számítástechnikára.

Az a jövőkép, hogy az informatikát mindenkinek tanítsuk, és hogy ennek a tárgynak értelmet adjunk, teljes mértékben összhangban van a Technological Pedagogical Content Knowledge (TPCK) [59][60] és a Meaning System Model (MSM) [45] koncepciójával, amelyek hatékonysága már bizonyításra került. A TPCK azt állítja, hogy a modell mindhárom aspektusának (technológia, pedagógia, tartalom) jelen kell lennie a tanítási-tanulási folyamatban, függetlenül a tantárgytól. Az MSM egy lépéssel tovább megy, és megfogalmazza, hogy a felépített tudáselemek túl a tudományok folyamatos fejlődésébe vetett hit is döntő szerepet játszik a folyamatban. Ezek az eredmények teljes összhangban vannak Hattie [33][61] kutatási eredményeivel, aki különbséget tesz tapasztalt és szakértő tanárok között. Hattie szakértői olyan tanárok, akiknek

magas a tanítási önhatékonyaságuk és erősen hisznek a tudományok fejlődésének inkrementális jellegében.

Az informatika (számítástechnika, digitális kultúra) mint iskolai tantárgy viszonylag korai bevezetése ellenére Magyarország az informatika tanításának-tanulásának alacsony hatékonyságával szembesül. Az informatika tanításának többnek kellene lennie, mint a digitális eszközök értelmetlen használata. Az első figyelmeztetés 2009-ben érkezett a PISA 2009 formájában: Students On Line (1. ábra) [62]. A mérésből kiderült, hogy Magyarországon rendkívül erős negatív korreláció van a számítógép iskolai használata és a számítógépes problémamegoldó képességek között.

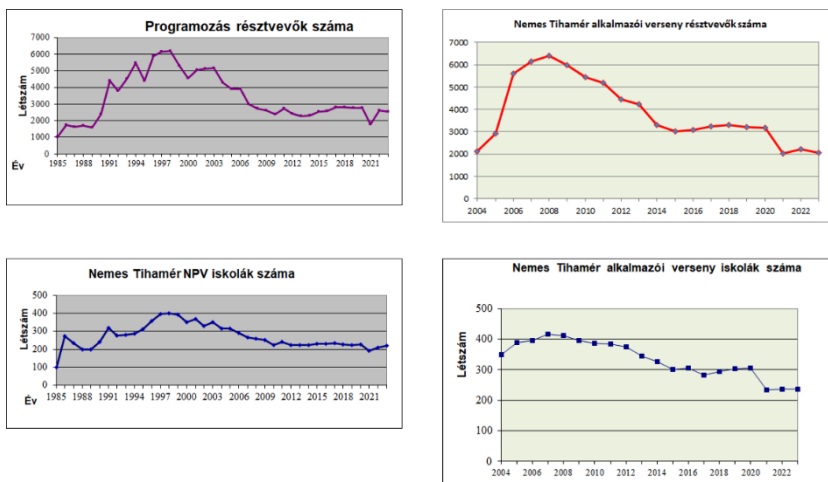
■ Table VI.A ■
AN OVERVIEW OF PERFORMANCE IN DIGITAL READING, NAVIGATION AND COMPUTER USE

	Digital reading performance			Computer use at home			Computer use at school		
	Mean score	Gender difference in digital reading scores between boys and girls	Index of number of relevant pages visited (navigation skills)	Percentage of students who use a computer at home	Percentage difference between top and bottom quarters of the PISA index of economic, social and cultural status	Difference in digital reading scores between those students who use and those who do not use a computer at home	Percentage of students who use a computer at school	Percentage difference between top and bottom quarters of the PISA index of economic, social and cultural status	Difference in digital reading scores between those students who use and those who do not use a computer at school
OECD average	499	-24	46.3	92.3	16.0	80	74.2	0.3	9
OECD									
Korea	568	-18	52.8	87.5	19.5	49	62.7	3.5	2.1
New Zealand	537	-40	49.7	92.5	20.2	90	83.4	6.4	20
Australia	537	-28	49.6	96.7	7.8	84	91.6	5.6	42
Japan	519	-23	50.1	75.9	38.6	48	59.3	2.6	14
Iceland	512	-30	47.5	99.1	1.2	74	79.5	5.1	22
Sweden	510	-26	47.8	97.7	4.7	105	89.1	4.7	28
Ireland	509	-31	47.4	93.2	10.9	60	62.9	0.4	-3
Belgium	507	-24	47.7	96.9	9	102	62.8	-1.1	9
Norway	500	-35	46.9	98.7	2.7	77	93.0	2.5	25
France	494	-20	46.1	m	m	m	m	m	m
Denmark	489	-6	47.2	98.8	2.8	79	93.0	1.8	6
Spain	475	-19	44.7	97.6	14.4	78	65.5	-4.0	11
Hungary	468	-21	41.6	91.8	23.6	102	69.3	-8.9	-27
Poland	464	-29	42.0	92.1	22.9	84	60.6	-9.1	-8
Austria	459	-22	43.3	98.2	3.7	94	84.1	-3.2	-6
Chile	435	-19	37.7	73.2	60.3	69	56.8	-2.0	2
Partners									
Hong Kong-China	515	-8	48.1	96.4	5.2	33	82.6	0.2	3
Macao-China	492	-12	46.5	96.4	5.2	61	80.1	-1.0	4
Colombia	368	-3	31.5	m	m	m	m	m	m

Notes: Values that are statistically significant are indicated in bold (see Annex 3).
Source: OECD, PISA 2009 Database, Tables VI.2.4, VI.3.1, VI.5.1, VI.5.10a, VI.6.2 and VI.6. StatLink <http://dx.doi.org/10.1787/888932436670>

1. ábra: A PISA 2009 eredményei: Students On Line [62].

Egy másik figyelmeztető jel a Nemes Tihámér programozói [63] és alkalmazói [64] versenyeken résztvevő iskolák és diákok számában megfigyelhető tendencia [15]. Ez a verseny már 1985-ben elindult, és a 2018-ban részt vevő iskolák és diákok száma közel megegyezett az 1990-es évi adatokkal. A 2004-ben indult alkalmazói versenyen az első két évtől eltérően 2018-ban vett részt a legkevesebb iskola és diák [65][66] (2. ábra).



2. ábra: A Nemes Tihamér programozói (bal) és alkalmazói (jobb) versenyeken résztvevő tanulók és iskolák száma [65][66].

Hasonlóan alacsony szintű ismereteket mértek, amikor a felsőfokú tanulmányaik megkezdésekor az elsőéves informatikus hallgatókat tesztelték [33][67]. A diákok tanulmányi képességeinek ezen független mérései azt mutatják, hogy a megoldás nem kizárólag az eszközök bevezetésében és fejlesztésében rejlik, beleértve a hardvert és a szoftvert is, hanem inkább az alkalmazott oktatási megközelítésekben és módszerekben [33][40][45][68][69]. Ez a megállapítás teljes összhangban van Soloway 1993-as javaslatával [40]. Az is bebizonyosodott, hogy Prensky elképzelése a digitális bennszülöttekről [70], miszerint a Z-generációnak nincs szüksége informatikai tanulmányokra, mert beleszülettek az eszközök használatába, csak mítosz [71][72], és gyors elterjedése több kárt okoz, mint hasznot. Az is bizonyított, hogy a mobil eszközök intenzív használata inkább negatív hatással van a diákok algoritmikus készségeinek fejlődésére [73].

A magyarországi informatikaoktatás alacsony hatékonyságának egyik további oka a népi pedagógia (folk pedagogy) megközelítés népszerűsége [68]. Az 1980-as évek közepén az egyetemek didaktikailag teljesen felkészületlenül elindították az informatikatanár képzést. Ilyen körülmények között alacsony hatékonyságú tanítási módszereket mutattak be, gyakoroltak és példaként állítottak, akár tudatosan, akár nem. A frissen végzett tanárok diplomával és azzal a teherrel hagyták el az egyetemeket, hogy megtanították őket arra, hogyan viselkedjenek és cselekedjenek úgy, mint a népi tanárok. Ráadásul miután elhagyták az egyetemek oltalmát, ezek a tanárok magukra maradtak. Közülük a legambiciózusabbak saját megközelítések és módszerek kidolgozásába kezdtek, míg a többiek megelégedtek azzal, hogy a diákok nem csinálnak mást,

csak játszanak az órákon [73]. A népi orvoslás és a népi pedagógia összehasonlítása Lister nagy hatású tanulmányában rávilágít ennek a megközelítésnek a negatív és megkérdőjelezhető hatásaira [68], amelyek közé tartozik, hogy a népi pedagógia a „tudományos orvoslástól (pedagógiától)” elkülönülő informális tudás kategóriája, általában íratlan (szájhagyomány útján terjed), nem feltétlenül illeszkedik koherens rendszerbe, és ellentmondásos lehet, néha a kuruzslással hozzák összefüggésbe, ugyanakkor, a múltból származó fontos tudást és kulturális hagyományt is megőrizheti.

Általában a tanárok olyan módszereket alkalmaznak, amelyeket a tanítási-tanulási folyamat valamennyi résztvevője – diákok, tanárok, iskolák, szülők, oktatáspolitikai, ECDL, érettségi stb. – kielégítőnek talál, anélkül, hogy ezen módszerek hatékonyságának bizonyítására támaszkodnának.

2.2. Módszertani eszközök

2.2.1. Táblázatkezelés

Széles körben elfogadott, hogy a táblázatkezelő programok a végfelhasználók számára kifejlesztett legnépszerűbb alkalmazások közé tartoznak; azonban heves vita folyik arról, hogy ezek a programok milyen szerepet játszanak a digitális világban. A hibás táblázatkezelési dokumentumok rendkívül magas száma egyértelműen jelzi, hogy a táblázatkezelés általában véve félreértett, és sokkal fontosabb szerepet játszik, mint azt a legtöbben gondolják [57][74]–[81]. A táblázatkezelés rengeteg tervezést igényel [47], és anélkül, hogy látnánk a (nem csak táblázatkezelőre korlátozódó) programok [17] mögötti koncepciókat, adatokat, algoritmusokat (alapinformatikát), nem beszélhetünk hatékony használatokról. A hibás táblázatkezelési dokumentumok – amelyek elárastják a cégeket, az iskolákat, az internetet és a személyes használatot –, és az ezekből a megvalósításokból származó súlyos anyagi veszteségek egyértelműen jelzik, hogy a táblázatkezelés erejét alábecsülik [57][74]–[81].

Ugyanakkor nem hagyhatjuk és nem is szabad figyelmen kívül hagynunk, hogy a táblázatkezelés is programozás [46]–[49][82]–[93]. A programozás egy egyszerűsített funkcionális nyelven (First-Order Visual Language) [86] „felhasználóbarát” grafikus felületen [94]–[97] történik, amely bevezető nyelvként szolgálna a komolyabb programozási nyelvekhez [48][49], az adatbáziskezeléshez, és ami még fontosabb, bevezető tárgyként az informatika alapfogalmaihoz, felhasználóbarát környezetben [86].

1992 körül megfogalmazták és bizonyításra is került, hogy a funkcionális nyelvek bevezető programozási nyelvként hatékonyabbak, mint az imperatív

nyelvek [42]. Később, amikor a táblázatkezelő programok megjelentek a piacon és váratlan ütemben terjedtek, világossá vált, hogy a végfelhasználók számára még egyszerűbb és megfelelőbb funkcionális programozást lehet kínálni.

Az is bebizonyosodott, hogy a Sprego (Spreadsheet Lego) az általános iskolától a felsőoktatásig és a felnőttképzésig tanítható [48][49][91]–[93][98][99], a megközelítés fő fogalmaira koncentrálva, de a tanulók életkorának, érdeklődésének, háttértudásának függvényében megfelelő módszereket és tartalmakat választva [100]–[104].

A Sprego adaptálja Pólya koncepcióalapú problémamegoldó megközelítést, valamint a kérdezéson alapuló (enquiry-based) tanítási módszerét [29], a 3R mentén a negyedik alapkészségként a számítógépes gondolkodás fejlesztését [27], Kahneman gyors és lassú gondolkodás elméletét [32], a kognitív terhelés elméletét [105] és a TPCCK fő gondolatát [44][60][100]. Mindezen elméletek teljes összhangban vannak Soloway iskolai programozásoktatásra vonatkozó megközelítésével [40] és a táblázatkezelővel való programozás általános elképzelésével [47][84][86][87][89][90]. A Sprego az algoritmusok és sémák felépítésére összpontosít, és a megvalósításban csak 12 függvényt használ [48][49] (3. ábra), a kognitív terhelést a lehető legalacsonyabb szinten tartva.

Szöveg	Szám	Pro
BAL()	SZUM()	HA()
JOBBA()	MIN()	HIBÁS()
HOSSZ()	MAX()	HOL.VAN()
SZÖVEG.KERES()	ÁTLAG()	INDEX()

3. ábra: A Sprego módszer által használt 12 alapfüggvény.

A valós problémák megoldása során a 12 függvényt összetett függvényekké építjük, és tömbképleteket használunk [47][89][92][93]. Ezeket a technikákat biztonságosabb táblázatok létrehozásához és további programozási fogalmak (pl. változó, konstans, ciklus, vektor, mátrix) bevezetéséhez alkalmazzuk.

A különböző algoritmikus és programozási megközelítések eredményesnek és hatékonyak bizonyultak, az áttörés azonban még nem történt meg. Ez minden bizonnyal azzal magyarázható, hogy az informatikaoktatáshoz köthető kutatási eredmények szinte alig érik el az oktatást [106][107]; a probléma azonban sokkal összetettebb. Következésképpen további tanulmányokat és elemzéseket kell végeznünk a táblázatkezelést figyelembe véve.

2.2.2. Szövegkezelés

A természetes nyelvi digitális szövegek kezelése (digitális szövegfeldolgozás, digital text management) egy kevésbé elismert és kutatott téma, digitális műveltségként leminősítve, az informatikából kihagyva [16][17], míg „a szövegszerkesztő használatát” a számítógéphasználat egyik legfontosabb képességeként azonosítják [108].

Feltehetjük a kérdést, hogy mi a különbség a táblázatok és a digitális szövegek között. Az egyik lehetséges magyarázat, hogy a veszteségek, amelyeket a hibás táblázatok okozhatnak, közvetlenül pénznemben kifejezett hatással (veszteséggel) járnak [76]–[79]. Ugyanakkor, a digitális szövegek nem okoznak közvetlen pénzügyi veszteséget. Ez magába foglalja egyrészt, hogy jelen vannak ezek a veszteségek, de a felhasználók – tanulók, tanárok, oktatás egyéb résztvevői, munkavállalók, munkáltatók stb. – figyelmen kívül hagyják ezeket. Másrészt a kutatóknak bonyolult objektív mérőrendszereket kell felállítaniuk ahhoz, hogy mérni tudják azokat a veszteségeket, amelyeket a hanyag digitális szövegkezelés és a digitális szövegek generálnak.

A McKenley & Company felmérésének eredményei (18 000 ember megkérdezése 15 országban) szerint a válaszadók jártassága a digitális kategóriában két képességcsoportban volt a legalacsonyabb:

- „szoftverhasználat és fejlesztés” és
- „digitális rendszerek megértése” [7].

Másrészt a felmérésből az is kiderült, hogy 2023-ban a munkaerőpiacon várható hét alapvető képesség a következő (4. ábra) [8]:

- analitikus gondolkodás,
- kreatív gondolkodás,
- ellenálló képesség, rugalmasság és agilitás,
- motiváció és önismeret,
- kíváncsiság és élethosszig tartó tanulás,
- technológiai műveltség,
- megbízhatóság és figyelem a részletekre.

1. Analytical thinking
2. Creative thinking
3. Resilience, flexibility and agility
4. Motivation and self-awareness
5. Curiosity and lifelong learning
6. Technological literacy
7. Dependability and attention to detail

4. ábra: 2023-ban a munkaerőpiacon várható hét alapvető képesség.

A kiadott felhasználói kézikönyvek, segédletek, tankönyvek, képzési anyagok, tanári kézikönyvek stb. segítséget nyújthatnának a tanulási-tanítási folyamatok hatékonyságának növelésében. Ezeknek az anyagoknak az elemzése azonban azt mutatja, hogy csak nagyon kevés tanulási-tanítási anyag veszi figyelembe az igényt valós problémamegoldásra [41][100][109]–[117]. A legtöbb forrás minősége legalább két okból megkérdőjelezhető. Ezek nyomdai úton előállított anyagok [94]–[98][118]–[134] a szoftvereszközökre és interfészekre összpontosítva és/vagy népi tanárok által készítettek [135]–[139] – a Lister által meghatározott [68] –, akik nem feltétlenül ismerik és nem feltétlenül érdeklődnek a közzétett kutatási eredmények iránt [106][107] és gyakran nincsenek tisztában az önértékelés alacsony megbízhatóságával [74][109].

Továbbá mindenképpen szem előtt kell tartanunk a szövegkezelés, szövegszerkesztés és a szövegszerkesztő használata közötti különbségeket. Nagy általánosságban megfogalmazhatjuk, hogy a szövegkezelés alig ismert. Még ha ismerik is, a három fogalmat szinonimaként használják. Ez azonban nem így van. A három fogalom közötti kapcsolatot a 5. ábra mutatja be.



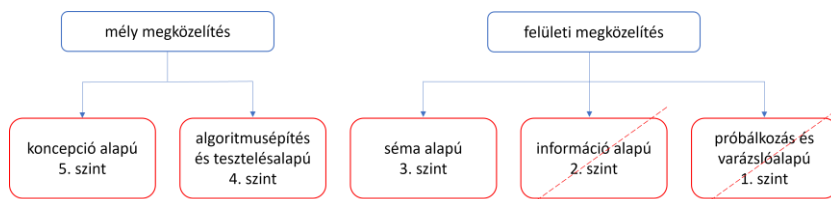
5. ábra: A szövegkezelési tevékenységek közötti kapcsolat szövegszerkesztési környezetben.

Az Encyclopedia of Information Systems szerint a szövegfeldolgozás csak a számítógép használatának művelete az írott, szóbeli vagy rögzített információk gépelt vagy nyomtatott formába történő átalakítására [140]. Kutatásaink azonban bizonyítják, hogy a szövegszerkesztő használata nem elegendő a hibamentes dokumentumok létrehozásához és kezeléséhez.

A szövegkezelés koncepciója elsősorban a problémamegoldásra, a tartalom iteratív folyamaton keresztül történő kezelésére összpontosít. Ugyanakkor, digitális szöveg létrehozásához szövegszerkesztési tevékenységeket is kell végezni. A szövegszerkesztéshez szövegszerkesztőt kell használni. Általánosságban elmondható, hogy a szövegszerkesztő program elsajátítása nem fed le minden olyan ismeretet és készséget, képességet, amely egy helyesen szerkesztett dokumentum hatékony és eredményes létrehozásához szükséges. Ez általában a fogalmi modellekkel szemben barkácsolásnak bizonyul [141][142].

Az Error Recognition Modell (ERM) egy olyan hibafelismerési modell [110], mely természetes nyelvi digitális szövegek hibáinak felismerésére, javítására és kiküszöbölésére alkalmas. Elsősorban szövegszerkesztett szövegekre, prezentációkra és weboldalakra összpontosít, de sok más, változó mennyiségű szöveget tartalmazó dokumentum kezelésére is alkalmas (pl. táblázatok, online kvízek, formatív értékelő eszközök, számítógépes játékok). Az ERM másik jellemzője, hogy egy kérdezősen alapuló tanulási-tanítási megközelítés, ahol mind a tanárok, mind a diákok kérdésekkel irányíthatják a folyamatokat, és elsősorban a tanárok segítik a diákokat a helyes nyomvonalak megtalálásában. A tanárok választják ki a mintaszövegeket, amelyek megfelelnek a tanulók életkorának, háttértudásának, érdeklődési körének (tantárgyi integráció és TPCK) [60].

Cél a tanulók számítógépes gondolkozásának fejlesztése [143]–[146], változatos valódi tartalmak prezentálása a tanulók érdeklődésének fenntartása, a megszerzett tudás változó komplexitású helyzetekben történő alkalmazása, ugyanakkor teret adni az ismétlésnek és a gyakorlásnak, annak érdekében hogy megbízható, alkalmazható tudást tudjunk szerezni (sémákat hívjunk elő, 6. ábra, 3. szint) és sémákat építsünk fel (6. ábra, 4. szint).



6. ábra: A számítógépes problémamegoldó megközelítések tipológiája.

A hibafelismerési modell (ERM) [110] a következő fogalmakon alapul:

- a helyesen formázott szöveg definiálása,
- a digitális szövegek hibáinak kategorizálása,
- nyomtatott és szerkeszthető andonok olvasása és megértése (vizuális eszközök a hibákra való figyelemfelhíváshoz) [147]–[150]
- a poka-yoke alkalmazása (a hiba előfordulásának megelőzése, azaz a hibabiztos rendszer kialakításának módszere) [1]–[5].

A megfelelően szerkesztett szöveg meghatározásának két kritériuma van:

- A szöveg megfelel a nyomtatott dokumentumokkal szemben támasztott követelményeknek (mennyiségi követelmény).
- A szöveg módosításokkal szemben invariáns (minőségi követelmény): a dokumentum szerkeszthető, de csak olyan módosítások engedélyezettek, amelyek megfelelnek a felhasználó eredeti szándékának.



7. ábra: A hibák kategorizálása digitális szövegekben.

A definíció két kritériumának megfelelően a digitális szövegek hibái két hipernim és ezeken belül három-három hiponim kategóriába sorolhatók (7. ábra). A kvantitatív hibák nyomtatott formában felismerhetőek, a dokumentum olvasója vagy nézője számára láthatóak:

- szintaktikai (a szöveg nyelvtana);
- szemantikai (a szöveg tartalma);
- tipográfiai (a szöveg megjelenése) [151]–[154].

A kvalitatív hibák csak szerkeszthető digitális formában ismerhetők fel; ezért a célközönség számára láthatatlanok, és kizárólag a szerző(k) és az eredeti vagy módosított fájlt kezelő többi résztvevő látja őket:

- tördelési (a szöveg karaktereinek és objektumainak elrendezése);
- formázási (formázási parancsok alkalmazása);
- stílus (stílusok kezelése és alkalmazása, definálása, formázása).

Nem ritka, hogy az egyik hibakategória kivált egy másikat vagy egy harmadikat. Ezekben az esetekben elsődleges és másodlagos kategóriákról beszélünk, attól függően, hogy melyik váltja ki a másikat.

Az ERM tanulási-tanítási környezetben a digitális szövegek kezelése a hibafelismerési folyamattal indul, ahol a dokumentumok hibáit fel kell ismerni, jelölni és kategorizálni kell. A folyamatnak két jól elkülöníthető fázisa van:

- hibafelismerés a nyomtatott szövegben, beleértve a különböző megjelenítési formákat (pl. papír, monitor),
- hibafelismerés a szerkeszthető formátumban, ha rendelkezésre állnak a szoftver által kínált andonok (pl. nem-nyomtatódó karakterek, rácsvonalak, horgonyok megjelenítése).

A hibák felismerése és megértése után a célállapot a hibák kiküszöbölése, egy hibamentes dokumentum létrehozása, ami a poka-yoke alkalmazása. Ebben a fázisban a hibák jellegének megfelelően elsősorban törlést és a hibás formázás eltávolítását végezzük el. Tulajdonképpen ez az első fázis, amikor egy szoftveres eszközt használunk, azonban csak egy nagyon korlátozott készletet alkalmazva.

Miután létrehoztunk egy hibamentes dokumentumot, következik a dokumentum formázása. Ez az a fázis, amikor szigorúan a szoftver azon eszközeit választjuk ki és alkalmazzuk, amelyeket a tartalom megkíván, mivel a tervezett formátumnak a tartalommal teljes összhangban kell lennie.

Az ERM lépéseiből egyértelmű, hogy a hangsúly a szövegre, a tartalomra helyeződik [110]. A modell egyrészt támogatja az olvasókat azáltal, hogy a szöveget a lehető legolvashatóbbá és legérthetőbbé teszi. Másrészt segít a kvantitatív hibák elkerülésében és kijavításában, valamint abban, hogy a szöveg készen álljon a csak a szerző/szerkesztő szándékainak és elképzeléseinek megfelelő hatékony módosításra és formázásra. A szövegszerkesztő alkalmazás és a GUI használata csak másodlagos szerepet tölt be, az ERM szövegkezelési folyamatában való jártasságra építve.

A hagyományos tankönyvekkel, tanítási-tanulási anyagokkal és segédletekkel [97][119]–[122][126]–[134][155][156] ellentétben, ez a koncepció nem az eszközökre fókuszál. A legtöbb segédanyag az alábbi tartalmakat foglalja magába:

- operációs rendszer parancsai (ezt a tudást tudástranszfer útján kellene behozni a szövegszerkesztésbe, és ha nem, akkor máshol kellene bevezetni és gyakorolni) [97][121][122];
- a betűformázásokkal való játszadozás [126]–[134];
- az alkalmazás új funkciói [122];
- a GUI-objektumok bevezetése [122];
- szöveg begépelése.

Ezek helyett az ERM-ben a tanítási-tanulási folyamat során valós tartalmú, valós életből vett szövegeket használunk, amely teljes összhangban van az alábbi hatékony rendszerek fő koncepciójával:

- Informatikai referenciakeret [17][157];
- TPCK [44][59][60];
- Pólya koncepció-alapú problémamegoldó megközelítése [29];
- Kahneman gyors és lassú gondolkodás elmélete [32];
- Sweller kognitív terhelés elmélete [105];
- Hattie eredményei a hatékony tanításról [61];
- Chen által felépített Meaning System Model [45].

Ezen elméletek kombinációját alkalmazva, a tanítási-tanulási folyamatban felhasznált dokumentumokat a tanulók (általában a végfelhasználók) életkorához, háttértudásához, érdeklődési köréhez, iskolai tantárgyaihoz, munkaköve-telményeihez stb. lehet igazítani.

Mind a digitális szövegek javítása, mind a formázás húzó (pull) rendszerben történik, ami azt jelenti, hogy nem a szoftverfelületet tanítjuk meg teljes részletességgel, hanem csak azokat az eszközöket, amelyek szükségesek a célállapot hatékony és eredményes eléréséhez (a kiválasztott feladat megoldásához).

Az ERM reaktív válasz a hibásan kezelt digitális szövegek kezelésére. Egy hibásan szerkesztett dokumentumra alkalmazott ERM-folyamat egy cél állapot a pazarlás csökkentésére egy már létező dokumentum módosítási folyamatában. A távoli cél azonban az, hogy proaktív válaszokat találjunk a keletkezett (új) problémák kezelésére. Ennek a célnak a teljesítéséhez meg kell tanulnunk a helyes természetes nyelvi digitális szövegek elkészítésének alapjait. Ehhez a problémamegoldás koncepcióalapú megközelítését kell alkalmaznunk, amely az oktatásban és az iparban is eredményesnek és hatékonyan bizonyult [6][29][158][159].

Ezek a megközelítések elsősorban nem lineáris, hanem iteratív folyamatok, amelyek az adott helyzetben megfelelően alkalmazott gyors vagy lassú gondolkodást igényelnek (2. táblázat), és a számítógépes problémamegoldás tipológiájában helyesen a felső három szintre helyezhetők (az 1. és 2. szintek nem valódi problémamegoldó, hanem felületi megközelítések) (6. ábra).

Cél	Gondolkodás mód	Számítógépes problémamegoldás
probléma megértése	lassú	5. szint
tervezési megoldás	lassú	5. szint
sémák kialakítása	lassú	4. szint
sémák hívása	gyors	3. szint
tesztelés	gyors és lassú	minden szint

2. táblázat: A koncepcióalapú problémamegoldó megközelítés lépései, az ezekhez a lépésekhez kapcsolódó gondolkodási mód, valamint a számítógépes problémamegoldás tipológiájában elfoglalt helyük.

A gyakorlatban szükségünk van egy jól definiált poka-yoke-ra [1]–[5], ami olyan rendszer, amely a hiányosságokból, tudatlanságból és/vagy túlzott magabiztosságból eredő hiba vagy tévedés (poka) megelőzését vagy kiküszöbölését (yoke) szolgálja. A poka-yoke célja a hiba megelőzése, azaz egy hibabiztos rendszer kialakításának módszere. Ezeket a szempontokat figyelembe véve az ERM hatékony és eredményes poka-yoke rendszernek bizonyult a digitális szövegkezelésben. Továbbá, ismereteink szerint az ERM az egyetlen olyan rendszer, amelyet végfelhasználók számára építettek, annak érdekében, hogy a természetes nyelvi digitális szövegeket komplexitásukban hatékonyan és eredményesen tudjuk kezelni.

2.3. Digitális fenntarthatóság

Korábbi kutatások kimutatták, hogy rendkívül nagyszámú hibás szöveges dokumentum van forgalomban, ezek újratermelődnek és mintaként szolgálnak mind az irodai világban, mind az oktatásban [33][141][142][110][160]–[162]. Ezeknek a dokumentumoknak a létrehozása, szerkesztése és módosítása komoly pénzügyi veszteségeket okoz, ha mind az emberi, mind a gépi erőforrásokat figyelembe vesszük.

Amikor dokumentumok digitális fenntarthatóságáról beszélünk, akkor a digitális fenntarthatóság gyűjtőfogalmát használjuk [163], amelynek célja a digitalizáció maximális kihasználása a fenntartható fejlődés érdekében. Ez azt jelenti, hogy a tudást olyan erőforrásnak kell tekinteni, amelynek önmagában is fenntarthatónak kell lennie, hogy megőrizze értékét a társadalom számára, és biztosítsa, hogy tartósan hozzájárulhasson a fenntartható fejlődés céljaihoz.

Általánosságban azonban, amikor a digitális fenntarthatóságról van szó, elsősorban a gyorsabb, könnyebb, kisebb, felhasználóbarátabb, kifinomultabb hardver vagy szoftver építésére szokás gondolni [164][165]. Másrészt az elektromos és elektronikus berendezések és az e-hulladék újrahasznosítását tekintik digitális fenntarthatóságnak [166][167].

Bár ezek a legnyilvánvalóbb szempontok, és természetesen mindegyik szükséges az energiahatékony digitális fenntarthatóság céljának eléréséhez, arra is rá kell mutatnunk, hogy ez nem elég. A fenntarthatóság kérdését újszerű nézőpontból kell megközelíteni, figyelembe véve, hogy kiknek készülnek ezek az eszközök, kik használják ezeket az eszközöket, és kik felelősek a „jó minőségű” adatok létrehozásáért: az emberek [163][168][169]. Az emberi erőforrások szerepe a fenntarthatóságban legalább olyan fontos, mint a hardver és a szoftver, valamint ezek kezelése. Tucker szerint [169], a digitális fenntarthatóság nem feltétlenül a digitális eszközökről szól. „Lehet valami sokkal nagyobb, de kevésbé kézzelfogható dolog is: a fenntartható kutatás mint olyan kritikai perspektívákhoz és gyakorlatokhoz való hozzájárulás, amelyek gazdagítják a humán tudományok minden jövőbeli munkáját”.

Elsődleges törekvésünk – a digitális fenntarthatóság és a digitális humán tudományok terén mutatkozó hiányt pótolva – elemezni:

- az irodai adatkezelés (office computing), az irodai alkalmazottak és munkáltatóik hatékonyságát,
- az emberi erőforrások pazarlásának forrását és következményét,
- az ezekhez a problémákhoz kapcsolódó oktatási kérdéseket, továbbá
- az elemzések révén kritikus szemléletet és gyakorlatot nyújtani.

A korábbi tanulmányok elemzéseit feltárták, hogy az irodai adatkezelés alacsony hatékonyságát szinte soha nem ismerik fel, nem tárják fel annak természetét és az okozott negatív pénzügyi hatásokat [33][141][110][142][160]–[162][170]. Az egyetlen kivétel a táblázatkezelés, ahol a hibák azonnal érzékelhetően komoly pénzügyi veszteséget okozhatnak [76][79][171]. A digitális természetes nyelvi szövegekben a hibák ravaszabbak, és következésképpen nehezebb felismerni, kezelni és kijavítani őket.

Kutatócsoportunk egy olyan megközelítést dolgozott ki, amely alkalmas a hanyag és helytelen dokumentumkezelés és az adatkezelési eljárások mögött meghúzódó problémák feltárására és az irodai számítástechnikai veszteségek mérésére [6][11][28][39][172], támogatva a hatékony adatkezelést elsősegítő számítástechnikai környezet létrehozását.

Kutatásunk pozicionálásához és a digitális fenntarthatóságban betöltött szerepének meghatározásához a digitális eszközökre, az őket körülvevő ökoszisztémákra és a fenntartható fejlődéshez való hozzájárulásukra vonatkozó tíz konkrét alapfeltételt alkalmazunk [163]. A tíz alapfeltételt a természetes nyelvi digitális szövegekre alkalmazva bizonyítékot találhatunk arra, hogy azok általában negatívak és károsak a fenntartható fejlődésre nézve (3. táblázat).

Feltétel	Természetes nyelvű digitális szövegek
kidolgozottság	Több millió dokumentum van. E dokumentumok többsége hibás, hiányos vagy rossz minőségű.
átlátható struktúrák	A dokumentumok szerkeszthető formátumban, andonnal vagy anélkül, és legtöbbször csak olvasható formátumban (PDF, papír stb.) állnak rendelkezésre. Álláspontunk szerint pl. az önéletrajzokat andonnal ellátott szerkeszthető formátumban kellene kérni.
szemantikus adatok	Általában nem állnak rendelkezésre szemantikus adatok, annak ellenére, hogy a programok stílusokat, elrendezéseket és további eszközöket kínálnak a tartalom megfelelő strukturálásához. Ha ezeket az eszközöket megfelelően alkalmazzák, akkor címkézőként (tag) szolgálnának.
elérhetőség, tárolás	Több millió digitális szöveges dokumentum érhető el az interneten, de sokkal több van magángyűjteményekben.
szabad felhasználás	Még a feltöltött digitális dokumentumok sem feltétlenül állnak rendelkezésre további publikációkhoz és kutatásokhoz.
megosztott hallgatólagos tudás	Olyan állapotban vagyunk, ahol a megosztott hallgatólagos tudás többet árt, mint használ, mivel elsősorban tévhitek és toló oktatási rendszerek terjedtek el.
részvételi kultúra	A digitális szövegek közösség szerzői és módosítási kultúrája nem ideális, mivel a szerkesztők közül sokan nem rendelkeznek alapvető ismeretekkel, de túlságosan magabiztosak. Másrészt a különböző magas színvonalú számítástechnika-oktatási elméletek nem kerülnek be az oktatási rendszerekbe.
jó példák, irányítás	A végfelhasználói számítástechnikát és a természetes nyelvi digitális szövegek kezelését újra kellene definiálni, és a fogalomalapú tanulási-tanítási módszereket és azok alkalmazását be kellene vezetni és el kellene terjeszteni.
diverzifikált finanszírozás	Finanszírozásra és támogatásra van szükség ahhoz, hogy a koncepcióalapú módszerek elérjék az oktatást. Elsőként azonban fejleszteni kellene a tanárok számítógépes gondolkodásának szintjét.
hozzájárulás a fenntartható fejlődéshez	A digitális dokumentumoknak közvetlen és közvetett hatásai vannak a fenntartható fejlődés mindhárom szintjén, melyek az életciklus, az üzemeltetés és a strukturáltság. E szempont figyelembevételével azt is bizonyítja, hogy a végfelhasználói számítástechnika döntő szerepet játszik a digitális fenntarthatóságban.

3. táblázat: A fenntartható digitális objektumok tíz alapfeltétele természetes nyelvi digitális szövegekre alkalmazva [163].

3. Mini-kompetenciateszt

2018 májusában egy mini-kompetenciatesztet állítottunk össze a tanulói tudástranszferelemek azonosítására, a tanulók számítógépes gondolkodásának és algoritmikus készségének mérésére hagyományos és nem-hagyományos programozási környezetekben. A mini-kompetenciateszt különlegessége, hogy a feladatokat és a kérdéseket rövid leírásokkal, ábrákkal egészítettük ki. Ezekkel a rövid és vizualizált feladatokkal elsődleges célunk, hogy csökkentjük az olvasás okozta kognitív terhelést, mivel a nagy igényű szövegek torzítanák a tesztek eredményét.

A korábbi PISA-tesztek (2015, 2018) [62][173] szerint a magyar tanulók eredményei mindhárom tantárgyból – olvasás, matematika és természettudományok – az OECD-átlag alatt vannak, és mindhárom tantárgyban az átlagnál jelentősen magasabb az alacsonyan teljesítők aránya. A legutóbbi PISA-teszt (2022) eredményei [174] nem térnek el szignifikánsan az OECD-átlagtól. A PISA 2009: Students On Line felmérés szerint negatív korreláció mutatkozik az eredmények és a számítógép iskolai használata között. Ezeket a publikált eredményeket figyelembe véve szeretnénk volna csökkenteni az olvasás kognitív terhelését, valamint az eszközközpontú tanítási megközelítések negatív hatását az informatika tantárgyban [31]. Céljaink elérése érdekében a feladatok összetettségét csökkentettük, a feladatok szövegét a PISA-feladatokhoz képest lerövidítettük és minden feladatot informatív képekkel támogattunk meg.

3.1. A teszt felépítése

A mini-kompetenciateszt [175] három fő részre osztható: általános kérdések; táblázatkezelő függvények ismerete és mini problémák megoldása.

A teszt első szakaszában (1. függelék és 2. függelék, G1–G11) általános információkra vonatkozó kérdéseket tettünk fel, továbbá itt kaptak helyet azok a kérdések is, amelyek arra vonatkoznak, hogy a tanulók hogyan értékelik tudásukat az egyes témakörökben (2. függelék, G11). Ezekhez egy 0–5 Likert-skála állt a kitöltők rendelkezésére. Az önértékeléssel kapcsolatban a diákokat arról is megkérdeztük, hogy tanulják/tanulták-e ezeket a témaköröket az iskolában vagy sem. Mivel az önértékelést és a feladatokban elért eredményeket jelentősen befolyásolhatják az informatikaórán végzett tevékenységek, ezért a diákokat arra is kértük, hogy az informatikaórákat jellemző tevékenységek előre meghatározott listájából válasszák ki (1. függelék, G7). Az általános rész a tanulók számítógép- és mobiltelefon használatára vonatkozó kérdéseket is tartalmaz mind az otthoni, mind az iskolai használatra vonatkozóan (1. függelék, G2–G5). Az ezen eszközök használatára fordított időt egy hételemű listá-

ból való kiválasztással kérdeztük meg. Négy kérdést állítottunk össze, figyelembe véve a számítógépen (asztali számítógép és laptop) és mobil eszközökön (táblagép, telefon és e-könyv olvasó) az informatikától eltérő órákon egy héten belül eltöltött időt és a napi személyes használatra eltöltött időt. Kérdést tettünk fel továbbá az 1–12. évfolyamon a heti informatikaórák számával (1. függelék, G6), az iskolában tanult programozási nyelv(ek)kel (2. függelék, G8), az általuk használt tankönyvekkel (2. függelék, G9), valamint ezek használatának gyakoriságával (2. függelék, G10) kapcsolatban is.

A teszt második részében (2. függelék, G12) a diákokat arra kértük, hogy sorolják fel az általuk legfontosabbnak tartott tizenöt táblázatkezelő függvényt. A függvények nevét rangsorolás nélkül kellett megadni. A feladat egyik célja, hogy kiderüljön, a diákok emlékeznek-e az órán, a tankönyvben és a táblázatkezelő alkalmazásokban bemutatott függvényekre [31]. A tizenöt legfontosabb függvényt felsoroló feladat további céljai közé tartozik annak megállapítása, hogy az iskolák megtanítták-e a függvényeket a tanulóknak, és a tanulók ténylegesen tudják-e a függvényeket, vagy csak navigálnak a felhasználói felületen.

A teszt harmadik részében (3. függelék–6. függelék) gyakorlati feladatokkal igyekeztünk mérni a tanulók számítógépes gondolkodását, problémamegoldó képességét és a tudástranszferelemek aktiválását. A 2012-es kerettantervek elemzése során [9] azt tapasztaltuk, hogy a megfogalmazott tudástartalmak közel azonosak az egyes tematikai egységekben. Mivel minden évfolyamon (7–10. évfolyam) ugyanazt a tesztet töltötték ki a tanulók, így vizsgálatainkkal arra is rá tudunk mutatni, hogy a hasonló tudástartalmú témaköröket tekintve van-e szignifikáns különbség az egyes évfolyamok eredményei között.

A feladatok a perifériák, a fájlkezelés, a szövegkezelés, a táblázatkezelés, az algoritmizálás és programozás, valamint a forráskezelés témakörökből kerültek összeállításra, melyek mindegyike saját összeállítású vagy a korábbi évek méréseiből átvett feladatok, és a valós életből vett problémákra kérdeznek rá számítógépes környezetben.

A kérdések típusát tekintve a teszt 70%-a feleltválasztós és 30%-a rövid választ igénylő feladat, ezzel elősegítve azt, hogy a teszt feladatai 45 perc alatt, azaz egy tanóra alatt kitölthetőek, megoldhatóak legyenek. A tesztben elérhető maximális pontszám 66 pont. A következőkben azoknak a feladatoknak a részletes bemutatására kerül sor, melyeket a dolgozatomban bemutatott eredmények során felhasználtam.

3.2. Táblázatkezelés

A tanulók táblázatkezelői ismereteit a képletkiegészítés (6. függelék, F14), a műveleti sorrend (6. függelék, F15), az adattípusok (6. függelék, F17 1–4. kérdés) és a tömbképlet (6. függelék, F17, 5. kérdés) feladatokkal mértük.

3.2.1. Képletkiegészítés

A kitöltőnek a képletkiegészítés feladatban (8. ábra, felül a hiányos, alul a kitöltött feladat) szintaktikailag helyes képletekké kellett az ábrákat kiegészíteni, majd megadni azt a tartományt, amelyre hivatkozik.

	A
1	SZUM(A2 A20

	A
1	SZUM(A2 A20

	A
1	=SZUM(A2:A20)

	A
1	=SZUM(A2;A20)

A2-től A20-ig

A2 és A20

8. ábra: A képletkiegészítés feladat (felül) és megoldása (alul) magyar nyelvi táblázatkezelői beállításokkal.

Tudástranszfert mérő tudáselemek:

- informatika: táblázatkezelőben minden képletet „=” jellel kezdünk,
- matematika és informatika: függvény fogalma, az argumentumok ke-
rek zárójelek között szerepelnek,
- informatika: összefüggő és nem összefüggő tartomány,
- tartományok megnevezésekor a „-tól” és „-ig”, valamint az „és” sze-
repe.

A pontozáskor minden helyesen beírt elemet 1-1 ponttal értékeltünk, így összesen 8 pontot szerezhettek a tanulók erre a feladatra:

- két egyenlőségjel
- két berekasztó zárójel
- két tartományírásjel
- két tartományhivatkozás.

3.2.2. Műveleti sorrend

A méréskor érvényben lévő általános iskolai informatika kerettanterv alapján [24] a tanulók képesek cellahivatkozások használatára, képletek szerkesztésére táblázatkezelői környezetben, továbbá képesek egy feladat megoldásainak lépését szöveges, rajzos formában, értelmezni, prezentálni. Ez a tudás tovább bővül középiskolában, ugyanis a méréskor érvényes informatika kerettantervek alapján [25] ekkorra a tanuló már képes a különböző tantárgyak problémáit informatikai eszközök segítségével megoldani, ezekhez a megoldásokhoz algoritmust készíteni. Ezeket figyelembe véve készítettünk a tesztbe egy olyan feladatot, mely azt hivatott mérni, hogy a tanulók meg tudják-e határozni helyesen egy táblázatkezelői környezetben írt képlet műveleti sorrendjét (6. függelék, F15).

A tanulók feladata tehát az volt, hogy meghatározzák, hogy milyen sorrendben hajtja végre a megadott képletben (9. ábra) a műveleteket a táblázatkezelő. A feladat megoldását a 4. táblázat mutatja (rövid válaszok a zárójelben megadva).

$$=HA(\text{ÁTLAG}(D2:D58)-50<A5;"a";"")$$

9. ábra: A műveleti sorrend feladatban megadott képlet.

1. lépés	Átlag kiszámítása (átlag).
2. lépés	A kapott eredményből kivonunk 50-et (kivonás).
3. lépés	Eldöntendő kérdés: kisebb-e mint 50? (eldöntendő kérdés)
4. lépés	HA() függvény: ha igen, akkor kiíratunk egy „a” betűt, ha nem, akkor kiíratjuk az üres sztringet (HA() függvény hívása).

4. táblázat: A műveleti sorrend feladat megoldása.

A feladat tudástranzfert mérő tudáselemei a következők:

- matematika: műveletek végrehajtási sorrendje,
- matematika: átlag,
- informatika: HA() és ÁTLAG() függvények,
- informatika: D2:D58 (nem osztás, hanem tartománykijelölés).

Általános iskolában a tanulók megtanulják matematika órán a négy alpművelet és a zárójel használatakor fennálló helyes műveleti sorrendet, mely által a műveleti sorrend tantárgyközi tudástranzferelemként jelenik meg a teszt ezen feladatában. Ezen túlmenően megismerkednek az átlag definíciójával is.

Az elsőfokú függvények fogalma és gyakorlata a felsőtagozaton kerül bevezetésre, a függvények értelmezési tartományával és értékészletével együtt. Ezen túlmenően, az informatika tantárgy keretein belül kezdődik meg a táblázatkezelői ismeretek tanítása. Ezen előismeretek alapján feltehető, hogy a tanulók a felismerik az ÁTLAG() függvény argumentumát, a D2:D58 tartományt. Az informatika tantárgyon belüli tudástranszferelemként jelenik meg a HA() és az ÁTLAG() függvény, illetve a már említett tartománykijelölés is, mint az ÁTLAG() függvény argumentuma.

Ebben a feladatban megértés SOLO kategóriáit alkalmaztuk a megoldások értékelésekor [176]–[178], mely segítségével könnyen meghatározhatjuk, hogy a tanulók milyen mélységben sajátították el a tudáselemeket. A megértés SOLO kategóriáit a klasszikus programozási ismeretek mérésére, a programozási feladatok megértési szintjeinek azonosítására hozták létre és a tanulói teljesítmények alapján öt szintet határoztak meg (5. táblázat, 1. oszlop). Korábbi kutatások egyértelművé tették, hogy a SOLO kategóriák alkalmazhatóak táblázatkezelői környezetben is [67] (5. táblázat, 2. oszlop), figyelembe véve, hogy a koncepció alapú táblázatkezelői megközelítések első-rendű funkcionális programozási megoldások [47]–[49]. Fontos megjegyezni, hogy a teszt feladatival az Extended abstract szint mérésére nem volt lehetőség.

Eredeti SOLO kategóriák	Módosított SOLO kategóriák	Jelentés
	Ignored (I)	A tanuló nem foglalkozik a feladattal.
Prestructural (P)	Prestructural (P)	A válasz és a kérdés, valamint a válasz és a helyes válasz között nincs kapcsolat.
Unistructural (U)	Unistructural (U)	A válaszban több tudáselemből egyet vesz észre és jelöl meg helyesen, a többi pontatlan.
Multistructural (M)	Multistructural (M)	A válaszban több tudáselemet is megnevez, de ezeket nem tudja egységként kezelni és a helyes választ teljes egészében megfogalmazni.
Relational (R)	Relational (R)	Minden tudáselemet felismer, össze tudja ezeket kapcsolni és a választ is meg tudja fogalmazni.
Extended abstract (EA)	Extended abstract (EA)	A tanuló képes általánosítani a feladatot.

5. táblázat: Az eredeti SOLO kategóriáik programozási feladatokhoz, a módosított SOLO kategóriák táblázatkezelési feladatokhoz és a kategóriák jelentése.

Figyelembe véve az előzetes kutatási eredményeket, a teszt ezen feladatához a következő SOLO kategóriákat fogalmazzuk meg.

- Prestructural kategóriába soroltuk azokat a tanulókat, akik 0, vagy 1 pontot szereztek:
 - hozzá sem kezdtek a feladathoz, vagy
 - értékelhetetlen hibás megoldást adtak, vagy
 - egyetlen egy helyen találták el a megoldást, a szövegkörnyezetből teljesen kiszakítva.
- Unistructural kategóriába kerültek azok a diákok, akik 2 pontot szereztek, azonban nem úgy, hogy az első és az utolsó lépést határozták meg helyesen:
 - két helyen találták el a megoldást, de ebben az esetben is a szövegkörnyezetből teljesen kiszakítva, véletlenszerűen.
- Multistructural kategóriába azok kerültek, akik
 - 2 pontot szereztek úgy, hogy az első és az utolsó lépést határozták meg, vagy
 - 3 vagy 4 pontot kaptak a megoldásra.

Mindkét esetben felismerhető, hogy a tanulók nem véletlenszerűen számozták be a lépéseket, hanem felismerték a sorrend legfontosabb elemeit, de még nem voltak képesek a teljes szerkezet hibátlan azonosítására: felismerték, hogy a műveletsorban melyik az első és utolsó lépés, vagy azonosítani tudtak legalább három lépést a megfelelő sorrendben.

- Relational kategóriába az 5 vagy 6 pontot érő megoldások kerültek. Azok a tanulók szereztek ennyi pontot akik hibátlanul meg tudták oldani a feladatot vagy felismerték a lépések közötti kapcsolatot, de egy apró hibát ejtettek (pl. két lépésnek jelölték a HA() függvény IGAZ és HAMIS ágának kiértékelését).

3.2.3. Adattípusok, adatelemzés

Az adattípusok, adatelemzés feladatban a probléma gyökere a különböző nyelvekben eltérő szemantikájú vesszőben rejlik. A táblázatot a „Top 250 YouTube Channels in Hungary” című weboldalról [180] töltöttük le (10. ábra), ahol a nyelv kiválasztása után a tartalom igazodik a választott nyelvhez.

Rank	Grade	Username	Uploads	Subs	Video Views
1st	A	LetsGoMartin - Nursery Rhymes	194	2,050,447	493,359,935
2nd	B+	REAL TRILL MUSIC	372	328,644	263,907,715
3rd	B+	KerekMese	143	353,603	662,791,261
4th	B	#MISSHMUSIC	124	340,360	201,745,398

10. ábra A YouTubers csatorna magyar változata a vesszővel, mint ezres elválasztó karakterrel.

Hiába választjuk ki azonban a számunkra megfelelő nyelvet, a weboldal a nyelv szintaktikáját figyelmen kívül hagyja, így az angol ezres elválasztó karaktert, azaz a vesszőt alkalmazza elválasztó karakterként függetlenül a kiválasztott nyelvtől. A fájl magyar (és a legtöbb európai) táblázatkezelőben történő megnyitásokor a táblázatkezelő a vesszőt decimális karakterként értelmezi, és ha a számban egy vessző szerepel, akkor az egész számot valós számmá konvertálja (11. ábra, B6 és B7, C3–C251 cellák), ha két vessző is szerepel a számban, akkor azt sztringgé konvertálja a táblázatkezelő (11. ábra, C2 és D2–D251 cellák).

	A	B	C	D
1	Felhasználó	Feltöltés	Feliratkozó	Megtekintés
2	VamosART	484	1,107,555	226,195,766
3	Videómánia	338	833,23	254,545,702
4	PamKutya	120	809,866	223,441,355
5	LetsGoMartin	176	725,638	162,798,559
6	TheVR	1,062	592,675	213,550,948
7	luckeY	1,183	561,13	150,341,428
8	Peter Gergely	100	548,241	79,713,757
9	Scribble Netty	159	546,049	74,234,471
248	Szilvaglam	87	61,899	3,918,538
249	rance flow	524	61,863	53,275,385
250	KIS GRÓFO (official)	9	61,65	30,712,031
251	KODIAK	736	61,467	14,599,194

11. ábra: Az adattípusok, adatelemzés feladatban szereplő táblázat.

Az adattípusok feladatban a tanulóknak meg kellett határozni, hogy

- milyen típusú adatok szerepelnek az egyes oszlopokban,
- hány adatsort (rekordot) tartalmaz a táblázat,
- a B oszlopban mi a látható legnagyobb szám,
- a B oszlopban mi a látható legnagyobb szám.

A feladat a táblázatok adattípusainak felismerésére, az ezres elválasztó és decimális karakter helyes/helytelen használatára, valamint a különböző adattípusokra vonatkozó ismeretek adatbáziskezelés és programozás témakörökbe való átemelésére is irányul. A tudástranszfer egy másik aspektusa is kiemelt szerepet játszik: a számok szintaktikája. A számok írásának szabályait minden nyelv esetén a helyesírási szabályzatok fogalmazzák meg. Magyar nyelvű szövegek esetén az Akadémiai Kiadó A magyar helyesírás szabályai, 12. kiadás című kiadványát használjuk, ami 2015. szeptember 1-jétől érvényes [181]. A számok írására vonatkozó tudáselemek tehát az anyanyelvi és az idegen nyelvi órákról érkeznek. Ebben a tudástranszfer-láncban a táblázatkezelés átmenetként funkcionál a természetes és a mesterséges számítógépi nyelvek között.

Tudástranszfert mérő tudáselemek:

- a vessző szemantikája,
- programozás: adattípus,
- adatbáziskezelés: adattípus,
- matematika: egész szám, valós szám, tizedes karakter, ezres elválasztó karakter,
- automatikus típus felismerés és az igazítás kapcsolata.

A feladatban szereplő táblázat (11. ábra) oszlopaiban az adatok típusát a 6. táblázat részletezi. Az adattípusok feladatra 6 pontot lehetett szerezni.

Oszlop	Adattípusok
A oszlop	szöveg
B oszlop	egész szám: (B2:B5) és (B8:B251) valós szám: (B6:B7)
C oszlop	szöveg: C2 valós szám: (C3:C251)
D oszlop	szöveg

6. táblázat: Az adattípusok feladatban szereplő táblázat oszlopaiban szereplő adatok típusa.

A táblában rejtett sorok vannak, így összesen 251 sor található benne, de az első sor a mezőnevekre van fenntartva, így a táblázat 250 adatrekordot tartalmaz. A mintán a B oszlopban szereplő legnagyobb szám 736, míg a legkisebb szám 1,062. Az adatelemzés feladatra 3 pontot szerezhettek a tanulók.

Ha a tanulók ismerik a különböző adattípusokat a táblázatkezelésben, akkor ezt a tudást később más környezetben is alkalmazhatják, például az adatbáziskezelésben és/vagy a programozásban. Ennek az állításnak a megfordítása is igaz. Általában az adatkezelést igénylő egyéb interfészek tanítása előtt táblázatkezelést érdemes tanítani a tanulóknak, mivel a táblázatkezelő felületek

könnyebben kezelhetők és a funkcionális nyelvek hatékonyabbnak bizonyultak első szövegalapú programozási nyelvként, mint az adatbáziskezelés vagy a programozás [42]. Fontos megemlíteni, hogy a MS Excel az adatok igazítását a felismert adattípus szerint állítja be: a cellában balra igazítja a karakterláncokat, jobbra a számokat, míg a logikai és hiba értékeket középre, így az adattípusok minden cellában egyértelműen leolvashatóak. Ezzel a vizualizációval még a nem szándékosan különböző típusú területek is könnyebben felfedhetők. Azonban figyelembe kell vennünk a táblázatokban az automatikus típusfelismerést, mivel a táblázatkezelő nem minden adatot ismer fel eredeti szándékunknak és a szemantikai tartalomnak megfelelően (például a teszt F17 feladatában), és meg kell tanulnunk a vizualizáció értelmezését, olvasását.

A grafikus felületek megjelenése kétségtelenül leegyszerűsítette a számítógépek használatát azok számára, akik nem rendelkeznek fejlett informatikai ismeretekkel. Ezeknek a grafikus „üzeneteknek” az értelmezése azonban olyan kompetenciákat igényel, amelyek tanításának hatékonysága jól szervezett és tervezett oktatási környezetben jelentősen növelhető. A teszt eredményei megmutathatják, hogy a Z-generáció [70] diákjai ezzel a tudással születnek-e vagy sem, és hogyan képesek ezek a diákok a nem szöveges adatok olvasására és a vizuális reprezentáción alapuló információk elérésére.

3.2.4. Tömbképlet

A tömbképlet feladatban (6. függelék, F17, 5. kérdés) a mintán megadott (12. ábra) képlet eredményét kellett a tanulóknak meghatározni, mely a következő: a meghatározza (kiírja) az L betűvel kezdődő felhasználók számát.

$$\{=SZUM(HA(BAL(A2:A251)="L";1))\}$$

12. ábra: A tömbképlet feladatban szereplő képlet.

Tudástranszfert mérő tudáselemek:

- matematika: műveletek végrehajtási sorrendje,
- matematika: átlag,
- informatika: HA(), SZUM(), BAL() és ÁTLAG() függvények,
- matematika: összetett függvény, többváltozós függvény,
- programozás: függvényhívás,
- programozás: függvényírás.

A feladat pontozásakor a megértés SOLO kategóriáit vettük figyelembe (7. táblázat). A feladat megoldásának algoritmusát a 8. táblázat mutatja be.

Pontszám	Megoldás
4	hibátlan
3	számlálás
3	összegzés és az L kezdőbetűre szűrés
2	összegzés
1	az L kezdőbetű vagy a balról 1 karakter vizsgálata
1	ciklusleírás
0	minden más eset (beleértve a „hibás a képlet” megoldást is)

7. táblázat: A tömbképlet feladat pontozása a megértés SOLO kategóriáit figyelembe véve.

1.	<i>Input:</i> szöveg Meghatározzuk a szöveg első karakterét. <i>Output:</i> a szöveg első karaktere
2.	<i>Input:</i> a szöveg első karaktere és L betű Feltesszünk egy eldöntendő kérdést: Az előző lépés eredménye (az első betű) egyenlő L-lel? <i>Output:</i> egy vektor IGAZ és HAMIS értékekkel
3.	<i>Input:</i> egy vektor IGAZ és HAMIS értékekkel és 1 Ha igaz, akkor a rekordot megjelöljük egy 1-essel, ha nem figyelmen kívül hagyjuk. <i>Output:</i> egy vektor 1-es és HAMIS értékekkel
4.	<i>Input:</i> egy vektor 1-es és HAMIS értékekkel Összeadjuk az 1-eseket. <i>Output:</i> egy egész szám

8. táblázat: Algoritmus a feltételes számlálás problémájára tömbképlettel történő megoldás esetén.

3.3. Szövegkezelés

A szövegkezelés témakörből a számozás feladat (4. függelék, F7) eredményeit vizsgáltuk, melynek középpontjában az automatikus számozás és a kézi számozás megkülönböztetése áll. A tanulók feladat az volt, hogy megjelöljék a mintán szereplő (13. ábra) helyes számozásokat. A grafikus felhasználói felületen a kurzor pozíciója elegendő ahhoz, hogy megállapítsuk, hogy egy számozás kézi vagy automatikus.

- A 1.A növényrészei
- B **3.Termodinamikus kölcsönhatás**
- C b.) Gabonafélék
- D ♥ → **Petőfi szerelmi költészete**
- E 4. Magyarázza el a ciklusok működését!

13. ábra: A számozás feladat mintái.

Tudástranszfert mérő tudáselemek:

- automatikus számozás,
- kurzorpozíció,
- nem-nyomtatódó karakterek,
- tipográfia.

A feladatban 2 pontot szerzett a tanuló, ha csak a helyes számozást jelölte meg (E-vel jelölt válaszlehetőség). Amennyiben a kiválasztott számozások között a jó válaszon kívül egy helytelen válasz is szerepelt, akkor 1 pontot szerzett, minden más esetben 0 pontot.

3.4. Minta

Az előzetes egyeztetések alapján a tervezett kitöltői létszám 93 iskolából 11 425 fő volt. A teszt átvételétől kezdve 2 hét állt az iskolák rendelkezésére. 8 880 kitöltött tanulói teszt érkezett vissza hozzánk, 8 788 papír alapon és 92 online. A kitöltők 52,3%-a fiú, 47,7%-a lány. A mérés célcsoportja a 7–10. évfolyamos tanulók, mely kiegészült további kitöltőkkel az (5–6. és 11–13. évfolyamról).

4. Tézisek

- [T1] Az eszközcentrikus táblázatkezelési megközelítések negatívan befolyásolják a hatékony tanulói adat- és táblázatkezelést, valamint az önértékelést. Az eszközökre fókuszálva a tanulók alacsony szintű adatkezelési ismeretekkel rendelkeznek, amely megakadályozza a hatékony problémamegoldást, ugyanakkor teret enged egy irreálisan magas önértékeléshez.
- [T2] A digitális eszközök, illetve azok iskolai és magánéleti használata nem fejleszti hatékonyan a tanulók adat- és táblázatkezelési ismereteit, képességeit, kompetenciáit.
- [T3] Az automatikus számozás szemantikája alapján a szükséges egy bitnél lényegesen több információt kell a tanítás-tanulás csatornájára helyezni ahhoz, hogy a végfelhasználókhöz érthetően eljusson és hatékonyan tudják azt alkalmazni.
- [T4] A természetes nyelvi digitális szövegekben a módosítási feladatok elvégzése több adatot igényel a hibás dokumentumokban, mint azok helyes megfelelőjében.
- [T5] A fenntarthatósági arány – a $[0; 1]$ intervallumon értelmezve – hibás dokumentumokban alacsonyabb, mint ezek helyes megfelelőjében.

5. Táblázatkezelés és önértékelések

A fejezet célja a [T1] tézis vizsgálata, mely szerint az eszközcentrikus táblázatkezelési megközelítések negatívan befolyásolják a hatékony tanulói adat- és táblázatkezelést, valamint az önértékelést. Az eszközökre fókuszálva a tanulók alacsony szintű adatkezelési ismeretekkel rendelkeznek, amely megakadályozza a hatékony problémamegoldást, ugyanakkor teret enged egy irreálisan magas önértékeléshez.

	A	B	C	D
1	Felhasználó	Feltöltés	Feliratkozó	Megtekintés
2	VamosART	484	1,107,555	226,195,766
3	Videómánia	338	833,23	254,545,702
4	PamKutya	120	809,866	223,441,355
5	LetsGoMartin	176	725,638	162,798,559
6	TheVR	1,062	592,675	213,550,948
7	luckeY	1,183	561,13	150,341,428
8	Peter Gergely	100	548,241	79,713,757
9	Scribble Netty	159	546,049	74,234,471
248	Szilvaglam	87	61,899	3,918,538
249	rance flow	524	61,863	53,275,385
250	KIS GRÓFO (official)	9	61,65	30,712,031
251	KODIAK	736	61,467	14,599,194

14. ábra: Az adattípusok, adatelemzés feladatban szereplő táblázat.

Az adattípusok feladatban szereplő táblázat (14. ábra) egyes oszlopaiban előforduló adattípusokra adott válaszok számát a 9. táblázat mutatja, kiemelve a helyes válaszokat (9. táblázat).

Oszlop	Egész szám	Valós szám	Szöveg	Logikai	Dátum	Nincs
A	642	485	5774	284	138	372
B	3722	3374	860	437	206	259
C	2138	3711	1201	793	311	392
D	2370	2596	1159	1078	290	650

9. táblázat: Az egyes oszlopokban az adattípusokra adott válaszok száma, kiemelve a helyes válaszok.

A szélsőérték feladatban (B oszlopban látható legkisebb és legnagyobb érték meghatározása) a tanulók többsége 1183-at írt a legnagyobb és 9-et a legkisebb számra. A helytelen válaszokat figyelembe véve arra következtethetünk, hogy ezek a tanulók nem voltak tisztában a vessző szemantikájával; a vesszőt ezres elválasztó karakterként ismerték fel, és nem vették észre a szere-

pének megváltozását. A B oszlopban szereplő legkisebb értékre 865 tanuló válaszolt helyesen, ami a minta 10%-a, a maximális értékre pedig 963 tanuló, ami a minta 11%-a.

Az adatrekordok számát tekintve a tanulók 17%-a válaszolta helytelenül, hogy 251 adatrekordot tartalmaz a táblázat, 357 fő a 12-t, 365 fő a 13-at, 954 fő pedig 3-at vagy 4-et válaszolt. Azok a tanulók, akik a 251-es választ adták, felismerték, hogy a táblázatban rejtett sorok vannak, de nem választották el a mezőnevek sorát az adatrekordoktól. Azok, akik 12-es választ adtak, tisztában vannak az adatrekord és a mezőnév fogalmával, de nem vették észre, hogy vannak rejtett sorok a táblázatban. Ezzel szemben a 13-as választ adók nem vették észre sem a rejtett sorokat, sem a mezőnév sorát. Mindössze annyit tettek, hogy megszámozták a mintatáblázat sorainak számát. Azok, akik 3 vagy 4 választ adtak, nem ismerték sem az adatrekord, sem a mezőnév definícióját.

Függetlenségi és homogenitási tesztet végeztünk, hogy feltárjuk az összefüggést az egész és valós számok felismerése között, és megállapítottuk, hogy szimmetrikusan hibáznak, ami azt jelenti, hogy ha felismerik az egyik adattípust, akkor nincs garancia arra, hogy a másikat is felismerik, tehát tudásuk nem stabil, mely bizonyítéka a [T1] tézisnek.

Annak vizsgálatára, hogy azokban az esetekben, amikor a tanulók felismerik a legnagyobb számot egy vektorban, a legkisebb számot milyen arányban ismerik fel helyesen, függetlenség vizsgálatot végeztünk a Chi-négyzet tesztel. A kereszttáblázat eredménye alapján 0,85 valószínűséggel mondhatjuk, hogy a mátrix aszimmetriája miatt abban az esetben, amikor a tanulók felismerik a legnagyobb számot egy vektorban, a legkisebb számot is jelentősen nagyobb arányban ismerik fel helyesen ($p < 0,001$). Megállapíthatjuk azonban, hogy azok a diákok, akik felismerték az egyik adattípust, nem feltétlenül ismerték fel a másikat is. Másrészt az elemzés bebizonyította, hogy ha az egyik szélsőértéket megtalálják, akkor a másikat is. Ezek az eredmények tovább erősítik a [T1] tézist. Az értékelés ezen szakaszában azonban nem volt egyértelmű, hogy mi a kapcsolat az adattípusok és a szélsőértékek között.

Feltételeztük, hogy ha a tanulók mindkét adattípust felismerik (egész és valós), akkor a B oszlopban a legkisebb és a legnagyobb értékeket is meg tudják határozni. Ennek vizsgálatára Chi-négyzet tesztet végeztünk. Az eredmények azt mutatják, hogy a tanulók nagyobb arányban ismerték fel a szélsőértékeket azokban az esetekben, amikor felismerték az adattípusokat ($p < 0,001$). Az adattípusok egyikét sem felismerő tanulóknak van a legalacsonyabb eredményük a szélsőértékek megtalálásában. Azok közül, akik felismerték valamelyik adattípust, 10,6% ismerte fel a legkisebb értéket és 11,9% a legnagyobb értéket. Azok közül, akik mindkét adattípust felismerték, 18,1% helyesen határozta

meg a legkisebb, illetve 19,3% a legnagyobb értéket. A párok összehasonlításából kiderült, hogy mindössze 154 olyan tanuló van, aki mind az adattípusokat (egész és valós), mind a szélsőértékeket felismerte. Ez az eredmény azt mutatja, hogy mindössze 154 tanuló volt képes a feladat megoldásához szükséges tudáselemek – egész és valós adattípusok, a számok magyar nyelvi szintaktikája és a vessző értelme a számokban – kombinálására, ami a minta 1,7%-át jelenti.

$$=HA(ÁTLAG(D2:D58)-50<A5;"a";""))$$

15. ábra: A műveleti sorrend feladatban megadott képlet.

A műveleti sorrend feladatot (15. ábra) a táblázatkezelői környezetre adaptált SOLO kategóriák alapján értékeltük és megállapítottuk, hogy a tanulók 81,7%-a Prestructural kategóriába került a feladatban elért pontszám alapján, továbbá mindössze 8,9%-a került a Relational kategóriába. Ezek alapján elmondható, hogy a tanulóknak függetlenül a matematika és az informatika órán megszerzett előzetes tudástól, problémát okoz a helyes műveleti sorrend meghatározása, amely tovább erősíti a [T1] tézist.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2">A</th></tr> <tr><td style="width: 20px; text-align: center;">1</td><td>SZUM(A2 A20</td></tr> </table>	A		1	SZUM(A2 A20	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2">A</th></tr> <tr><td style="width: 20px; text-align: center;">1</td><td>SZUM(A2 A20</td></tr> </table>	A		1	SZUM(A2 A20
A									
1	SZUM(A2 A20								
A									
1	SZUM(A2 A20								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2">A</th></tr> <tr><td style="width: 20px; text-align: center;">1</td><td>=SZUM(A2:A20)</td></tr> </table>	A		1	=SZUM(A2:A20)	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2">A</th></tr> <tr><td style="width: 20px; text-align: center;">1</td><td>=SZUM(A2;A20)</td></tr> </table>	A		1	=SZUM(A2;A20)
A									
1	=SZUM(A2:A20)								
A									
1	=SZUM(A2;A20)								
A2-től A20-ig	A2 és A20								

16. ábra: A képletkiegészítés feladat (felül) és megoldása (alul) magyar nyelvi táblázatkezelői beállításokkal.

Azt a tudást, hogy a zárójelek mindig párban működnek – amely tudás szintén a matematikából származik – alkalmazni kell a képletkiegészítési feladatban (16. ábra). A két képletkiegészítő feladatban a zárójelek összehasonlításából kiderült, hogy 3106 tanuló töltötte ki az első képletet, 1985 tanuló a másodikat és 1949 tanuló mindkét képletet. Az első és a második képletben a berekesztő zárójelek között erős korreláció mutatható ki ($R = 0,707$). Ezzel szemben a többi tudáselem között – kivéve az első berekesztő zárójel és az első

tartományírásjel ($R = 0,82$) – közepes vagy gyenge korreláció mutatható ki (10. táblázat). Megállapíthatjuk, hogy a két legerősebb tudáselem a sémából és a matematikából kerül átadásra. Itt azonban meg kell jegyeznünk, hogy a séma – $SZUM(A2:A21)$ vagy $SZUM(A2,A21)$ – nem jár együtt a kifejezések szemantikájával ($R = 0,323$ és $R = 0,324$), és a másik megoldás tartományírásjével sem ($R = 0,462$, nincs a táblázatban).

	2. berekesztő zárójel	1. egyenlőségjel	1. tartományírásjel	1. tartományhivatkozás
1. berekesztő zárójel	0,707	0,451	0,820	0,323
	1. berekesztő zárójel	2. egyenlőségjel	2. tartományírásjel	2. tartományhivatkozás
2. berekesztő zárójel	0,707	0,508	0,549	0,324

10. táblázat: A képletkiegészítés feladatban a képletek berekesztő zárójele és a többi elem közötti kapcsolat.

Összehasonlítottuk a képletkiegészítés és a végrehajtási sorrend feladatot a tömbképlet feladattal és megállapítottuk, hogy a SOLO kategóriák szerinti tudásnövekedés az első két feladatban – még a Relational szintet is elérve – nem eredményezi a növekedést a tömbképlet feladatban. Ezek az eredmények azt mutatják, hogy a szintaktikai és a matematikai tudáselemek továbbra sem kapcsolódnak egymáshoz, a tanulók nem képesek alkalmazni a matematikából származó ismereteiket a programozás-orientált táblázatkezelési problémák megoldása során. A matematikából származó tudástranszfer-aktiválás lehetővé teszi, hogy a tanulók alapvető matematikai ismereteket alkalmazzanak (zárójelek), illetve implicit matematikai problémákat oldjanak meg (végrehajtási sorrend). Ha azonban a matematikai háttér csak explicit módon van jelen, a tanulók nem képesek alkalmazni azt.

Az adattípusok feladatban 25%, a képletkiegészítés feladatban 61%, a végrehajtási sorrend feladatban 76% és a tömbképlet feladatban 93% a nulla pontos megoldás. A maximális pontszámot 1%, 1%, 7% és 1% tanuló érte el az egyes feladatokban. A módusz 3, a medián 2 az adattípusok feladatban, a többiben a módusz és a medián is 0.

Mivel az adattípusok feladatban a véletlenszerű jelölések részben helyes válaszokhoz vezetnek, továbbá a feladat nem kapcsolódik szorosan a másik három feladathoz (képletkiegészítés, végrehajtási sorrend, tömbképlet), ezért a továbbiakban az adattípusok feladatot kihagytuk az elemzésből. Megállapítható, hogy a három feladat eredményeinek átlagai között

szignifikáns különbség van ($p < 0,001$). A három feladat összehasonlításakor (Spearman-féle korreláció) megállapítottuk, hogy a képletkiegészítés és a végrehajtási sorrend feladatok között közepes korreláció van ($R = 0,453$). A tömbképlet és a másik két feladat között gyenge korrelációt találunk ($R = 0,206$ a képletkiegészítés és $R = 0,263$ a végrehajtási sorrend esetében).

A Chi-négyzet keresztábrák a feladatpárok közötti összefüggések részleteit tárják fel (11. táblázat). Ha a tanulók tudása Prestructural a képletkiegészítési feladatban, akkor 91%-os valószínűséggel a végrehajtási sorrend feladatban is Prestructural. Ahogy a megértés szintje növekszik a képletkiegészítési feladatban, úgy a megértés szintje Prestructural vagy Relational a végrehajtási sorrend feladatban. Ez alapján megállapíthatjuk egyrészt, hogy a szintaktikai tudás nem segíti a végrehajtási sorrend felismerését, másrészt a képletkiegészítési feladatban nyújtott jobb teljesítmény lehetővé teszi a tudástranszfer aktiválását, amelynek révén a kifejezések végrehajtási sorrendjére a matematikából következtetünk. Úgy tűnik, hogy ezek a matematikai tudáselemek sokkal inkább megszilárdultak, mint a tanulók táblázatkezelési ismeretei.

		végrehajtási sorrend			
		P	U	M	R
képletkiegészítés	P	91,0%	3,2%	2,1%	3,7%
	U	52,9%	7,3%	9,7%	30,2%
	M	39,1%	10,6%	10,3%	40,0%
	R	84,3%	3,9%	3,3%	8,5%

11. táblázat: A képletkiegészítés és a végrehajtási sorrend feladatok összehasonlítása SOLO kategóriák szerint.

Ahogy a 3.1. fejezetben említésre került, a tanulónak értékelni kellett tudásukat az egyes témakörökben, továbbá be kellett jelölniük, hogy tanulták-e az egyes témaköröket. Összehasonlítottuk és elemeztük a *Tanult*, a *Nem tanult* és a *Nem válaszolt* csoportok eredményeit. Annak ellenére, hogy a feladatokban elért átlagok rendkívül alacsonyak, a kérdés az volt, hogy van-e különbség a *Tanult* és a *Nem tanult* csoportokban a feladatokban elért átlagos pontszámok között (12. táblázat, 13. táblázat). A tömbképlet feladatban olyan alacsony azoknak a tanulónak a száma, akiknek az eredménye nagyobb nullánál, hogy nem tudtuk őket két csoportra szétválasztani. Következésképpen a feladatot kihagytuk az elemzésből.

	Mindenki	7. osztály	8. osztály	9. osztály	10. osztály
Összesen	1,32	0,75	1,28	1,28	1,79
Tanult	1,52	1,06	1,37	1,42	1,96
Nem tanult	0,27	0,16	0,44	0,32	0,44
Nem válaszolt	0,82	0,40	0,79	1,02	0,87

12. táblázat: A tanulók eredményei a képletkiegészítés feladatban a 7–10. évfolyamon.

	Mindenki	7. osztály	8. osztály	9. osztály	10. osztály
Összesen	0,86	0,54	0,89	0,74	1,21
Tanult	0,99	0,78	0,95	0,82	1,33
Nem tanult	0,14	0,09	0,26	0,17	0,17
Nem válaszolt	0,56	0,29	0,58	0,62	0,66

13. táblázat: A tanulók eredményei a végrehajtási sorrend feladatban a 7–10. évfolyamon.

Az alacsony átlagok ellenére jelentős különbség mutatkozik a *Tanult* és a *Nem tanult* csoportok között. Az összes különböző csoportot figyelembe véve, a *Tanult* csoport eredményei mindkét feladatban magasabbak, mint a *Nem tanult* csoporté. A *Nem válaszolt* csoport által elért eredmény a másik két csoport eredménye közötti érték. A következő táblázatok mutatják, hogy a *Tanult* és a *Nem tanult* csoportba tartozó tanulók hány százaléka kapott 0–8 pontot a képletkiegészítés (14. táblázat) és 0–6 pontot a végrehajtási sorrend feladatban (15. táblázat).

	0	1	2	3	4	5	6	7	8
Tanult	56,1%	4,0%	11,3%	11,8%	5,6%	4,3%	4,4%	0,9%	1,6%
Nem tanult	89,8%	2,2%	2,9%	2,6%	1,7%	0,7%	0,1%	0,0%	0,0%

14. táblázat: A tanulók eredményeinek eloszlása a képletkiegészítés feladatban pontszámok szerint.

	0	1	2	3	4	5	6
Tanult	72,4%	6,1%	4,0%	3,8%	3,1%	2,0%	8,5%
Nem tanult	94,8%	1,6%	1,7%	0,4%	0,2%	0,2%	0,7%

15. táblázat: A tanulók eredményeinek eloszlása a végrehajtási sorrend feladatban pontszámok szerint.

A képletkiegészítés feladatban a nulla pontot elért tanulókat leszámítva, a két leggyakoribb a három és a kettő pontos megoldás (872, illetve 855 tanuló). Ez azt jelenti, hogy a legtöbb tanuló tudása Presturctural. A két leggyakoribb

pontszám nem függ attól, hogy a diákok tanultak-e táblázatkezelést az iskolában vagy sem.

A végrehajtási sorrend feladatban a két leggyakoribb pontszám a hatos és az egyes (629, illetve 450 diák). Ez azt jelenti, hogy a diákok többsége – csak azokat figyelembe véve, akik legalább egy pontot kaptak – vagy csak egy lépést ismert fel vagy az összeset a helyes sorrendben (15. táblázat). Ebben a feladatban a két leggyakoribb elem a *Tanult* csoportban a hat és az egy (584 és 422 fő), míg a *Nem tanult* csoportban a kettő és az egy (16 és 15).

Az elemzésnek a következő szakaszában a táblázatkezelésre vonatkozó önértékelési értékeket és a tanult/nem tanult az iskolában kérdésekre adott válaszokat vettük figyelembe. A diákok eredményeinek értékeléséhez hasonlóan a 0–5 önértékelési értékeket SOLO kategóriákká alakítottuk át (16. táblázat). Ezek a kategóriák teljes mértékben megfelelnek a PISA 2018-ban a természettudományos tudásszinteknek, ahol a 2. szint jelenti a minimumot [173]. Itt kell megjegyeznünk, hogy 516 diák nem válaszolt erre a kérdésre.

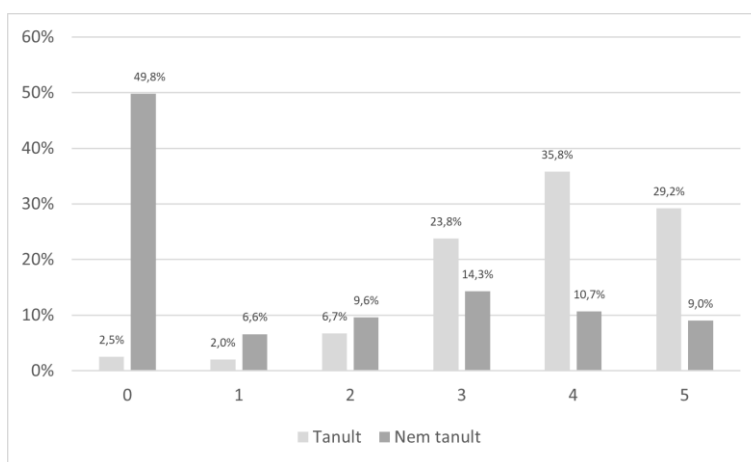
Önértékelési érték	SOLO kategória	A természettudományos tudásszint a PISA-ban
5	Relational vagy Extended abstract	elmagyarázza az ismeretlen, összetettebb kifejezéseket
4	Multistructural	két vagy több független változó
3	Unistructural	közepesen összetett tartalmi ismeretek
2	Unistructural	mindennapi tartalmi ismeretek és alapvető eljárási ismeretek
1	Prestructural	gyengén teljesítők
0	Prestructural	

16. táblázat: Az önértékelési értékek SOLO kategóriái a PISA 2018 [173] teljesítményszintjeivel összehasonlítva.

Ezek a megértési kategóriák összhangban vannak az Európai Képesítési Keretrendszerrel (EQF) [182], és ezen dokumentum alapján a DigComp 2.1-gyel [183]. A teszt adminisztrációs folyamata miatt az EQF és a DigComp első szintje nem vizsgálható, ahol a munkát vagy tanulást közvetlen felügyelet mellett, strukturált környezetben végzik. Hasonló módon a 7. és 8. szintek sem azonosíthatók a tesztelési eljárás korlátjai miatt.

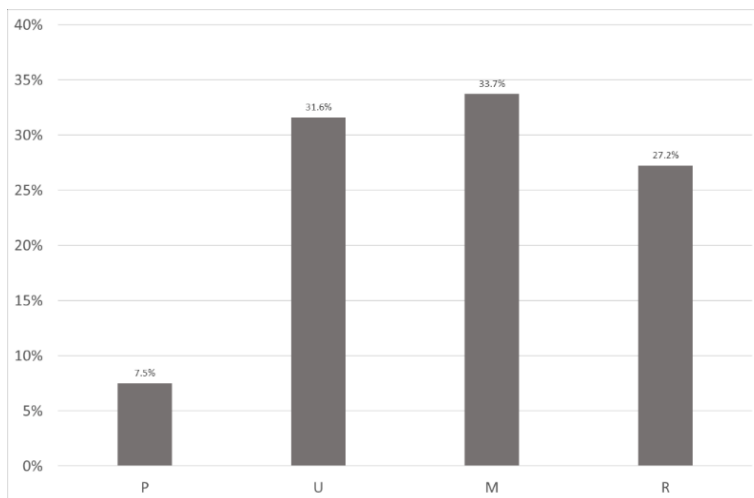
A méréskor érvényes kerettantervek [24][25] szerint a táblázatkezelés oktatása-tanulása a 7–10. osztályig kötelező. Felmerül tehát a kérdés, hogy a vizsgált diákok 10,79%-a miért állította, hogy nem tanult táblázatkezelést az isko-

lában – vagy azért, mert nem tanulták vagy azért, mert nem tudták, hogy tanultak –, továbbá 8,36% miért nem válaszolt a kérdésre. A *Tanult* és a *Nem tanult* csoportok összehasonlítása azt mutatta, hogy azok a diákok, akik az iskolában tanultak táblázatkezelést, magasabbra értékelték tudásukat, mint azok, akik nem tanultak. Sőt, 65,1%, 19,7% és 29,8% értékelték négyes vagy ötös szintre a tudását a *Tanult*, a *Nem tanult* és a *Nem válaszolt* csoportokban. Az önértékelésük alapján a tudásuk Multistructural vagy Relational. A *Tanult* csoportban 5% állította, hogy annak ellenére, hogy az iskolában tanult táblázatkezelést, a tudása nem létezik, és csak a legalacsonyabb, Prestructural szintű a megértése. A *Nem tanult* csoportban 56%, a *Nem válaszolt* csoportban 40% állította, hogy a tudása Prestructural.



17. ábra: Önértékelés a táblázatkezelés iskolai tanulása kapcsán.

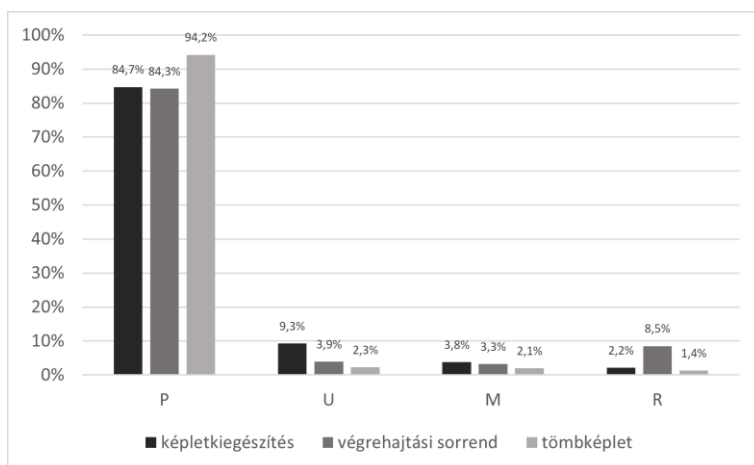
Összesen a *Tanult* csoport 4,5%-a rendelkezik Prestructural ismeretekkel (nulla (2,5%) vagy egy (2%)). Ezzel szemben a *Nem tanult* csoport 49,8%-a nullával, 6,6%-a eggyel értékelté tudását. Ez azt jelenti, hogy a *Nem tanult* csoport diákjainak 56,4%-a rendelkezik Prestructural szintű ismeretekkel az önértékelések alapján (17. ábra).



18. ábra: Az önértékelési értékekhez rendelt összesített SOLO kategóriák.

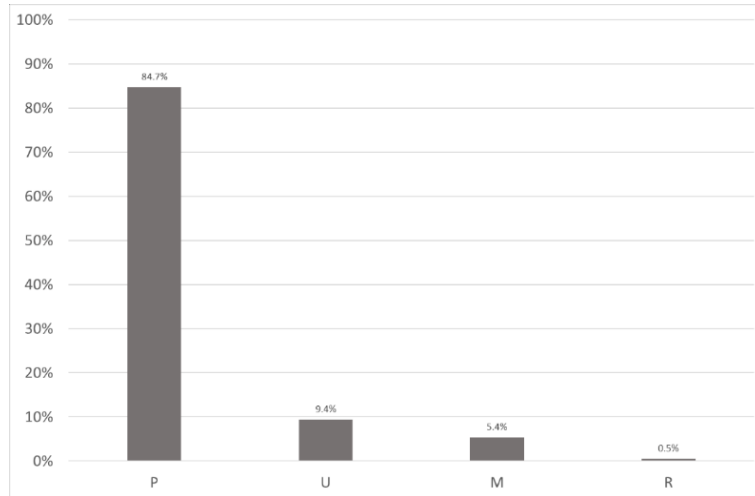
Az összes tanulót figyelembe véve az önértékelésnél 7,5% Prestruktural, 31,6% Unistuktural, 33,7% Multistuktural, 27,2% Relation (18. ábra). A kategóriák alapján feltételezhető, hogy a tanulók szilárd táblázatkezelői ismeretekkel rendelkeznek.

A tanulók táblázatkezelésben elért eredményeit is SOLO kategóriákba [67] konvertáltuk, mely alapján kiszámítottuk az egyes kategóriákban a gyakoriságot és a relatív gyakoriságot. Az elemzésből kiderül, hogy a leggyakoribb SOLO kategória mindhárom feladatban a Prestruktural (P) (19. ábra).



19. ábra: A tanulók eredményei a három feladatban, a SOLO megértési kategóriái alapján értékelve.

A három feladat SOLO kategóriái alapján minden egyes tanuló eredményéhez egy összesített SOLO kategóriát rendelünk. Az összesített SOLO kategóriák ugyanazokat az eredményeket mutatják, mint az egyedi SOLO kategóriák: a tanulók megértési szintje elsősorban Prestructural (20. ábra).



20. ábra: A diákok eredményei a SOLO kategóriák összesített értékelése alapján.

A tanulók önértékelései alapján felállított SOLO kategóriáinak (18. ábra) és eredményeinek (20. ábra) összehasonlítása azt mutatja, hogy az önértékelés értékei jóval magasabbak, mint a tanulók valós tudása, amely a [T1] tézist igazolja.

A 17. táblázat az önértékelés (BP, BU, BM, BR) (B jelöli a becslést) és az eredmények (EP, EU, EM, ER) (E jelöli az eredményt) összehasonlításában szereplő tanulók számát mutatja be. A mátrix főátlójában szerepel azon diákok száma, akik valós önértékelési értéket adtak meg, a mátrix felső háromszögében azon tanulók száma, akik túlbecsülték tudásukat, a mátrix alsó háromszögében azon tanulók száma, akik alábecsülték tudásukat.

	BP	BU	BM	BR
EP	586	2235	2199	1700
EU	9	190	320	270
EM	5	93	161	192
ER	0	8	18	15

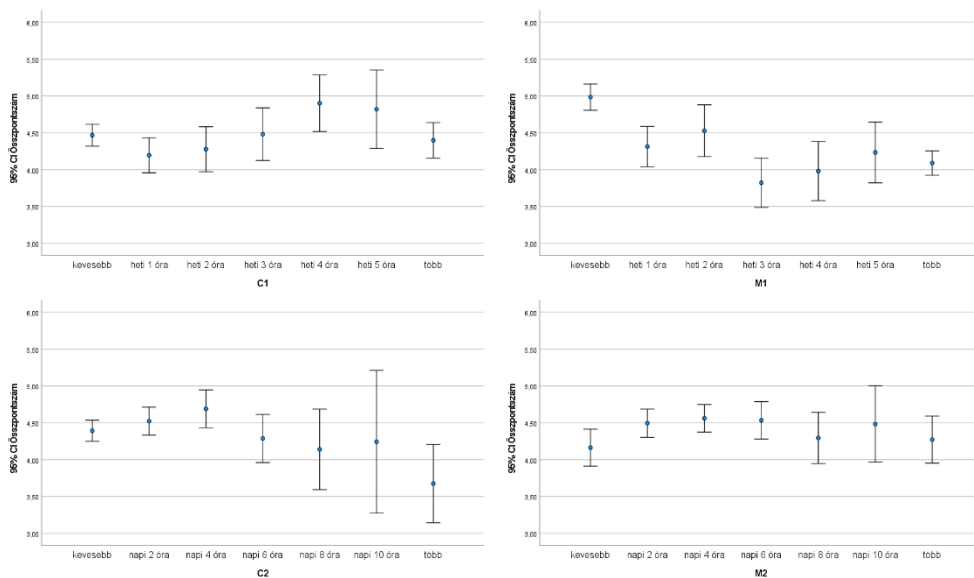
17. táblázat: Az önértékelési értékek (BP, BU, BM, BR) (B jelöli a becslést) SOLO kategóriái a tanulók eredményeivel (EP, EU, EM, ER) (E jelöli az eredményt) összehasonlítva.

Általánosságban elmondható, hogy a tanulók 12%-a helyesen tudta megbecsülni táblázatkezelési ismereteit, 86%-a a tényleges tudásuknál magasabb önértékelést adott, 2%-a pedig alábecsülte tudását, mely a [T1] tézist támasztja alá.

6. Digitális eszközökre fordított idő

Ebben a fejezetben a digitális eszközökre fordított idő és a tanulók táblázatkezelési kompetenciáinak, valamint önértékelési értékeinek kapcsolatát vizsgálom. [T2] tézisem, hogy a digitális eszközök, illetve azok iskolai és magánéleti használata nem fejleszti hatékonyan a tanulók adat- és táblázatkezelési ismereteit, képességeit, kompetenciáit.

Célunk volt annak vizsgálata, hogy a digitális eszközök, illetve azok iskolai és személyes használata hogyan befolyásolhatja a táblázatkezelési kompetenciákat és az adatok táblázatkezelési környezetben történő kezelését. A vizsgálatokhoz a digitális eszközökre fordított időt (1. függelék, G2–G5), valamint a táblázatkezelés témakör feladatait (6. függelék, F14, F15, F17) használtuk. Ahogyan az előző fejezetben bemutatásra került, a tanulók minden feladatban alacsony pontszámot értek el, és a nulla pontok rendkívül magas száma deformálja a magasabb pontszámmal rendelkezők eredményeit. Az adattípusok feladat kivételt képez, mivel ez egy olyan feleletválasztós kérdés, ahol a véletlenszerű jelölések részben helyes válaszokhoz vezetnek. Az összes helyes válasz száma azonban ebben a feladatban nem magasabb, mint a többi feladatban.



21. ábra: A diákok táblázatkezelési feladatokban elért eredményei az eszközökre fordított idő függvényében (összesen 27 pont volt szerezhető a témakörben).

A tanulók összesített eredményeit a digitális eszközökre fordított idő függvényében a 21. ábra mutatja be. Szignifikáns különbségeket találtunk a meg-

jelölt időértékek között az iskolai mobilhasználat (M1) ($p < 0,05$) és az általános számítógéphasználat (C2) ($p = 0,020$) esetében. Az iskolai számítógéphasználat (C1) és az általános mobilhasználat (M2) esetében nem találtunk szignifikáns különbségeket ($p = 0,075$, illetve $p = 0,216$). Ez az eredmény alátámasztja a [T2] tézist és azt jelzi, hogy a számítógépek iskolai használata nem befolyásolja a tanulók táblázatkezelési ismereteit. Az ezeken az eszközökön végzett tantermi tevékenységek nem kapcsolódnak a táblázatkezeléshez (C1). Ugyanez igaz a napi mobilhasználatra is (M2).

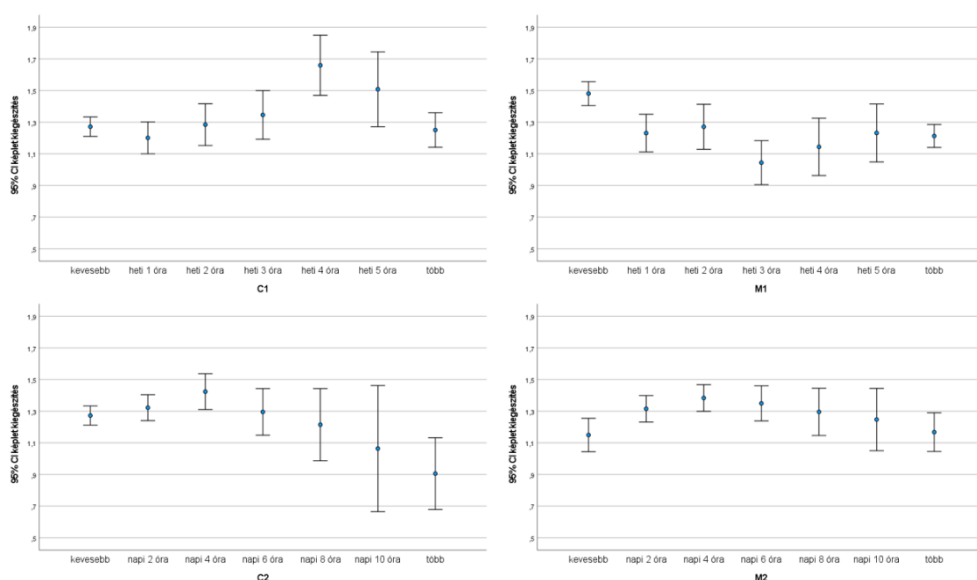
A szignifikáns különbségeket (M1, C2) Games-Howell Post hoc teszttel ellenőriztük tovább. A teszt egyértelműen jelzi, hogy az iskolai mobilhasználat (M1) negatív hatással van a tanulók táblázatkezelési ismereteire. Minél kevesebb időt töltenek ezekkel az eszközökkel az iskolában, annál magasabb pontszámot értek el a táblázatkezelés és az adatkezelés témakör feladataiban, azaz az, hogy több a mobilhasználat az iskolában, az nem fejleszti a tanulók táblázatkezelési ismereteit. A napi számítógéphasználat (C2) esetében mérsékelt fejlődés tapasztalható, mivel a számítógépet egyáltalán nem használó tanulók eredményei alacsonyabbak, mint azok eredményei, akik napi 4 órás számítógéphasználatot jelöltek meg. A kevesebb, mint 2 órás, 2 óra körüli és 4 óra körüli csoportok között nincs szignifikáns különbség. Ezek az eredmények azt jelzik, hogy a táblázatkezelést mérő feladatok eredményét nem befolyásolja a digitális eszközökön töltött idő. Ez az eredmény teljes összhangban van a PISA 2009 [62] eredményeivel, továbbá azt is jelzi, hogy a digitális eszközök elterjedésének nincs meg a feltételezett hatása a tanulók digitális készségeire, képességeire [72], mely a [T2] tézist támasztja alá.

A következő fázisban a feladatok eredményeit ANOVA-val elemeztük. A kérdés az volt, hogy vannak-e különbségek az egyes feladatok között, amelyek különböző kompetenciákat igényelnek vagy sem. Az eredményeket a 18. táblázat mutatja be, kiemelve a szignifikáns különbségeket.

	képletkiegészítés	végrehajtási sorrend	adattípusok	adatelemzés	tömbképlet
C1	0,001	0,185	0,112	0,020	0,176
M1	0,000	0,000	0,000	0,000	0,06
C2	0,005	0,322	0,104	0,062	0,723
M2	0,019	0,280	0,317	0,007	0,254

18. táblázat: Az eszközökre fordított idő összehasonlítása a tanulók eredményeivel a táblázatkezelési feladatokban.

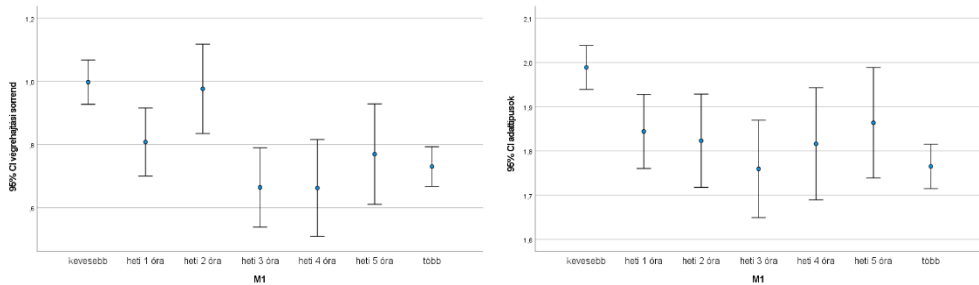
A feladatok közötti különbségek okainak megtalálása érdekében az eredményeket Games-Howell Post hoc teszttel ellenőriztük. A képletkiegészítés feladatban szignifikáns különbségeket találtunk az összes (C1, M1, C2, M2) eszközcsoportban (18. táblázat, 22. ábra). A görbeillesztési módszernél minden eszközcsoport esetében az volt a célunk, hogy olyan függvényt találjunk, amely a legjobban illeszkedik a szórt pontokhoz. Az iskolai számítógéphasználat (C1) esetében a legjobb illeszkedés egy konstans ($c = 1,281$). Az iskolai mobilhasználatot (M1) figyelembe véve, a képletkiegészítés feladatban azok a diákok érték el a legjobb eredményeket, akik a legkevesebbet használják ezeket az eszközöket az iskolában. A legjobb illeszkedés az $f(x)=0,027x^2-0,2x+1,5$ parabola ($\min_p = 3,7$, $\min_v = 1,13$). Ez azt jelenti, hogy a mobil eszközök körülbelül heti 4 órás használata befolyásolja a legkevésbé a táblázatkezelési ismereteket. Továbbá, az ilyen eszközökkel töltött több idővel a tanulók nem érik el a kevesebb időt töltők eredményeit. Megállapíthatjuk, hogy a mobil eszközökön való játék az iskolában nem tanítja meg, hogyan kell a legegyszerűbb táblázatkezelői környezetben megadott képletet kitölteni, mely szintén alátámasztja a [T2] tézist.



22. ábra: A tanulók eredményei a képletkiegészítés feladatban az eszközökre fordított idő függvényében.

A napi számítógép- (C2) és mobilhasználat (M2) pozitívabb hatással van a képletkiegészítés feladatra. Az eredmény azt mutatja, hogy a digitális eszközök mérsékelt – napi 4 óránál nem több – használata segíti a táblázatkezelők szintaktikai szabályainak megértését. Az illeszkedő görbe a napi számítógéphasználat esetében az $f(x)=-0,008x^2+0,059x+1,275$ parabola ($\max_p = 3,687$,

$\max_v = 1,383$), míg $f(x) = -0,006x^2 + 0,064x + 1,206$ ($\max_p = 5,3$, $\max_v = 1,377$) a mobil eszközök esetében (22. ábra).

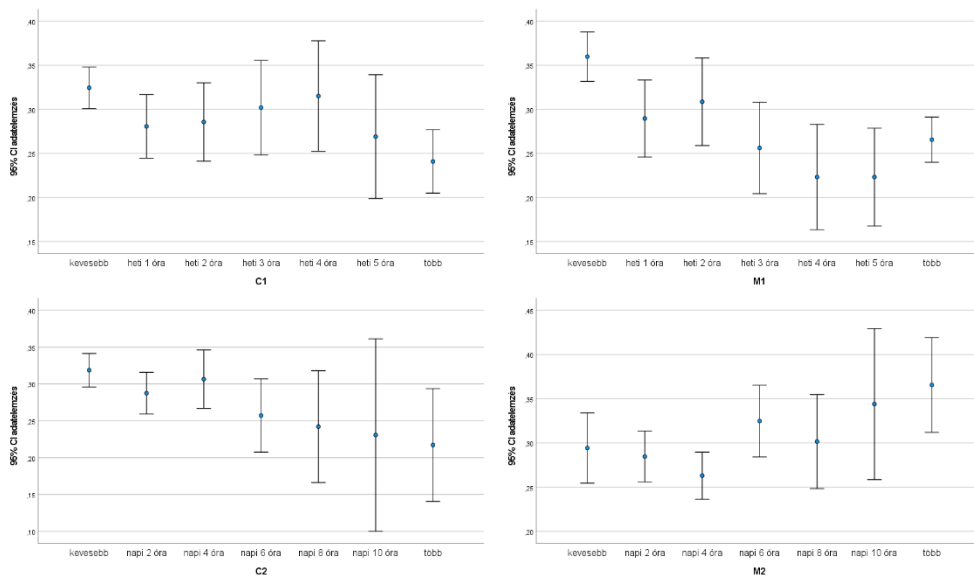


23. ábra: A tanulók eredményei a végrehajtási sorrend és az adattípusok feladatban az iskolai mobil eszközök használatának függvényében.

A végrehajtási sorrend és az adattípusok feladatokban szignifikáns különbséget találtunk a mobil eszközök iskolai használatában (M1: $p = 0,000$ mindkét feladatban) (18. táblázat, 23. ábra). Az illeszkedő görbék $f(x) = 0,013x^2 - 0,123x + 1,012$ ($\min_p = 4,73$, $\min_v = 0,72$) a végrehajtási sorrend feladatban és $f(x) = 0,009x^2 - 0,089x + 1,982$ ($\min_p = 4,94$, $\min_v = 1,76$) az adattípusok esetében. A képletkiegészítés feladathoz hasonlóan azok a tanulók érték el a legjobb eredményeket, akik nem használnak mobil eszközöket, a minimum pont a 4 órás használat körül van. Ezt a pontot követően emelkedő tendencia figyelhető meg, amíg el nem érjük az öt óránál hosszabb használatot, amely deformálja a kvadratikus függvényeket. Az öt óránál hosszabb használat negatív hatással van a tanulók táblázatkezelési ismereteinek és számítógépes gondolkodási készségeinek fejlődésére. Ez az eredmény tovább erősíti a [T2] tézist.

Az adatelemzés feladat esetében három kategóriában szignifikáns különbségeket találtunk (C1: $p = 0,020$, M1: $p = 0,000$, C2: $p = 0,062$, M2: $p = 0,007$) (18. táblázat, 24. ábra). A C1 esetében a legjobb illeszkedés az $f(x) = -0,012x + 0,322$ függvény, amely azt jelenti, hogy minél több időt töltenek a tanulók az iskolában számítógép használatával, annál alacsonyabbak az eredményeik az adatelemzés feladatban, amely igazolja a [T2] tézist.

A mobil eszközök használata (M1 és M2) esetében a pontokra parabolák illeszthetők az $f(x) = 0,006x^2 - 0,053x + 0,363$ ($\min_p = 4,417$, $\min_v = 0,246$) és a $f(x) = 0,0016x^2 - 0,006x + 0,294$ ($\min_p = 3$, $\min_v = 0,285$) függvényekkel. A mobilhasználat az iskolában a legpusztítóbb, abban a tekintetben, hogy azok a diákok érik el a legjobb eredményeket, akik nem használják ezeket az eszközöket. A mobil eszközök napi használata ígéretesebb, mivel a parabola minimumpontja 3 körül van, és ezt követően némileg növekvő tendencia mutatkozik.



24. ábra: A tanulók eredményei az adatelemzés feladatban az eszközökre fordított idő függvényében.

Általánosságban elmondható, hogy a digitális eszközök iskolai használata hatással van a tanulók táblázatkezelési készségeire és kompetenciáira. Fel kell azonban hívni a figyelmet arra, hogy mind a számítógép-, mind a mobilhasználat az iskolában negatív hatással bír, ami riasztó. A napi használat esetén rendkívül gyenge, de pozitív hatást észlelhetők ezen eszközök mérsékelt használata esetén. Összességében azt mondhatjuk, hogy az eszközök használata nem fejleszti hatékonyan a táblázatkezelési ismereteket, mely alátámasztja a [T2] tézist.

Nyilvánvaló, hogy a nulla pontok magas száma deformálja a kapcsolatot a táblázatkezelésben némi tudással rendelkező diákok eredményei és az időkategóriák között. Ennek következtében a nullás eredményeket kihagytuk az elemzésből, és megvizsgáltuk, hogy a táblázatkezelési és az alapvető adatkezelési ismereteket hogyan befolyásolják a digitális eszközök. A képletkiegészítés és a végrehajtási sorrend feladatban nem találtunk szignifikáns különbségeket (19. táblázat).

	képletkiegészítés	végrehajtási sorrend	adattípusok	adatelemzés	tömbképlet
C1	0,289	0,342	0,133	0,131	0,480
M1	0,986	0,053	0,472	0,419	0,371
C2	0,082	0,128	0,048	0,146	0,118
M2	0,331	0,812	0,021	0,000	0,038

19. táblázat: Az eszközökre fordított idő összehasonlítása a tanulók eredményeivel a táblázatkezelési feladatokban, a 0 pontos eredmények kihagyásával.

Ez az eredmény azt jelzi, hogy a képletkiegészítéshez és a műveletek sorrendjének megállapításához szükséges tudás (digitális írástudás [17] szerint) nincs összefüggésben a digitális eszközökkel töltött idővel mely szintén a [T2] tézist bizonyítja. Ez magában foglalja azt a tényt, hogy a tanulók nem használják ezt a tudást az informatikán kívüli órákon vagy egyáltalán nem használják.

Az adattípusok feladat esetében az iskolai eszközhasználat nem befolyásolja az adattípusokhoz kapcsolódó tudást, míg az általános napi használat szignifikáns különbségeket mutat (C2: $p = 0,048$, M2: $p = 0,021$) (19. táblázat). Az általános számítógéphasználat (C2) esetében a regresszióanalízis egy lineáris és egy kvadratikus modellt is talált ($p = 0,016$, illetve $p = 0,041$), az $f(x) = -0,077 + 3,533x$ és $f(x) = -0,006x^2 - 0,021x + 3,489$ függvényekkel. Ezek statisztikailag jobb közelítéseket jelentenek, mint a konstans. Az eredmények azt mutatják, hogy minél több időt töltenek a tanulók számítógépen, annál alacsonyabbak az eredményeik. A mobil eszközök napi használatát figyelembe véve a legjobb illeszkedés egy parabola, amelynek függvénye az $f(x) = 0,019x^2 - 0,175x + 3,561$ ($\min_p = 4,6$, $\min_v = 3,158$). Megállapíthatjuk, hogy azok a tanulók, akik napi négy óránál többet töltenek a mobil eszközök használatával, magasabb eredményeket érnek el az adattípusok felismerésében. Ezek az eredmények bizonyítják a [T2] tézist.

Az adatelemzés és a tömbképlet feladatokban csak a mobil eszközök általános használata esetében voltak szignifikáns különbségek (19. táblázat). Azok a diákok, akik több időt töltöttek mobil eszközökkel, jobb eredményeket értek el. Nem szabad azonban elfelejtenünk, hogy az adatelemzés feladatban a diákok 18%-ának, míg a tömbképlet feladatban 7%-ának van nulla pontnál több pontja ($\max = 3$, $\max = 4$). Mindkét feladat eredményei becsülhetők lineáris és kvadratikus függvényekkel is. Az adatelemzés feladatra a legjobban illeszkedő görbe az $f(x) = 0,013x^2 - 0,074x + 1,533$ ($\min_p = 2,846$, $\min_v = 1,428$), míg a tömbképlet feladatra az $f(x) = 0,074x + 1,845$ és $f(x) = 0,001x^2 + 0,06x + 1,869$ ($\min_p = -30$, $\min_v = 0,969$). Mindkét esetben megállapítható, hogy minél több

időt töltenek a diákok egy nap a mobil eszközökkel, annál jobbák az eredményeik.

Összefoglalva tehát, ha elhagyjuk a nulla pontos eredményeket, akkor az adattípus felismerésében és kódmagyarázási feladatban az általános mobilhasználat pozitív hatással van a diákok tudására. Nem szabad azonban elfelejtenünk, hogy az ezekben a feladatokban nulla pontnál többet elérő tanulók száma rendkívül alacsony, így nem vonhatunk le szilárd következtetéseket.

7. A grafikus információk értelmezése a szövegszerkesztésben

7.1. A mérés bemutatása

Ahogy az korábban említésre került, általában a grafikus felhasználói felületen egyetlen információ – a kurzor pozíciója – elegendő ahhoz, hogy megállapítsuk, hogy egy számozás kézi vagy automatikus. Az egy bit két értéke megegyezik azzal, hogy a kurzor a számtól/karaktertől balra helyezhető-e vagy sem. Annak eldöntésére, hogy mennyi információt kell a csatornára – a tanítási-tanulási folyamatra – helyezni ahhoz, hogy ez a végfelhasználókhöz eljusson, olyan módszert terveztünk és valósítottunk meg, amely magában foglalja a tanítási, tanulási, oktató és tesztelési források elemzését, az interneten vagy zárt csoportokban megosztott Word dokumentumok összegyűjtését és elemzését, a mini-kompetenciateszt számozás feladatának elemzését és entrópiájának kiszámítását. Azért az automatikus számozásra esett a választás, mert ez a digitális szövegek egyik legszélesebb körben használt algoritmusvezérelt jellemzője. Ez egy olyan jellemző, amelynek helytelen használata hibás listákat (megszakított számsorozat, duplikált számozás) és komoly anyagi veszteségeket eredményezhet a dokumentum létrehozásának és módosításának fázisában egyaránt. Továbbá ez az egyik olyan jellemző, amely a mögöttük álló algoritmusok miatt egyértelműen megkülönbözteti a gépirókat a szövegszerkesztőtől/prezentációszerkesztőtől/honlapszerkesztőtől.

A statisztikai elemzést megelőzően egy intuitív mérőrendszert állítottunk fel, amely az automatikus számozás szemantikáján és a GUI-elemein alapul. Ezt az elemzést követően kiszámítottuk az entrópiát és a redundanciát, amelyek objektív mérőrendszert biztosítanak az automatikus számozás tanításához az eszköz pusztá használatán túl. Az entrópia kiszámításához a mini-kompetenciateszt eredményeit használtuk fel. A kiszámított entrópia objektív mérőszámot biztosít a redundancia mérésére, hogy a tanulók/végfelhasználók elérjék a grafikus felhasználói felületen elhelyezett adatokat, ami alapvető fontosságú a grafikus felhasználói felületen alapuló szövegszerkesztés hatékony használatához. Ezzel a módszerrel megtalálhatjuk azokat a kifejezőmódokat, amelyek túlmutatnak az egyszerű eszközhasználaton, de lehetővé teszik a különböző szövegszerkesztési funkciók algoritmusainak feltárását. A teszteredmények és az automatikus számozás szemantikájának kombinációját használtuk az automatikus számozás entrópiájának mérésére.

A grafikus és a számított adatok felhasználásával a grafikus felhasználói felület és a mini-kompetenciateszt alapján fogalmaztuk meg tézisünket. A két érték összehasonlítása megmutatná, hogy a tanítási-tanulási folyamat során mennyi adatot kell a csatornára helyezni ahhoz, hogy a tanulók megértsék a grafikus üzeneteket, valamint hatékonyan és eredményesen használják azokat

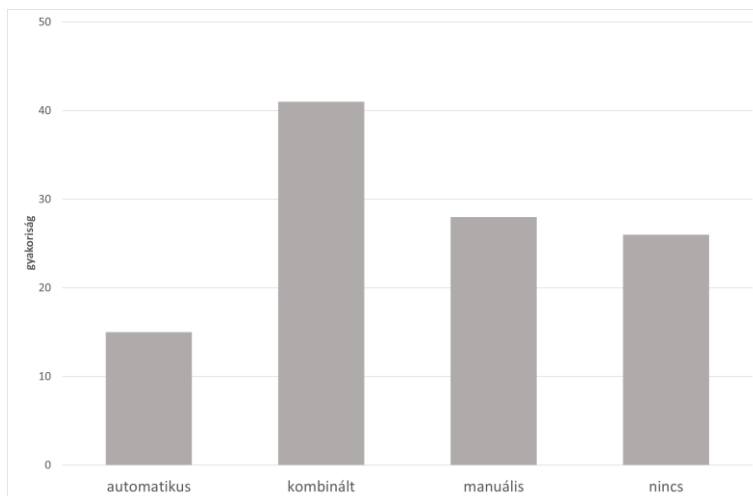
a valós problémák megoldására. Megközelítésünk lényege, hogy a csatornán elhelyezett adatok információtartalmát mérjük.

Az összes létező szövegszerkesztő alkalmazás lefedése messze meghaladja a téma kereteit, ezen felül a nem-nyomtatódó karakterek döntő szerepet játszanak a dokumentumok elemzésében, mivel alapvető információkat hordoznak. Következésképpen megállapítottuk, hogy az elemzéseket olyan szövegfeldolgozóban kell elvégezni, amelyet széles körben használnak, és amelyben a nem-nyomtatódó karakterek megjelenítése egyszerű [147]–[150]. Ezeket a követelményeket figyelembe véve a Microsoft Word funkcióit és dokumentumait elemeztük.

A mérés középpontjában tehát az automatikus számozás áll, beleértve a felsorolásjeleket és a többszintű számozást. Nem szabad megfeledkezni azonban arról a tényről, hogy mind a Wordben, mind más alkalmazásokban ezt a három parancsot három különböző funkcióként (parancsként) kezelik – három gombot kínálnak fel az elérésükhöz, annak ellenére, hogy ez nem így van. A funkciók mögött álló algoritmus ugyanaz, csak a vezető karakterek különböznek. Innentől kezdve mindhárom eszközre számozásként hivatkozhatunk.

Olyan Word dokumentumokat gyűjtöttünk össze, melyek számozott listá(ka)t tartalmaznak, annak érdekében, hogy feltárjuk, hogyan történik a számozás, és a felhasználók mennyire tudatosan alkalmazzák az automatikus számozást. A dokumentumok összegyűjtése és elemzése során kiderült, hogy a számozás népszerű funkció, és a tanítási, tanulási és oktatóanyagok szerint csak egy kattintással elvégezhető egyszerű parancs. 110 különböző nyelvű dokumentumból álló korpuszt elemeztünk, amely az internetről és elsősorban diákok és tanárok által létrehozott magángyűjteményekből gyűjtött dokumentumokból áll. A korpusz dokumentumainak elemzése csak az automatikus számozásra összpontosított. A 25. ábra a számozás használata alapján végzett kategorizálás eredményeit mutatja.

Az elemzett dokumentumok körülbelül háromnegyede tartalmazott számozást. A számozással rendelkező dokumentumok 18%-a használt automatikus számozást, míg a többiben a tisztán kézi számozást vagy az automatikus és a kézi számozás kombinációját (82%) alkalmazták (25. ábra). Ezek az arányok egyértelműen azt mutatják, hogy a számozás sokkal komplexebb, mint ahogy azt a tananyagok állítják, és a végfelhasználóknak több információra van szükségük ennek a funkciónak a helyes használatáról.



25. ábra: A számozás nélküli, a számozást helyesen használó (automatikus), a számozást manuálisan létrehozó (manuális), valamint az automatikus és a kézi megoldást is alkalmazó (kombinált) dokumentumok gyakorisága az összes elemzett dokumentumot figyelembe véve.

A 110 darab Word dokumentum között vannak a szövegszerkesztéshez kapcsolódó tanítási, tanulási, oktató és vizsgaanyagok is. A következő lépésben ezeket a dokumentumokat vizsgáltuk. Ennek a tesztelési fázisnak az volt a célja, hogy feltárjuk, mennyire megfelelően használják ezek a példatárak a számozást. Azt tapasztaltuk, hogy ezeknek jelentős része hibásan alkalmazza a kézi számozást vagy az automatikus és a kézi számozás kombinációját.

7.2. A vizsgálat eredményei

Az eredményekkel célunk volt a [T3] tézis alátámasztása, mely szerint az automatikus számozás szemantikája alapján a szükséges egy bitnél lényegesen több információt kell a tanítás-tanulás csatornájára helyezni ahhoz, hogy a végfelhasználókhöz érthetően eljusson és hatékonyan tudják azt alkalmazni.

A mini-kompetenciafeladat eredményeinek elemzésekor első lépésben meghatároztuk azoknak a tanulónak a számát, akik helyesen válaszoltak a kérdésre. Az eredményeket 20. táblázat mutatja. A kétpontos válaszok alacsony aránya azt jelzi, hogy a tanulók nem tudják, hogy a kurzor helyzetét kell ellenőrizni. A helyes válasz mellett egy további helytelen választ is elfogadtunk és egy ponttal értékeltünk. Ezekben az esetekben további adatok vezethetik félre a tanulókat a válaszuk megjelölésekor.

Pontszám	Mindenki	7. osztály	8. osztály	9. osztály	10. osztály
2	9,7	5,9	9,8	10,7	10,8
1	16,0	14,7	13,6	17,5	16,6
0	74,3	79,4	76,6	71,8	72,6

20. táblázat: A mini-kompetenciateszt számozás feladatában 2, 1 és 0 pontot kapott tanulók százalékos megoszlása.

A legalább egy választ megjelölő tanulók mintáinak feltárása érdekében kétlépcsős klaszterelemzést végeztünk. Az elemzés öt klasztert talált, melyek a 21. táblázat–25. táblázatban bemutatott szabályok alapján kerültek kialakításra.

Megoldások	Gyakoriság	Százalék
00010	309	15,1
00011	301	14,7
00110	29	1,4
00111	8	0,4
01010	451	22,1
01011	373	18,3
01110	20	1,0
01111	18	0,9
10010	265	13,0
10011	102	5,0
10110	30	1,5
10111	3	0,1
11010	51	2,5
11011	27	1,3
11110	6	0,3
11111	48	2,4
Összesen	2041	100,0

21. táblázat: A válaszok kombinációi az 1. klaszterben.

A megoldások oszlop a különböző mintákat sorolja fel, melyekben az 1 a megjelölt válaszlehetőséget, a 0 pedig a nem megjelölt válaszlehetőséget jelenti. A gyakoriság és a százalék oszlop az adott kombinációt választó tanulók számát és százalékos arányát jelzi. Figyelemre méltó, hogy minden klaszterben

van egy domináns válasz, és a további válaszok a domináns válasz köré csoportosulnak. Az 1. klaszterben (21. táblázat) a domináns válasz a D – a szív karakter, amelyet az automatikus számozást imitáló hamis tabulátor követ. Ez a klaszter 16 különböző azonosított mintával rendelkezik, ami azt jelenti, hogy a domináns válasz nem nyújt elegendő információt, és nem játszik vezető szerepet. A leggyakoribb minta a 01010 (B és D), ahol nincs kapcsolat a két válasz között. Két minta tartozik a 2. klaszterbe (22. táblázat). A domináns a 00001 (E), amely a helyes válasz, míg az A és E kombináció alkotja a kisebb csoportot. Ebben a klaszterben a kurzor pozíciója játssza a főszerepet. Továbbá a szám és az elválasztó karakter – Tabulátor vagy semmi – az irányelv.

Megoldások	Gyakoriság	Százalék
00001	822	68,3
10001	382	31,7
Összesen	1204	100,0

22. táblázat: A válaszok kombinációi a 2. klaszterben.

A 3. klaszter (23. táblázat) nyolc különböző mintát tartalmaz, amelyek közül a C a domináns. Ebben a klaszterben van a második legtöbb minta. A leggyakoribb minták a 00100, 10100 és 01100 (C egyedül, C az A-val és C a B-vel).

Megoldások	Gyakoriság	Százalék
00100	268	31,7
00101	48	5,7
01100	150	17,8
01101	62	7,3
10100	157	18,6
10101	53	6,3
11100	55	6,5
11101	52	6,2
Összesen	845	100,0

23. táblázat: A válaszok kombinációi a 3. klaszterben.

A 4. klaszterben (24. táblázat) a domináns minta a 01000 (csak a B), egy szám és elválasztó karakterként egy gépelt szóköz kombinációjával. A második legnépszerűbb minta ebben a klaszterben a 01001 (B az E-vel).

Megoldások	Gyakoriság	Százalék
01000	989	43,4
01001	632	27,7
11000	350	15,4
11001	309	13,6
Összesen	2280	100,0

24. táblázat: A válaszok kombinációi a 4. klaszterben.

Az 5. klaszter (25. táblázat) egy mintát tartalmaz, amely 10000 (A). Az A válaszban egy szám van, elválasztó karakter nélkül. Azok a tanulók, akik ezt a választ választották, csak a számot ismerték fel anélkül, hogy tudták volna, hogy a számot alapértelmezés szerint elválasztó karakter követ.

Megoldások	Gyakoriság	Százalék
10000	939	100,0
Összesen	939	100,0

25. táblázat: A válaszok kombinációi az 5. klaszterben.

A klaszterek domináns válaszait a 26. táblázat mutatja.

Klaszter	Domináns válasz	Mintázat
1	D	???1?
2	E	?0001
3	C	??10?
4	B	?100?
5	A	10000

26. táblázat: A klaszterek domináns válaszai.

7.3. Az automatikus számozás entrópiája

A tesztben bemutatott minták szerint a kurzor pozíciója elegendő a kérdés megválaszolásához. Ezt az információt figyelembe véve a feladat egy IID-moddal írható le. Ebben a modellben mind a helyes, mind a helytelen válaszok valószínűsége $p = 0,5$. Az összes balra pozícionált kurzor helytelen, és az a kurzor helyes, amelynek a bal szélső pozíciója a behúzásnál van. Ez az információ egyetlen bittel kódolható [184]–[192]. Ha azonban a kurzor pozíciója

nem hordozza az információt, akkor a redundáns információk segítenék az ismeretek felidézését a hosszú távú memóriából. Az automatikus számozás jellemzői szerint négy változót találtunk, amely hordozná az automatikus számozás redundáns információit:

- a számozás karakter (szám, betű, speciális karakter) (NC),
- az elválasztó karakter (nincs, szóköz, tabulátor) (SC),
- a bekezdés behúzása,
- a kurzor pozíciója.

A kérdés az volt, hogy az információs entrópia elmélete hogyan támasztja alá tézisünket. Célunk volt egy objektív mérőszámot találni arra vonatkozóan, hogy hány bitre van szükség ahhoz, hogy az információ átmenjen a tanítási-tanulási folyamat csatornáján. Az egyes klaszterek domináns válaszai és az automatikus számozás négy változója alapján a 27. táblázatban szereplő csoportokat azonosítottuk.

Csoport	Mintázat	Gyakoriság	Valószínűség	Azonosítók
A	10000	939	0,13	domináns = 5. klaszter
B	01000	989	0,14	domináns ∈ 4. klaszter
C	00100	268	0,04	domináns ∈ 3. klaszter
D	00010	309	0,04	domináns ∈ 1. klaszter
E	00001	822	0,11	domináns ∈ 2. klaszter
ABE	11001	309	0,04	szám az NC ∈ 4. klaszter
BE	01001	632	0,09	szám az NC és követi SC ∈ 4. klaszter
DE	00011	301	0,04	tabulátor az SC ∈ 1. klaszter
BC	01100	150	0,02	szóköz az SC ∈ 3. klaszter
BCE	01101	62	0,01	szám vagy betű az NC és követi SC ∈ 3. klaszter
AE	10001	382	0,05	szám az NC és nincs SC = 2. klaszter
Egyéb		2146	0,29	

27. táblázat: A klaszterek domináns válaszai és az automatikus számozás három független változója alapján kialakított válaszcsoportok (az NC a számozás karaktert, az SC pedig a szám és a begépett szóveg közötti elválasztó karaktert jelöli).

Az automatikus számozás entrópiájának kiszámításához minden egyes csoport esetében kiszámítottuk a gyakoriságot, majd a valószínűséget (27. táblázat, 1. egyenlet).

$$p_1 + \dots + p_{12} = \sum_{i=1}^{12} p_i = 1 \quad (1)$$

A következő lépésben kiszámítottuk az egyes csoportokra vonatkozó információkat (2. egyenlet).

$$I_k(p) = -\log_2 p_k \quad (2)$$

Utolsó lépésként a csoportok valószínűsége és információi alapján kiszámítottuk az automatikus számozás entrópiáját (3. egyenlet).

$$E(X) = \sum_{i=1}^{12} p_k \cdot I_k = -\sum_{i=1}^{12} p_k \cdot \log_2 p_k = 3.0961 \quad (3)$$

Megállapítható, hogy az entrópia 3,0961 bit. Ez azt jelenti, hogy legalább három bitre van szükség ahhoz, hogy a szövegkezelő felhasználói felületének adatai a végfelhasználóhoz eljuttassuk. Ahhoz, hogy elérjük azt a tudásszintet, amelynél egy bit – a kurzor pozíciója – elég ahhoz, hogy megmondjuk, hogy a számozás automatikus-e, három bit információt kell a csatornára tenni. Ez azt jelenti, hogy a tanítási-tanulási folyamatban nem elég azt mondani, hogy az automatikus számozás nem más, mint egy kattintás az egyik számozási gombra. Továbbá a parancs technikai részletei – az eszköz leírása: hogyan változtatható meg a szín, a forma, a méret stb. – nem tartalmazzák a kézi számozás elkerülése érdekében szükséges információkat. A grafikus felhasználói felület üzeneteit tudatosan kell megtanítani és megtanulni. A végfelhasználóknak el kell sajátítaniuk azt a tudást, amelyet a szövegszerkesztők fejlesztői a GUI-ra helyeztek. Az entrópiával mért redundancia segítene a tanároknak abban, hogy stratégiáikat és módszereiket úgy építsék fel, hogy elegendő információt nyújtsanak a különböző háttérű végfelhasználóknak ahhoz, hogy megértsék az automatikus számozás szemantikáját. Ez az eredmény alátámasztja a [T3] tézist.

Ez azt jelenti, hogy nagy szükség van olyan kiváló tanárookra, akik nemcsak tapasztaltak, hanem tisztában vannak a tanítási-tanulási folyamat lényegével és a digitális korban hatékonyan és eredményesen alkalmazható módszerekkel [33][45][61][106][107].

Az eredményeket figyelembe véve a tanítási-tanulási folyamat hatékonysági arányát (e , 4. egyenlet) és redundanciáját (R , 5. egyenlet) vizsgáltuk meg. Az e érték kiszámításához figyelembe vesszük azt a tényt, hogy minél nagyobb egy véletlen változó entrópiája, annál közelebb van az adott véletlen változó ahhoz, hogy minden kimenetele egyformán legyen valószínű (H_{\max}).

$$e = \frac{H(X)}{H_{\max}} = \frac{3,0961}{3,7004} = 0,8367 \quad (4)$$

$$R = 1 - e = 1 - 0,8367 = 0,1633 \quad (5)$$

Mindezen megállapítások közül a legérdekesebb az, hogy az informatikától és a természetes nyelvektől eltérően a redundancia döntő szerepet játszik a tanítási-tanulási folyamatban [29][32][61][71]. A kérdés az, hogy mi az optimális redundancia mértéke az automatikus számozás tanítási-tanulási folyamatában. Megállapítottuk, hogy az automatikus számozás esetében a redundancia három és négy bit információ között van. Meggyőződésünk azonban, hogy a redundancia mértéke a szövegfeldolgozás különböző jellemzői esetében eltérő. Következésképpen további elemzésekre van szükség, hogy feltárjuk mind az entrópiát, mind a redundanciát más szövegfeldolgozási jellemzők és parancsok esetében. Általánosságban megállapíthatjuk, hogy a teszt eredményei, valamint az automatikus számozás számított entrópiája és redundanciája alapján olyan módszert építettünk fel, amely általánosítható és más szövegszerkesztési jellemzők esetében is alkalmazható lenne. Ezzel az objektív mérési módszerrel képesek vagyunk meghatározni, hogy mennyi információt – a szövegszerkesztő eszközök pusztá használatának tanításán túl – kell a csatornára helyezni ahhoz, hogy a felhasználói felület összes információját átadjuk a végfelhasználóknak.

8. A digitális szövegek entrópiája

8.1. Digitális szövegek entrópiájának elméleti háttere

8.1.1. Kapcsolat a kommunikációs entrópiával

Shannon eredeti tanulmányában [184]–[186] az események valószínűségét elsősorban a relatív gyakoriságukból számították ki. A mi esetünkben kénytelenek voltunk más forrásokat találni a valószínűségek kiszámításához. Úgy találtuk, hogy az események időtartama helyettesítheti a gyakoriságot, és alapul szolgálhat a szöveg-entrópia kiszámításához.

A dokumentumok javítási, formázási és módosítási fázisai digitális folyamatok, és mint ilyenek, az ANLITA segítségével naplózhatók. Az ANLITA (Atomic Natural Language Input Tracking Application) [193] a kutatócsoportunk által kifejlesztett dedikált naplózó alkalmazás, annak rögzítésére, hogy a végfelhasználók milyen műveleteket végeznek a digitális szövegekben. Az ANLITA-t elsősorban az összes billentyűzet- és egérművelet naplózására és a dokumentumokban végzett teljes szerkesztési folyamat videó rögzítésére használjuk. A program két fájlt ad outputként: egy szöveges fájlt, amely tartalmazza az összes billentyűzet és egér eseményt és egy videófájlt, amelyben a teljes szövegkezelési folyamatot rögzítjük. Az ANLITA által rögzített fájlok segítségével kiszámítható az eseményekre fordított idő (atomi lépések). Ezen eseményidők összege adja a feladatra fordított időt (6. egyenlet). Ezen időtartamok és a teljes idő felhasználásával kiszámíthatjuk a feladat valószínűségét (6. és 7. egyenlet) [184]–[192].

8.1.2. Digitális szövegek entrópiájának kiszámítása

A szöveg-entrópia három különböző típusú feladtból származhat: formázás, javítás (hibás dokumentumokban), módosítás. Minden típus esetében ugyanazt a módszert alkalmazzuk a szöveg-entrópia kiszámítására. A feladat algoritmus alapján eseményeket (atomi lépéseket) állítunk fel és minden eseményhez hozzárendelünk egy időt (t_k , amelyet a naplózási folyamatban rögzítünk, és két tizedesjegy pontossággal másodpercben mérünk). Ezen időértékek összege adja a feladat idejét (t) (6. egyenlet).

$$t = t_1 + \dots + t_n = \sum_{k=1}^n t_k \quad (6)$$

Az események időtartama (t_k) és a feladat időtartama (t) alapján kiszámítjuk az egyes események valószínűségét (p_k) (7. egyenlet). Ezeknek a valószínűségeknek az összege 1 (8. egyenlet). Az események valószínűségének kiszámítása a hozzájuk rendelt időből lehetővé teszi, hogy megkülönböztessük a szöveg-entrópiát az információ-entrópiától, miközben továbbra is ragaszkodunk a közös elméleti háttérhez.

$$p_k = \frac{t_k}{t} \quad (7)$$

$$p_1 + \dots + p_n = \sum_{k=1}^n p_k = 1 \quad (8)$$

Ezekből a valószínűségekből (p_k) kiszámítjuk az események információtartalmát (I_k) (9. egyenlet). Shannon [184]–[186] szerint az entrópia leírására bármilyen szám választható a logaritmus alapjaként. Mivel az informatikában a 2-es a legelterjedtebb, úgy döntöttünk, hogy ezt használjuk egy feladat elvégzéséhez szükséges adatmennyiség kiszámításához.

$$I_k(p) = -\log_2 p_k \quad (9)$$

A szöveg-entrópia kiszámításának utolsó lépését az 5. egyenlet mutatja be. A valószínűségek és az információtartalmak szorzatának összege lehetővé teszi számszerűsíteni a feladat-entrópiáját (E). Más szóval, meg tudjuk mondani, hogy a tervezett feladat végrehajtásához hány bit adatot kell a kommunikációs csatornán elhelyezni (10. egyenlet).

$$E(X) = \sum_{k=1}^n p_k \cdot I_k = -\sum_{k=1}^n p_k \cdot \log_2 p_k = \sum_{k=1}^n E_k \quad (10)$$

A szöveg-entrópia egész számként való megadásához egy további lépést kell alkalmazni, ahol E -t felfelé kerekítjük (E_b) (11. egyenlet).

$$E_b = [E(X)] + 1 \quad (11)$$

További kérdés, hogy milyen adatok jöhetnek számításba, amikor a szövegkezelésről van szó. Shannon az általános kommunikációs rendszerében a kommunikációs csatornára kerülő adatokat „információforrásnak” nevezte. A mi esetünkben a helyzet ugyanez. A digitális szövegek kezeléséhez a végfelhasználóknak olyan forrásokkal kell rendelkezniük, amelyek lehetővé teszik számukra a tervezett műveletek végrehajtását. Egy művelet entrópiájának kiszámítása nem határozza meg ezeket a forrásokat. A szövegkezeléshez kapcsolódó ismereteinkhez azonban két forrást biztosan azonosítani tudnak. Ezek az oktatás, valamint a szövegszerkesztő felhasználói felülete, a grafikus felület (GUI).

8.2. A minta

Két különböző tartalmú és hosszúságú dokumentumot elemeztünk, melyek mindegyike mennyiségi és minőségi hibákat tartalmazott. Az egyik dokumentum egy egyoldalas javaslat egy 6. osztályos tanulótól, aki az osztálya jobb iskolai körülményekre vonatkozó kéréseinek ad hangot (röviden: gyógyszerek). Ez a dokumentum csak szöveges elemeket tartalmaz. A másik dokumentum egy tanár háromoldalas tesztlapja (röviden frenchfood), amely szöveges tartalmak, képek, egyéb grafikai objektumok és két táblázat-imitáció kombinációját tartalmazza. Általában véve ezen dokumentumok rugalmasságát vizsgáltuk, azaz azt, hogy a tervezett módosítások mennyire hatékonyan alkalmazhatók rájuk. Következésképpen elsősorban a kvantitatív hibák [33][161][162][194][193][195] voltak a fő szempontjaink. Ezek azonosítása érdekében a nem-nyomtatódó karakterek láthatósága döntő szerepet játszik az elemzésben, ezért minden felvételt és ábrát a Minden látszik gomb bekapcsolásával [147]–[150] készítettünk, és néhányat a dokumentumrészek szöveghatárainak láthatóvá tételével.

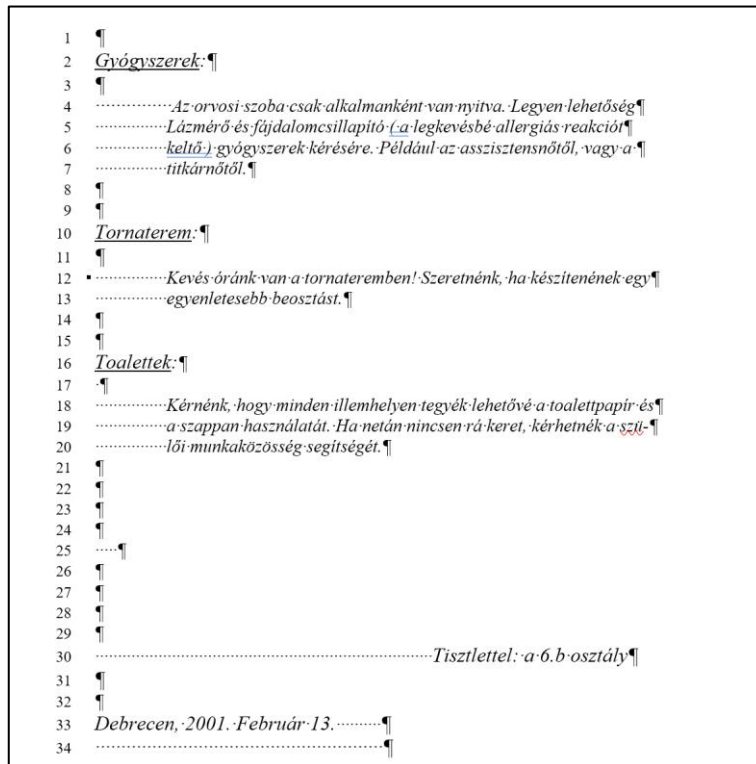
8.2.1. Gyógyszerek dokumentum

A gyógyszerek dokumentum (26. ábra) több tördelési hibát tartalmaz:

- többszörös szóköz és Enter karakterek,
- Enter karakterek a sorok végén (4., 5., 6., 12., 18. és 19. sor).

Ezek a karakterek általában a bekezdések bal oldali behúzását és függőleges távolságot, valamint a 30. sorban a jobb oldali igazítást hivatottak helyettesíteni. Nem tudtuk megmagyarázni a 34. sorban található többszörös szóköz karaktert. A 12. sorban van egy formázási és stílushiba, ami indokolatlan (az 1. címsor stílusa az Együtt a következővel formázással, átformázva, hogy megfeleljen a főszöveg megjelenésének).

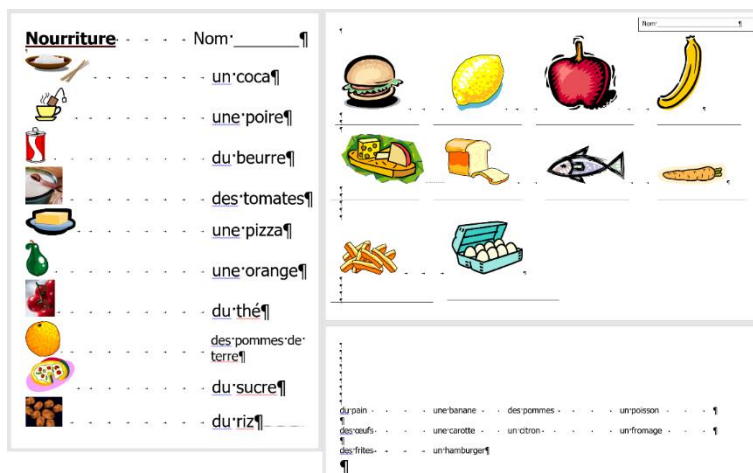
A sorok számozása nem volt része az eredeti dokumentumnak, ez azért látható az ábrán, hogy a hibák helyére kényelmesebb és megbízhatóbb módon lehessen hivatkozni.



26. ábra: Az eredeti egyoldalas gyógyszer dokumentum.

8.2.2. Frenchfood dokumentum

Az eredeti frenchfood dokumentum egy háromoldalas szöveg, amelyet egy tanár készített tesztelési céllal. A dokumentum hossza a Normál stílus beállításaitól függ, de három oldalasnak szánták. Az első oldalon (1. szakasz) álló, míg a második és harmadik oldalon fekvő tájolás (2. szakasz) van beállítva. A dokumentum megterhelt többszörös tabulátor, szóköz és Enter karakterek formájában megjelenő elrendezési hibákkal, a grafikus objektumok (vonalak és szövegdobozok a főoldalon és a hamis fejlécekben egyaránt) elrendezési és formázási hibák keverékével, stílus- és formázási hibákkal a betűtípusokon és bekezdéseken (27. ábra).



27. ábra: Az eredeti háromoldalas francia nyelvű dokumentum (frenchfood).

8.3. A módszer

A kiválasztott dokumentumokat szigorú elemzésnek vetettük alá, hogy feltárjuk a szerkezetüket és a hibákat. Ez az ERM [110] szerint a szövegfeldolgozás első lépése. Az ERM lépéseit követve a második fázisban az eredeti dokumentumok hibáit kijavítottuk, majd a megfelelő formázás következett. Végül mind az eredeti, mind a javított dokumentumokra módosításokat alkalmaztunk a dokumentumok entrópiájának mérésére és összehasonlítására. A részletezett lépéseket a munkacsoport két tagja végezte el. Mindketten a módszer szakemberei, ami segített abban, hogy minden szöveghez megfelelő algoritmust állítsunk fel, és hogy minimalizáljuk az algoritmus lépéseihez rendelt időt. Következésképpen a kevésbé tapasztalt végfelhasználók a lépések megisméltlésekor eltérő időt és valószínűséget mérhetnek. Ezek a különbségek azonban nem feltétlenül rontják az entrópiát, mint annak egy mérőeszközét, amely leírja, hogy mennyire hatékonyan lehet kezelni egy digitális szöveget, mennyire hatékonyan lehet kommunikálni a szövegkezelési folyamat eredményét.

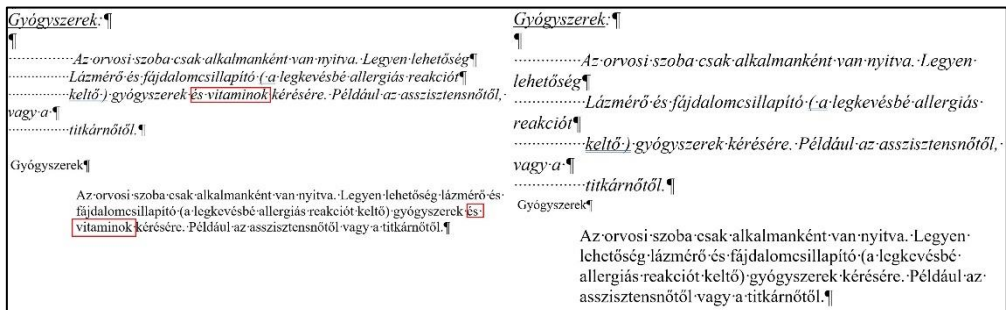
A digitális szövegek tartalmát tekintve minden szöveg egyedi. Az elemzésekből azonban kiderült, hogy az elsődleges probléma a nem-nyomatódó karakterek helytelen használata. Mivel nem voltunk jelen a dokumentumok készítésénél, nem tudjuk, hogy a szerzők ezeket be- vagy kikapcsolt állapotban használták-e [147]–[150]. Függetlenül attól, hogy nincs információnk arra vonatkozóan, hogy a szerzők ismerik-e a nem-nyomatódó karaktereket és tudatosan használják-e ezeket, a vizsgálat során a szövegeken alkalmazott összes folyamatot a bekapcsolt Minen látszik gombbal végeztük és rögzítettük. Ezzel a módszerrel egyértelműen eldönthetjük, hogy karaktereket vagy formázásokat

használt-e a szerző a megjelenés beállításához, amelyek döntő szerepet töltenek be mind az elemzésekben, mind az elemzések dokumentálásában, illetve később az eredményekben is.

Mind a gyógyszerek, mind a frenchfood dokumentumhoz egy-egy algoritmust állítottunk fel, amely alapján a javítást végrehajtottuk. Az algoritmus a dokumentum hibáinak kijavításán túlmenően azt a célt is szolgálja, hogy a kutatásunkban egy alsó korlátot állítsunk fel az egyes módosításokhoz szükséges adatmennyiséghez. Ezzel a módszerrel meghatározhatjuk a javítás-entrópiát és azt, hogy egy szöveg javításához legalább hány bit információra van szükség.

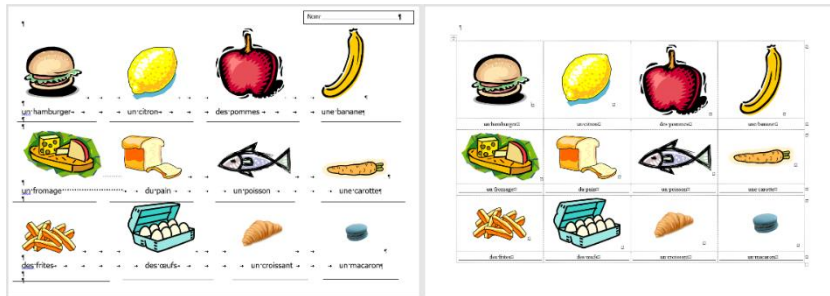
A dokumentumok javítását követően mindkét szöveget megfelelően formáztuk, beleértve a Normál stílus módosítását is. Továbbá, ahol szükséges volt, kisebb kiigazításokat végeztünk. Ez történt a frenchfood dokumentum esetében, ahol a második oldalon lévő néhány képet át kellett méretezni, hogy a második táblázat egy oldalra kerüljön. A javításhoz hasonlóan, az egyes formázások elvégzéséhez szükséges információtartalomhoz alsó korlátot állítunk fel. A javítás és formázás célja a dokumentumok entrópiájának meghatározása és egy olyan objektív mérőrendszer megtalálása volt, amely képes megkülönböztetni a hibás szövegeket a helyesen formázott megfelelőiktől.

A dokumentumok javítási és formázási entrópiájának kiszámításán túl a módosítási entrópiát is ki számítottuk. Ebben az összefüggésben a módosítás azt jelenti, hogy vagy tartalmat adunk hozzá a dokumentumhoz, vagy formázást hajtunk végre. A dokumentumok módosítását a szövegek tartalmának és jellemzőinek megfelelően terveztük meg és hajtottuk végre, melyeket az eredeti és a javított dokumentumokon is elvégeztünk. A gyógyszerek dokumentumokban két módosítást alkalmaztunk. Az első módosításnál a második bekezdéshez (helyes dokumentumban második bekezdés) az „és vitaminok” kifejezést adtuk hozzá, míg a második módosításnál a bekezdés betűméretét változtattuk meg (28. ábra).



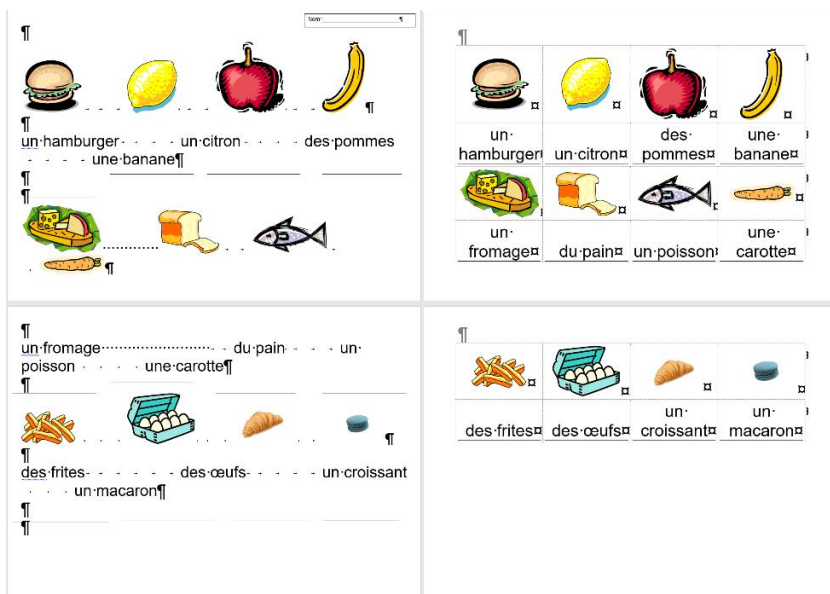
28. ábra: A bal oldali ábrán az „és vitaminok” kifejezés begépelve az eredeti (felső) és a javított (alsó) gyógyszerek dokumentumba. A jobb oldali ábrán a betűméret változása az eredeti (felső) és a javított (alsó) gyógyszerek dokumentumban.

Az eredeti frenchfood dokumentum második oldalán tíz ábra volt három sorba rendezve. Az első két sor négy-négy képből állt, míg a harmadik sor csak két képet tartalmazott (27. ábra). Ennek a hiányosságnak a pótlására a második oldal tartalmát két képpel és két étel nevével egészítettük ki (29. ábra).



29. ábra: Az eredeti (bal) és a javított (jobb) frenchfood dokumentumok második oldala az új élelmiszerek beszúrása után.

A második módosítás a frenchfood dokumentumban a betűméret megváltoztatása volt. A 30. ábra ennek a módosításnak az eredményeit mutatja be mind az eredeti, mind a javított dokumentumban.



30. ábra: A betűméret megváltoztatásának eredményei az eredeti (bal) és a javított, formázott (jobb) frenchfood dokumentumban.

8.4. Eredmények

Az eredményekkel célunk volt a [T4] tézis alátámasztása, mely szerint a természetes nyelvi digitális szövegekben a módosítási feladatok elvégzése több adatot igényel a hibás dokumentumokban, mint azok helyes megfelelőjében.

Egy dokumentum entrópiájának kiszámításához három művelettípust használunk fel:

- első szintű módosítások, amelyek alapvető műveletek elvégzését jelentik (pl. gépelés, betűméret módosítása és kép beillesztése),
- dokumentum javítása, annak érdekében, hogy elkerüljük az alacsony hatékonyságú módosítási műveleteket és
- dokumentum formázása.

Ha a dokumentum helyesen szerkesztett és formázott, a javítás és a formázás kimarad mind a szövegkezelési folyamatból, mind az elemzésből. Továbbá, ha az eredeti dokumentum hibákat tartalmaz, a módosításokat mind az eredeti, mind a javított dokumentumban alkalmazzuk, hogy össze lehessen hasonlítani a dokumentum két formája által hordozott üzeneteket.

A digitális szövegek entrópiájának mérési folyamatában egy további technika az ismétlődő lépések csoportosítása. Ezzel a módszerrel például a többszörös Szóköz, Tab és Enter karakterek kezelése nem karakterenként kerül naplózásra, továbbá teret enged az ilyen ismétlődéseket kezelő különböző megoldások alkalmazására (pl. törlés cserével, blokkokkal vagy dupla kattintással történő kijelölés). Ez a megoldás az algoritmus lépéseit kezelhető méretűre csökkenti, és az ezekhez a lépésekhez hozzárendelt idő másodpercekben mérhető. Az elemzések eredményeit a 28. táblázat–39. táblázat mutatja be, ahol a lépések az algoritmus oszlopban vannak feltüntetve és az egyes lépésekhez tartozó időtartam, az információtartalom és az entrópia értékek a t_k , I_k , illetve E_k mezőkben szerepelnek. A dokumentumok formázását a tesztelési folyamat során csak a dokumentumok helyes változatán végeztük el, hogy elkerüljük a hibák ismétlődését.

8.4.1. Gyógyszerek dokumentum javítása

A 28. táblázat a gyógyszerek dokumentum javításának algoritmusát mutatja be. Megállapítható, hogy a dokumentum javításához legalább két bit adatot (E_b) kell a kommunikációs csatornára helyezni.

Algoritmus	Idő (tk)	I _k	E _k
az összes betűtípus és bekezdésformázás törlése	5,00	5,5262	0,1199
szóköz blokkok törlése	22,93	3,3263	0,3316
bekezdést nyitó szóközök törlése	13,03	4,1431	0,2345
üres bekezdések törlése	25,07	3,1976	0,3485
a dátum az aláírás felett	2,96	6,2765	0,0810
sorvegi Enterek törlése, kötőjelek és bekezdészáró szóközök törlése	27,13	3,0844	0,3637
szintaktikai hibák javítása	133,95	0,7804	0,4543
Összesen	230,07		1,9336

28. táblázat: A gyógyszerek dokumentum javításának lépései.

A gyógyszerek dokumentum összes hibájának javítása körülbelül 230 másodpercet vett igénybe. A többszörös Szóköz és Enter karakterek eltávolítására két módszert rögzítettünk: a blokkok kiválasztása és törlése, a Csere parancs használata, ahol két Szóköz vagy Enter karaktert egy karakterrel helyettesítettünk, és kezeltük a bekezdést nyitó és záró szóköz karaktereket. Ebben a dokumentumban nem volt jelentős különbség a két módszerre fordított idő között.

Gyógyszerek¶

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

Tornaterem¶

Kevés óránk van a tornateremben! Szeretnénk, ha készítenének egy egyenletesebb beosztást.¶

Toailettek¶

Kérnénk, hogy minden illemhelyen tegyék lehetővé a toalettpapír és a szappan használatát. Ha netán nincsen rá keret, kérhetnék a szülői munkaközösség segítségét.¶

Tisztelettel: a 6.b osztály¶

Debrecen, 2001. február 13.¶

31. ábra: A javított gyógyszerek dokumentum.

A gyógyszerek dokumentum a javítás után nyolc bekezdésből áll, és fontos kiemelni, hogy nem tartalmaz felesleges karaktereket (31. ábra). A következő lépésben a javított dokumentum formázására került sor.

8.4.2. Gyógyszerek dokumentum formázása

A formázás algoritmusát, valamint az egyes atomi lépésekre (eseményekre) fordított időt, információtartalmát és entrópiát a 29. táblázat mutatja. A dokumentum formázásához legalább három bit adatot (E_b) kell a kommunikációs csatornára helyezni. Első ránézésre (és eredetileg a szerző szándéka szerint) ez a dokumentum egy könnyen kezelhető apróság. A formázás entrópiája azonban elárulja, hogy a dokumentum formázásához három bit adat szükséges, ami azt jelzi, hogy a feladat elvégzéséhez komoly háttértudásra van szükség. Ennek a „könnyű” feladatnak az entrópiájának kiszámítása megmagyarázza az eredeti dokumentum hibáit. Továbbá bizonyítja, hogy a minimális útmutatás [154] nem elegendő az alapvető szövegszerkesztési ismeretek elsajátításához. Sokkal több adatot kell a csatornára tenni, mint amit a tankönyvek és egyéb oktatási segédanyagok sugallanak [126]–[134].

Algoritmus	Idő (t_k)	I_k	E_k
Normál stílus módosítása	18,10	2,5217	0,4391
balra behúzás és másolás	15,89	2,7096	0,4142
címek: betűméret, térköz és formátummá- solás	32,12	1,6942	0,5236
dátum: térköz	14,00	2,8923	0,3896
aláírás: jobbra igazítás	10,88	3,2560	0,3408
kötőjeles írásmód	12,95	3,0047	0,3744
Összesen	103,94		2,4817

29. táblázat: A gyógyszerek dokumentum formázásának lépései.

8.4.3. Gyógyszerek dokumentum módosítása

A gyógyszerek dokumentum módosítása négy szakaszból áll: két szó beillesztése az eredeti (30. táblázat) és a javított (31. táblázat) dokumentumokba (28. ábra), egy bekezdés betűméretének növelése az eredeti (32. táblázat) és a javított (33. táblázat) dokumentumokban (28. ábra).

Algoritmus	Idő (t _k)	I _k	E _k
a kurzor pozicionálása	5,09	2,6941	0,4163
szavak beírása	6,86	2,2636	0,4714
Enter beírása	5,12	2,6856	0,4174
szóköz beírása	7,01	2,2324	0,4751
Enter törlése	2,86	3,5258	0,3061
szóköz törlése	6,00	2,4568	0,4475
Összesen	32,94		2,5338

30. táblázat: Két szó beillesztésének lépései az eredeti (hibás) gyógyszerek dokumentumba.

Algoritmus	Idő (t _k)	I _k	E _k
a kurzor pozicionálása	5,09	1,2313	0,5245
szavak beírása	6,86	0,8007	0,4597
Összesen	11,95		0,9841

31. táblázat: Két szó beillesztésének lépései a javított gyógyszerek dokumentumba.

A hibás szövegben végzett műveletek további lépéseket igényelnek az eredetihez hasonló megjelenés visszaállításához, a baloldali behúzás imitálásához. Ezek a további lépések a Szóköz és Enter karakterek beírása és törlése. Azonban, meg kell említenünk ennek a megoldásnak egy további hatását: minden alkalommal, amikor a szöveg tartalma megváltozik, meg kell ismételni a szöveg eredeti megjelenéshez való igazítását, ami rendkívül idő- és erőforrás-igényes folyamat. A két szó beillesztése az eredeti dokumentumban 3 bit (30. táblázat), a javított dokumentumban 1 bit adatot igényel (31. táblázat). Utóbbi esetben pusztán a végfelhasználó szándékát (a gépelést) kell elvégezni, semmilyen járulékos tevékenységet nem igényel. Ez az eredmény alátámasztja a [T4] tézist. A módosítások elvégzéséhez, azt megelőzően tehát az egyetlen hatékony megoldás a hibák kijavítása.

Algoritmus	Idő (t_k)	I_k	E_k
kijelölés	1,89	6,0947	0,0892
betűméret megváltoztatása	3,03	4,0582	0,2436
Enter beírása	1,17	4,7459	0,1769
szóköz beírása	4,80	4,2997	0,2183
Enter törlése	2,98	2,3086	0,4660
szóköz törlése	4,06	4,3763	0,2107
Enter beírása	16,14	3,7267	0,2815
szóköz beírása	3,85	3,9878	0,2514
Enter törlése	6,04	3,1481	0,3551
szóköz törlése	5,04	3,0051	0,3743
Enter beírása	9,02	3,3284	0,3313
szóköz beírása	9,96	4,3140	0,2169
Enter törlése	7,96	5,4028	0,1277
szóköz törlése	4,02	4,7219	0,1789
Összesen	79,96		3,5219

32. táblázat: A betűméret megváltoztatásának lépései az eredeti (hibás) gyógyszerek dokumentumban.

Algoritmus	Idő (t_k)	I_k	E_k
a kurzor pozicionálása	1,89	1,3803	0,5302
szavak beírása	3,03	0,6993	0,4307
Összesen	4,92		0,9609

33. táblázat: A betűméret megváltoztatásának lépései a javított gyógyszerek dokumentumban.

A betűméret módosítása az eredeti dokumentumban 4 bit (32. táblázat), a javított dokumentumban csak 1 bit adatot igényel (33. táblázat). Ez azzal magyarázható, hogy a felhasználónak mellékes műveleteket kell végeznie az imitált baloldali behúzáshoz. Ezek a műveletek túlmutatnak a végfelhasználó módosítási szándékán (amely egyszerűen csak a betűméret megváltoztatása volt), és az entrópia alapján rendkívül sok információt igényel. A kijavított dokumentumban a kiválasztott bekezdés betűméretének megváltoztatása kevesebb mint 5 másodpercet vett igénybe, és egy bit adatot kell a csatornára helyezni. Ez az eredmény szintén a [T4] tézist támasztja alá.

A két módosítás entrópiájának összehasonlítása azt mutatja, hogy egy hibás dokumentum kezelése sokkal több információt követel, mint egy helyes dokumentumé. Ebben az esetben két szó begépelése majdnem háromszor több időt

igényelt, és körülbelül háromszor több adatot kellett a csatornára helyezni a hibás dokumentumban. Megállapíthatjuk, hogy ha a végfelhasználók hatékonyan akarnak dolgozni egy digitális szövegben, akkor a tartalmat először helyesen kell szerkeszteni és formázni. Minden más esetben a szövegszerkesztési tevékenység veszteséges, időt, pénzt és erőforrásokat veszítünk.

8.4.4. Frenchfood dokumentum javítása

A frenchfood dokumentum két táblázatot tartalmaz, az egyiket az 1. oldalon, a másikat a 2. oldalon. Ahelyett, hogy a szerző két valódi táblázatot használna, az eredeti dokumentum tele van a két táblázat imitálására tett kísérletekkel. A dokumentum nagyszámú tördelési hibája miatt a javítás több mint 380 másodpercig tartott, és legalább négy bit adat (E_b) szükséges hozzá (34. táblázat). Ez azt mutatja, hogy a dokumentum hibáinak felismerése meglehetősen igényes, és a tanítási-tanulási folyamat során tudatosabban kellene kezelni, elsősorban azért, hogy elkerüljük ezeket a hibákat, másodsorban pedig azért, hogy még a dokumentumon történő módosítás előtt ki tudjuk őket javítani.

Algoritmus	Idő (t_k)	I_k	E_k
minden betűtípus és bekezdésformázás törlése	6,14	5,9627	0,0956
az ellenőrzés nyelvének megváltoztatása franciára	16,91	4,5011	0,1988
szövegdoboz törlése az 1. oldalon és a szöveg áthelyezése	36,09	3,4074	0,3211
a 2. oldalon a szövegdoboz törlése	19,02	4,3315	0,2151
sorok törlése a 2. oldalon	32,85	3,5431	0,3040
a papírméret ellenőrzése	51,03	2,9076	0,3875
a margók alapértelmezettre állítása	14,98	4,6759	0,1829
több lap törlése	30,95	3,6290	0,2933
a 2. oldalon a szóköz karakterek Tabulátorral való helyettesítése	1,03	8,5383	0,0230
az 1. oldal első bekezdésének áthelyezése a fejlécbe	7,15	5,7430	0,1072
aláhúzás karakterek törlése	22,94	4,0611	0,2433
bekezdést záró Tabulátorok törlése	42,02	3,1879	0,3498
táblázat létrehozása az első oldalon (T1)	18,81	4,3475	0,2136
T1 formázása	35,09	3,4479	0,3160
táblázat létrehozása a 2. oldalon (T2)	12,92	4,8894	0,1650
T2 formázása	34,99	3,4520	0,3154
Összesen	382,92		3,7316

34. táblázat: A hibás frenchfood dokumentum javításának lépései.

8.4.5. Frenchfood dokumentum formázása

A javított dokumentum formázása körülbelül 100 másodpercet vett igénybe, és három bit adatot kell a csatornára helyezni (35. táblázat). A dokumentumon az alábbi formázási lépéseket kell végrehajtani:

- alapvető betűtípus- és bekezdésformázások,
- táblázat szegélyezése,
- a fejlécben szöveg pozicionálása tabulátorokkal,
- fejléc formázása.

Algoritmus	Idő (t _k)	I _k	E _k
betűméret beállítása T1-ben	8,93	3,4856	0,3112
az igazítás beállítása a T1-ben	3,00	5,0593	0,1517
vonalak (szegély) hozzáadása a T2-ben	11,11	3,1705	0,3521
térköz hozzáadása a T2-ben	22,84	2,1308	0,4865
az igazítás beállítása a T2-ben	13,16	2,9262	0,3850
a T2-ben lévő szegélyek (üres vonalak)	40,99	1,2871	0,5274
Fejléc formázása pozícionált tabulátorokkal, fejléc beállítása csak az 1. oldalon	38,93	1,3615	0,5299
Összesen	100,03		2,7438

35. táblázat: A frenchfood dokumentum formázásának lépései.

8.4.6. Frenchfood dokumentum módosítása

A dokumentum módosítási fázisában a 2. oldalon a táblázat üres celláiba két étel képe és azok neve került be (29. ábra). A módosítást mind az eredeti (hibás), mind a javított és formázott dokumentumokban elvégeztük. A folyamatok lépéseit a 36. táblázat–37. táblázat mutatja be.

Algoritmus	Idő (t _k)	I _k	E _k
új bekezdések beillesztése	9,05	5,4766	0,1230
szavak mozgatása	99,84	2,0130	0,4987
igazítás Tabulátorokkal/szóközökkel	65,18	2,6282	0,4251
nem-nyomtatódókarakterek beállítása	25,05	4,0078	0,2491
karakterek törlése a dokumentum végéről	3,84	6,7134	0,0640
képek hozzáadása	31,95	3,6568	0,2899
a képméret beállítása úgy, hogy a dokumentum csak két oldalas legyen	60,07	2,7460	0,4093
élelmiszerek nevének hozzáadása	33,84	3,5739	0,3001
szóközökkel és tabulátorokkal való igazítás	74,15	2,4422	0,4494
Összesen	402,97		2,8087

36. táblázat: A két kép és név beillesztésének lépései az eredeti frenchfood dokumentumba.

Megállapítást nyert, hogy az eredeti dokumentum módosítása nagyobb igénybevételt jelent, mint a helyes dokumentumé. Több mint kétszer annyi időt és tudáselemet igényelt a hibás dokumentum, mint a javított. Mindkét esetben

be kellett illeszteni a képeket és gépelni kellett a szavakat, ami körülbelül ugyanannyi időt vett igénybe. A képek méretének beállítása mindkét dokumentumban hasonló időt igényelt. Az eredeti dokumentumban azonban további lépésekre volt szükség a képek és a szavak (egymás alatt és vízszintesen középre állítva) beállításához. Ezek a lépések időt vettek igénybe.

Algoritmus	Idő (t _k)	I _k	E _k
szavak áthelyezése a T2 üres celláiba	96,96	0,9319	0,4885
karakterek törlése a dokumentum végéről	4,04	5,5169	0,1205
képek beillesztése a T2 üres celláiba	34,83	2,4090	0,4536
a T2 üres celláiba ételek nevének hozzáadása	20,13	3,2000	0,3482
a képméret beállítása úgy, hogy a dokumentum csak két oldalas legyen	29,02	2,6722	0,4192
Összesen	184,98		1,8300

37. táblázat: A két kép és név beillesztésének lépései a javított frenchfood dokumentumba.

Az eredeti és a javított dokumentumok Normál stílusának megváltoztatása a betűtípussal (Arial) és a betűmérettel (32 pt) figyelemre méltó különbségeket mutat.

Algoritmus	Idő (t _k)	I _k	E _k
Normál stílus módosítása	16,00	4,2287	0,2255
formázás törlése	6,96	5,4296	0,1260
tabulátorok törlése az 1. oldalon	5,05	5,8924	0,0992
1. szövegdoz módosítása	22,00	3,7693	0,2764
2. szövegdoz módosítása	16,05	4,2242	0,2260
tabulátorok törlése a 2. oldalon	13,02	4,5261	0,1964
sorok elrendezése	51,82	2,5333	0,4376
tabulátorok és szóközök törlése	30,14	3,3151	0,3331
sorok elrendezése	69,96	2,1003	0,4898
tabulátorok törlése	31,00	3,2745	0,3384
sorok elrendezése	37,98	2,9816	0,3775
Összesen	299,98		3,1260

38. táblázat: A betűméret megváltoztatásának lépései az eredeti (hibás) frenchfood dokumentumban.

Míg az eredeti dokumentumban ez körülbelül 300 másodpercet vett igénybe (38. táblázat), addig a javított dokumentumban csak körülbelül 20 másodpercre volt szükség (39. táblázat).

Algoritmus	Idő (t_k)	I_k	E_k
Normál stílus változása	12,06	0,6615	0,4182
egyértelmű formázás	7,02	1,4431	0,5307
Összesen	19,08		0,9489

39. táblázat: A betűméret megváltoztatásának lépései a javított frenchfood dokumentumban.

Az is figyelemre méltó, hogy a betűméret megváltoztatása, majd a képek és a szöveg beállítása négy bit adatot igényel (38. táblázat), míg a javított dokumentumban egy bitre van szükség (39. táblázat). Általánosságban megállapíthatjuk, hogy a hibás dokumentumok kezelése nehezebb, bonyolultabb és igényesebb, mint a helyes dokumentumoké. Ez az eredmény tovább erősíti a [T4] tézist.

9. Dokumentumok fenntarthatósági aránya

9.1. Elméleti háttér

Ahogy az előző fejezetben is bemutatásra került, a hibás és a helyes dokumentumok entrópiájának méréséhez és összehasonlításához a szövegkezelési folyamatokat az ANLITA alkalmazással rögzítettük [193]. A felvételek által szolgáltatott nyers adatokból kiindulva meg tudjuk határozni, hogy a felhasználó mennyi időt töltött az egyes atomi lépések végrehajtásával. A lépések számából és a hozzájuk rendelt időből a Shannon-féle információelmélet alapján kiszámítható az egyes feladatok entrópiája [184]–[186] és a digitális szöveg entrópiája [160]. Miután kiszámítottuk az entrópiát, meg tudjuk mondani, hogy hány bitnyi adat (entrópia felfelé kerekítve) (E_b) szükséges az adott feladat elvégzéséhez, azaz hány bit adatot kell a kommunikációs csatornára tenni. A feladatra fordított idő és a feladat elvégzéséhez szükséges adatmennyiség alapján számítható ki a feladat fenntarthatósági aránya (1. definíció). Ezekkel az értékekkel meghatározhatjuk a hanyag/helytelen szövegkezeléshez kapcsolódó veszteség mértékét, figyelembe véve mind az emberi, mind a számítógépes erőforrásokat.

1. definíció Jelölje a helyes (correct) dokumentum módosítására fordított időt tm_C , a hibás (erroneous) dokumentum módosítására fordított időt pedig tm_E . Ekkor az időhányadosot (q_{tm}) a két érték hányadosaként határozhatjuk meg (12. egyenlet).

$$q_{tm} = \frac{tm_E}{tm_C} \quad (12)$$

2. definíció Jelölje a helyes dokumentumban bekövetkezett módosítás entrópiáját em_C , a hibás dokumentumban bekövetkezett módosítás entrópiáját pedig em_E . Ekkor az entrópia-hányados (q_{em}) a következőképp számítható ki (13. egyenlet).

$$q_{em} = \frac{2^{em_E}}{2^{em_C}} = 2^{em_E - em_C} \quad (13)$$

3. definíció A fenntarthatósági arány (S_m) a következőképpen határozható meg (14. egyenlet).

$$S_m = \frac{1}{q_{tm} \cdot q_{em}} \quad (14)$$

1. tétel: Tegyük fel, hogy q_{tm} és q_{em} nem csökkenő. Ekkor

$$\lim_{q_{tm} \rightarrow \infty} S_m = 0 \quad \text{és} \quad \lim_{q_{em} \rightarrow \infty} S_m = 0$$

Bizonyítás: Mivel q_{tm} és q_{em} nem csökkenő, ezért

$$\lim_{q_{tm} \rightarrow \infty} q_{tm} = \infty \quad \text{és} \quad \lim_{q_{em} \rightarrow \infty} q_{em} = \infty$$

Egy helyes dokumentumban a módosítási feladat fenntarthatósági aránya 1 (S_m), mivel mind q_{tm} , mind q_{em} egyenlő 1-gyel. Ebből arra következtetünk, hogy minél közelebb van a fenntarthatósági arány az 1-hez, annál kevésbé hibás egy dokumentum. Az előző fejezet alapján tudjuk, hogy a javítási feladatok entrópiája egy kiszámítható érték [196]. A kérdés az, hogy hogyan értékeljük a javításokat, amikor a fenntarthatósági arányt számítjuk.

4. definíció A hibás dokumentumokból származó javítási feladat fenntarthatósági aránya definíció szerint 0.

A javítási feladat fenntarthatósági aránya 0 kell, hogy legyen, mert a javítás idő- és erőforráspazarlás, és elkerülhetőek a megfelelően strukturált és formázott szövegek létrehozásával [110][160]–[162].

5. definíció Több módosítás esetén a kumulatív fenntarthatósági arányt (jelölve S -sel) az egyes módosítások fenntarthatósági arányainak összege és n hányadosaként határozzuk meg, ahol n a módosítások számát jelöli (15. egyenlet).

$$S = \frac{\sum_{i=1}^n S_{m_i}}{n} \quad (15)$$

9.2. A minta

A fenntarthatósági arány kiszámításához egy 213 oldalas digitális jelentést választottunk, amely egy számozott listákkal túlsúlyolt MS Word dokumentum. A dokumentumban található hibák hossza és jellege miatt szelektíven csak a fejezetek számozására, a bekezdések számozására és a módosítások következményeire (beleértve a tartalomjegyzéket és a kereszthivatkozásokat) összpontosítottunk. A dokumentum egyéb listákat is tartalmaz, például lábjegyzeteket, ábrákat és táblázatokat, melyeket nem vettünk figyelembe a mérés során.

Az ERM-mel összhangban [110] egy tanítási-tanulási folyamatban a dokumentum kezelésének első fázisa az elemzés, ezt követi a javítás, majd a megfelelő formázás, végül az ellenőrzés [111][86]. A hibás és a helyes dokumentumok összehasonlításához, valamint a két dokumentumra alkalmazott feladatok entrópiájának kiszámításához [160][185][186] azonban a kutatási projekt során egy további lépést hajtottunk végre, amelyben első szintű módosításokat alkalmaztunk a szövegeken, ahol gépelésre és másolásra került sor [197].

9.3. A módszer

A bekezdések számozása a kiválasztott dokumentum egyik furcsa jellemzője. A főszöveg minden bekezdése számozott. Ez a fejezetcímek és alcímek esetében ésszerű, de a tartalmi bekezdések esetében nehezen indokolható vagy magyarázható. Az elemzés során felfedeztük, hogy a számozás nagy része kézi számozás, valamint függetlenül attól, hogy kézi vagy automatikus, csak első szintű számozást használtak a szerzők. A megfelelő többszintű számozás létrehozásához először az összes eredeti számozást törölni kellett. Azt is meg kell jegyeznünk, hogy az eredeti dokumentum 143 különböző stílust alkalmazott, amire nem találtunk ésszerű magyarázatot. Ezeket a stílusokat töröltük és a Normál stílust újra definiáltuk, mielőtt a teljes dokumentumra alkalmaztuk volna. Az automatikus listákat a Normál stílusnak a teljes dokumentumra történő alkalmazásával töröltük, míg a kézi számozást a beépített Csere parancs segítségével korrigáltuk. A cserék sorozata előtt egy második körös alapos elemzést végeztünk, hogy feltárjuk a dokumentumban található kézi számozás különböző típusait. Különböző mintákat azonosítottunk, amelyek a számjegyek és az azokat közvetlenül követő szimbólumok száma alapján írhatók le (40. táblázat). A táblázatban a t_k a k . eseményre (atomi lépésre) fordított időt, I_k a k . esemény információtartalmát, E_k pedig a k . esemény entrópiáját jelöli. Továbbá $\wedge\#$ egy számjegyet, $\wedge\$$ egy betűt, \wedge^t pedig egy tabulátort jelöl.

Algoritmus	Idő (tk)	I _k	E _k
^#^#^#.^t	36,01	2,3862	0,4564
^#^#.^t	11,07	4,0879	0,2404
^#.^t	9,11	4,3691	0,2114
^#^#^#^t	20,00	3,2346	0,3436
^#^#^t	7,02	4,7450	0,1769
^#^t	7,99	4,5583	0,1935
^\$^\$^\$.^t	40,89	2,2028	0,4785
^\$^\$.^t	12,12	3,9572	0,2548
^\$.^t	8,93	4,3978	0,2086
^\$. ^t	7,00	4,7492	0,1766
^#)^t	28,11	2,7435	0,4097
Összesen	188,15		3,1505

40. táblázat: A dokumentum bekezdéseinek számozásán végzett javítás algoritmusai és a szekvenciák a dokumentumban a kézi számozás helyettesítésére.

Az összes kézi számozás törlése az eredeti dokumentumból 188,15 másodpercet vett igénybe, 3,1505 entrópia mellett (40. táblázat). Mindezek a javítási lépések és az elemzésre fordított idő veszteség, amit egy megfelelően megtervezett, strukturált és formázott szöveggel el lehetett volna kerülni [33][162][196].

Az elemzés azt is feltárta, hogy mind a tartalomjegyzéket, mind a kereszt-hivatkozásokat kézzel illesztették be. Következésképpen mindegyiket meg kellett találni és újrászámolni, amikor a fejezetek számozásában változás történt. A megfelelően strukturált és formázott dokumentum létrehozása érdekében ezeket a hibákat kijavítottuk és automatikus kereszt-hivatkozásokat illesztettünk be. A teljes dokumentum javításának részletei terjedelmük miatt nem kerülnek bemutatásra, azonban a módosítási feladatok fenntarthatósági arányának kiszámításához ezt a helyes dokumentumot használtuk az eredeti (hibás) dokumentummal történő összehasonlításra. A teljes dokumentum javítását munkacsoportunk két tagja végezte és megközelítőleg 67 órát igényelt, így összességében 2-szer 67 órára volt szükség.

Annak demonstrálására, hogy mennyire volt problémás az eredeti dokumentum, és mennyire kiszámíthatóvá vált a helyes, mindkét dokumentumon ugyanazokat a módosításokat végeztük el.

- Az eredeti 1. fejezet elé egy új 1. fejezetet illesztettünk be. Ezzel a feladattal demonstrálhattuk és mérhettük, hogy egy ilyen módosítás

hogyan hat a fejezetek számozására, a tartalomjegyzékre és a fejezetekre való keresztivatkozásokra.

- Az első tartalmi bekezdést bemásoltuk és beillesztettük az eredeti 1. bekezdés elé. Ezzel a másolással bizonyítani tudtuk a kézi átszámozás hátrányait és az automatikus számozás hatékonyságát.

A módosítási feladatokat, valamint az entrópia és a fenntarthatósági arányok kiszámítását a következő sorrendben hajtottuk végre.

- Minden egyes módosítási feladathoz egy algoritmust állítottunk fel.
- Az algoritmus alapján az ANLITA által naplózott adatokból azonosítottuk az atomi lépésekre fordított időt (tk) [193][196].
- Az atomi lépések felhasználásával kiszámítottuk az egyes lépések valószínűségét, információtartalmát (Ik) és entrópiáját (Ek). Az Ek értékek összege adja az entrópiát (E), a kerekített entrópia pedig megadja, hogy hány bitet (Eb) kell a kommunikációs csatornára tenni az adott feladat megoldásához. Az adatok elsődleges forrásai az oktatás (képzés) és a szövegszerkesztő felhasználói felülete (GUI).
- Végül a fenntarthatósági arányt az eredeti és a helyes dokumentum módosítási feladatainak összehasonlításából számítottuk ki.

9.4. A fenntarthatósági arány alkalmazása

A fenntarthatósági arány alkalmazásával célunk volt alátámasztani a [T5] tézist, mely szerint a fenntarthatósági arány – a $[0; 1]$ intervallumon értelmezve – hibás dokumentumokban alacsonyabb, mint ezek helyes megfelelőjében.

9.4.1. Fejezetek számozásának megváltoztatása

Az eredeti dokumentum elemzése feltárta, hogy a tartalomjegyzék és a keresztivatkozások kézzel készültek, a tartalomjegyzék nem létező oldalszámokat tartalmaz, és a dokumentum nem képes automatikusan alkalmazkodni a fejezetek számozásának változásaihoz, nem frissül (32. ábra).

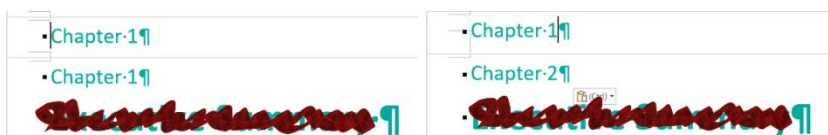


Chapter 8:	→	207
Chapter 9:	→	225
	→	230
	→	236

32. ábra: Kézi számozás az eredeti dokumentum tartalomjegyzékében.

A képernyőképen látható, hogy a fejezetek oldalszámait nem frissítették, ami helytelen és nem létező számozáshoz vezetett. Másrészt, ha a fejezetek felsorolása és stílusa automatikusan történik, valamint a tartalomjegyzék és a kereszthivatkozások a meglévő automatikus számozás és stílus alapján készülnek, akkor a szövegszerkesztő képes nyomon követni a változásokat, és ennek megfelelően módosítani a dokumentumot.

A következő mintákon a kurzort a sor első lehetséges pozíciójára helyeztük, hogy szemléltessük, hogy az eredeti hibás dokumentumban a Chapter szó és a hozzá tartozó számozás tartalomként került be (33. ábra, bal oldali minta), míg a helyes dokumentumban egy formázást alkalmaztunk (33. ábra, jobb oldali minta).



33. ábra: Egy új 1. fejezet beillesztése az eredeti dokumentumba kézi számozással (bal) és a helyes dokumentumba automatikus számozással (jobb). A kurzor a legelső pozícióba kerül.

A 41. táblázat összefoglalja, hogy mennyi időre van szükség egy új fejezet beillesztéséhez a helyesen szerkesztett dokumentumban. Az eredeti dokumentumban 16-szor több időre volt szükségünk egy új Chapter 1 fejezetcím beillesztéséhez, mint a helyes dokumentumban.

Algoritmus	Idő (t_k)	I_k	E_k
kurzor pozíciója	3,96	4,0172	0,2481
új bekezdés beillesztése Normál stílusban	1,05	5,9323	0,0971
kiválasztás	2,09	4,9392	0,1610
másolat formázás	5,87	3,4493	0,3158
frissítés	20,18	1,6678	0,5249
tartalomjegyzés frissítése	30,97	1,0499	0,5071
Összesen	64,12		1,8540

41. táblázat: Egy új 1. fejezet beillesztése a helyesen szerkesztett dokumentumba és annak időigénye.

Egy olyan feladat, amely körülbelül 1 perc alatt elvégezhető, a hibás dokumentumban több mint negyedórát igényel, anélkül, hogy biztosítva lenne, hogy minden szükséges frissítés megtörtént. A hibás dokumentumban nincs garancia arra, hogy a dokumentum minősége eléri a magas színvonalat.

A Chapter1 módosításának algoritmusait mind az eredeti (hibás), mind a helyes dokumentumokra felállítottuk. A helyes dokumentumban 6 atomi lépés azonosítható, az ezekre a lépésekre fordított idő 64,12 másodperc, az entrópia pedig 1,8540 (41. táblázat). Az eredeti (hibás) dokumentumban, ahol a frissítést kézzel kellett elvégezni, ugyanennek a módosításnak a végrehajtásához 63 lépésre és 1048 másodpercre volt szükség, az entrópia 4,0874. Megállapíthatjuk, hogy a módosítás elvégzéséhez a helyes és az eredeti dokumentumban 2, illetve 5 bit adatot (E_b) kell a kommunikációs csatornán elhelyezni.

Összességében megállapíthatjuk, hogy a módosítási feladatok több adatot (nagyobb entrópiát) igényelnek a hibás dokumentumokban, mint a helyes dokumentumokban, amely a [T4] tézist igazolja.

Az eredeti (hibás) dokumentumban ezen módosítás (új fejezetcím beillesztése) fenntarthatósági arányának kiszámításához a rögzített időt és a számított entrópiát használtuk. A 9.1 fejezetben bemutatott módszer és képletek alapján 0,0078 fenntarthatósági arányt kaptunk, ami rendkívül közel van a 0-hoz, ez részben igazolja a [T5] tézist.

9.4.2. Számozott bekezdések beillesztése

A következő fázisban egy új számozott bekezdést illesztettünk be a szövegbe, közvetlenül az első bekezdés elé. A 34. ábra a kurzort mutatja az első lehetséges pozícióban, az eredeti és a helyes dokumentumban, a bekezdés másolása előtt és után.

1. → The- secu any: A'su our- sper by'tl	1. → The- secu any: A'su our- sper by'tl	1. → The sect any den traf mor the-	1. → The sect any den traf mor the-
2. → For- regu TV,:' com sign othe deli the- cop billic	1. → The- secu any: A'su our- sper by'tl 2. → For- regu TV,:' com	2. → For- tion sou: corr sign oth deli the- the- cop take	2. → The sect any den traf mor the- 3. → For- tion sou corr
3. → The- pace com new Net	com sign othe deli the- the- the- the-	3. → The pac cati atio (ISD)	3. → The corr sign oth deli the-

34. ábra: Az eredeti és a módosított kézi számozás (1. és 2. oszlop), valamint az eredeti és a módosított automatikus számozás (3. és 4. oszlop). A kurzor a legelső pozícióba kerül.

A helyes dokumentumban az első bekezdés másolása 4 atomi lépést igényelt, amelyet 6 másodperc alatt 1,9265 entrópiával ($E_b = 2$) tudunk elvégezni.

Ezeket a lépéseket a 42. táblázat mutatja be. Ugyanennek a feladatnak az elvégzéséhez a hibás szövegben 150 atomi lépésre volt szükségünk, ami 714 másodpercet vett igénybe 7,1244 entrópiával ($E_b = 8$). Ez az eredmény szintén a [T4] tézist támasztja alá.

Algoritmus	Idő (t_k)	I_k	E_k
cím kiválasztása	1,88	1,6694	0,5248
másolás a vágólapra	2,05	1,5445	0,5295
kurzor pozíciója	1,01	2,5658	0,4334
beillesztés	1,04	2,5236	0,4389
Összesen	5,98		1,9265

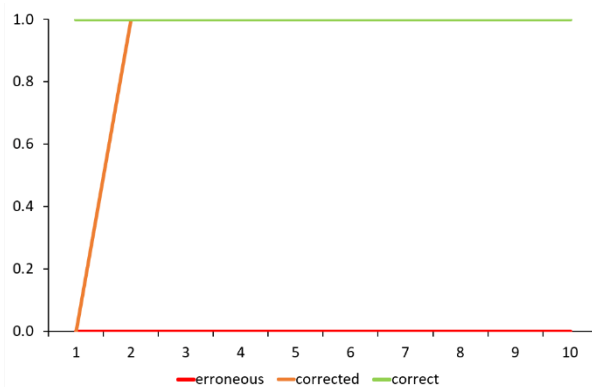
42. táblázat: Algoritmus egy új számozott bekezdés beillesztésére a helyes dokumentumba.

Az elemzések következő fázisában ugyanazt a módosítást tízszer megisméltük a hibás, a javított és a helyes dokumentumban, rögzítve a folyamatot, és mérve a feladatok elvégzéséhez szükséges időt. Az eredményt a 35. ábra mutatja be. Amennyiben egy hibás dokumentumot többször módosítunk, akkor az lineáris közelítéssel írható le. Ha azonban a hibákat kijavítjuk, akkor a javítás idejét egyszer használjuk fel, és utána csak a módosítás idejét adjuk hozzá. A hibás dokumentumban ennek az egyszerű feladatnak (egy számozott bekezdés beillesztése) a fenntarthatósági aránya 0,0001. Ez az eredmény azt mutatja, hogy a kézi számozás kezelése idő-, emberi és gépi erőforrás veszteséget termel, ezzel igazolva a [T5] tézist. Továbbá a kézi számozás magában hordozza a kihagyott vagy duplikált számok kockázatát, ami tovább rontja a dokumentum minőségét.



35. ábra: Egy számozott bekezdés beillesztéséhez szükséges idő az eredeti, a javított és a helyes dokumentumokban.

A három dokumentumban megismételt módosítások fenntarthatósági arányának kiszámítása és ezek ábrázolása (36. ábra) azt mutatja, hogy a dokumentum javítása után csak azokat a lépéseket hajtjuk végre, amelyeket a felhasználó eredetileg tervezett [162]. A javításon túl nincs veszteség, mert nem kell ismételten végrehajtani a járulékos lépéseket. Ez egyrészt bizonyítja, hogy a hibás dokumentumok fenntarthatósági aránya alacsonyabb, mint a helyes analógjuké, másrészt, hogy a dokumentum javítása az ismételt módosítási feladatok előtt növeli a fenntarthatósági arányt. Ezek a megállapítások alátámasztják a [T4] tézist.



36. ábra: A számozott bekezdésnek beillesztésének fenntarthatósági aránya az eredeti, a javított és a helyes dokumentumba.

10. Összefoglaló

Korábbi kutatások bebizonyították, hogy a táblázatkezelés programozási szemléletű oktatása [46][47][88][90] sokkal hatékonyabb [42][92][93], mint a széles körben elfogadott, eszközközpontú, alacsony mathability módszerek. A felületalapú megközelítési módszerek – amelyek a felhasználói felület működésének tanítására, a táblázatok begépelésére, a varázslók és a súgó böngészésére, a minél több funkciót és parancsot bemutató tankönyvekre és az újabb és újabb szolgáltatásokat kínáló szoftvekre összpontosítanak – nem támogatják a sémaépítést. A hosszú távú memóriában felépített sémák nélkül a gyors és lassú gondolkodás [32] nem alkalmazható hatékonyan [198], ami hibás táblázatkezelési dokumentumokhoz [78][198], és végső soron komoly pénzügyi veszteségekhez vezet mind az emberi, mind a gépi erőforrásokat tekintve [79].

Dolgozatom egyrészt a magyarországi 7–10. évfolyamos tanulók országos tesztelésének táblázatkezelés témakörhöz köthető részleteit mutatja be. A feladatok többségét a megértés SOLO kategóriái alapján értékeltük, amely a programozás tanításában széles körben elfogadott módszer. A tesztben bemutatott feladatok programozási jellegének megfelelően mind a táblázatkezelés, mind az önértékelés SOLO kategóriáinak négy szintjét állítottuk fel. A tanulók által megadott önértékelési értékek azt mutatják, hogy a tanulók nagyon magabiztosak a táblázatkezelési ismereteik megítélésében, különösen azok a tanulók, akik az iskolában tanultak táblázatkezelést. Ezzel szemben a diákok eredményeinek többsége Prestructural, ami azt jelenti, hogy nem rendelkeznek azonosítható tudáselemekkel. A diákok eredményei azt mutatják, hogy az informatikaoktatást jellemző dekontextualizált (tartalom nélküli vagy toy situation szöveggörnyezet), eszközközpontú, alacsony mathability megközelítések nem megfelelőek a tanulók problémamegoldó és számítógépes gondolkodásának fejlesztéséhez [10][14][31][47][58][88][90][92][93][177][197]. Továbbá, teljes összhangban a Dunning-Kruger-effektussal [199], a megértés Prestructural szintje (a lényeg kihagyása) megakadályozza, hogy a tanulók lássák, hogy mit tudnak és mit nem.

A tudástranszferelemek aktiválását tekintve az eredmények azt mutatják, hogy a matematikából származó egyes tudáselemek lehetővé teszik a tanulók számára, hogy elérjék a megértés Relation szintjét egy-egy matematikához köthető feladatban. Az adatkezeléshez köthető tudástranszferelemek nyomai azonban alig mutatkoznak, ami megakadályozza, hogy a tanulók a tudáselemeket egyik feladatból a másikba, vagyis az órai tevékenységből a problémamegoldásba, illetve a függvények nevének ismeretétől a Relation szintű megértésre transzformálják. Jelenleg úgy tűnik, hogy a táblázatkezelés tanítása csak a felhasználói felület működésének megismerését szolgálja, minden további értelmes cél nélkül. A teszt eredményeiből arra következtethetünk, hogy

a táblázatkezelés oktatása nem támogatja a tanulók számítógépes gondolkodásának és algoritmikus készségének fejlesztését. Ennek eredményeképpen a tanulók nem tudják elérni a megértés Relation szintjét, nem is beszélve az Extended abstract kategóriáról. Azt is meg kell jegyeznünk, hogy ellentétben Prensky digitális generációkkal kapcsolatos, nem bizonyított állításaival [70], még a digitális gyerekeknek is nagy szükségük van oktatási támogatásra. Ennek a támogatásnak a biztosításához olyan szakértő tanárookra van szükség [33][45][61], akik képesek elszakadni a széles körben elfogadott eszközközpontú, alacsony mathability tanítási és tesztelési megközelítésektől, és átváltani a magas mathability problémamegoldó megközelítésekre [11][13][48][49][92][93]. Ezen eredményekkel a [T1] tézis bizonyításra került.

A mérés következő fázisában a digitális eszközhasználat és a táblázatkezelési ismeretek közötti összefüggést vizsgáltuk. A digitális eszközhasználatra fordított időt az iskola esetében egy hétben, személyes használatot tekintve pedig egy napban mértük. A kérdés az volt, hogy a digitális eszközökön töltött idő hogyan befolyásolhatja a tanulók táblázatkezelési tudását. A teszt táblázatkezelési feladatainak eredményei azt mutatták, hogy nincs kapcsolat a tanulók táblázatkezeléssel kapcsolatos tudása és a digitális eszközökkel töltött idő között. Az eszközökiskolai használata esetén találtuk a legkedvezőtlenebb eredményt, ahol az eszközökre fordított idő negatívan hat a diákok tudására. Ez a negatív hatás erősebb az iskolai mobilhasználat esetében. A további elemzések során, ahol csak azoknak a tanulóknak az eredményeit vettük figyelembe, akik nulla pontnál többet értek el, azt találtuk, hogy ezek az eszközök nem hatnak negatívan a tanulók eredményeire. Az adatelemzés és képletkiegészítés feladatokban pozitív hatás mutatható ki.

A vizsgálatok során nem tértünk ki az osztálytermi tevékenységek részleteire, azonban azt tapasztaltuk, hogy a diákok az iskolában elsősorban kommunikációra és adatlekérdezésre használják ezeket az eszközöket, beleértve a rendkívül népszerű online szavazó- és értékelőrendszereket is. Ezek alapján a diákoknak nincs meg a motivációjuk és az igényük arra, hogy a digitális eszközöket problémamegoldásra és/vagy produktív tevékenységekre használják és az iskola nem készíti a diákokat értékes digitális termékek létrehozására, és ugyanez az iskola nem tud objektív mérést biztosítani a diákok digitális tevékenységeinek értékelésére az értelmetlen kattintgatáson túl.

Ezek az eredmények egyértelműen bizonyították a [T2] tézist, továbbá mutatják, hogy a digitális bennszülöttek születése nem jelent digitális tudást, legalábbis a táblázatkezelés terén nem. A tanulók digitális kompetenciái, számítógépes gondolkodási készségei és a számítógéppel való kommunikáció képessége tudatos oktatási koncepciókat, a tanárok digitális és módszertani készségeinek és képességeinek fejlesztését igénylik.

Dolgozatom második felében a mini-kompetenciateszt szövegkezelési eredményeit és a digitális szövegek entrópiáját, fenntarthatósági arányát mutattam be. A mini-kompetenciateszt számozás feladatának eredményei azt mutatják, hogy a 7–10. osztályos tanulók többsége nem tud különbséget tenni a kézi és az automatikus számozás között a MS Word mintákban. Nem tudják, hogy a felhasználói felületen elhelyezett egyetlen információ – a kurzor pozíciója – elég lehet annak eldöntéséhez, hogy melyik minta kézi vagy automatikus számozású. A tanulást támogató, segítő dokumentumok jellemzően az eszközökre koncentrálnak, nem fordítva figyelmet a felhasználói felületre helyezett információkra, a szövegszerkesztő parancsok szemantikájára [195], vagy a nem-nyomtató karakterek által hordozott információkra [147]–[150]. A mért adatok alapján arra a következtetésre jutottunk, hogy az automatikus számozás kezelésének entrópiája 3,0961. Ez azt jelenti, hogy legalább három bitre van szükség ahhoz, hogy a szövegkezelő felhasználói felületének információit a végfelhasználóhoz eljuttassuk. Nem elég tehát csak azt mondani, hogy az automatikus számozás egy kattintás az egyik számozási gombra. A GUI üzeneteit tudatosan kell megtanítani és megtanulni, ami magába foglalja a hibás dokumentumok elemzését, a hibák felismerését is. Ez az eredmény alátámasztja a [T3] tézist.

A [T4] tézis bizonyítására két hibás dokumentumot elemeztünk, javítottunk, formáztunk, majd módosításokat végeztünk mind az eredeti, mind pedig a javított dokumentumokban. Megállapítást nyert, hogy az eredeti hibás dokumentumok módosítása lényegesen erőforrásigényesebb, mint a helyes dokumentumoké. Általánosságban megállapíthatjuk tehát, hogy a hibás dokumentumok kezelése nehezebb, bonyolultabb és forrásigényesebb, mint a helyes dokumentumoké.

Megállapítottuk továbbá, hogy egy feladat fenntarthatósági aránya levezethető a feladat atomi lépéseinek időtartamából, információtartalmából és entrópiájából. Ez az érték a $[0, 1]$ tartományban van, ahol 0 a járulékos javításokat, 1 pedig a helyes dokumentum tervezett módosításainak fenntarthatósági arányát jelzi. Hibás dokumentum esetén ez a tartomány $]0, 1[$. A mérési folyamat lépéseit az ANLITA naplózó alkalmazással rögzítettük [193], amely a végfelhasználói műveletek nyomon követésére szolgál. A hibás dokumentum módosításának a fenntarthatósági aránya 0-hoz közeli érték az elemzett dokumentumokban, mely a [T5] tézist támasztja alá. Ez az eredmény azt jelenti, hogy az emberi- és gépi erőforrásveszteség rendkívül magas, a végfelhasználók alacsony számítógépes gondolkodási készségei, képességei és a minőségi digitális adatfeldolgozás iránti igény hiánya miatt. Sokan úgy vélik, hogy minél kevésbé kidolgozott egy dokumentum, annál könnyebb kezelni vagy módosítani. Ez a tévhit abból ered, hogy összekeverik a felhasználói készségek fejletlenségét a digitális termékek fejletlenségével [199][200]. Mi azonban bizonyítékot

találtunk arra, hogy minél jobban strukturáltak a szövegek, annál kevésbé igényesek a módosítási folyamatok. Ez a paradoxon arra utal, hogy azok a végfelhasználók hoznak létre hibás dokumentumokat, akiknek a számítógépes gondolkodási készsége alacsony, holott az ilyen dokumentumok kezelése nagyobb szakértelmet igényel, mint a helyes megfelelőjük feldolgozása. Ez egy ördögi kör, amelyből csak úgy szabadulhatunk ki, ha a végfelhasználók számítógépes gondolkodási készsége fejlődik. Az eredmények nyilvánvalóvá teszik, hogy egy feladat elvégzéséhez szükséges adatmennyiség nem feltétlenül jelenti az eredmény minőségét, ami magába foglalja azt is, hogy az alacsony minőségű adatok feldolgozásához több adatra van szükség. Az is bizonyítást nyert, hogy a hibák kijavítása rendkívül hatékony módszer a veszteségek elkerülése érdekében. Ennek megfelelően, mielőtt ismételt módosításokat alkalmaznánk egy hibás dokumentumban, a hibákat el kell távolítani. Ezek az eredmények teljes összhangban vannak a hatékony ipari termelékenységgel kapcsolatos, húzó rendszerű elméletekkel [5][146].

Hibás Word dokumentumok milliói keringenek zárt közösségekben és az interneten egyaránt, amelyek helytelen mintákat szolgáltatnak. Általánosságban azonban megállapítható, hogy ezek a kérdések és problémák szinte észrevétlenül maradnak mind az oktatás, mind a kutatás, mind az ipar résztvevői előtt. Tovább nehezíti és bonyolítja a helyzetet, hogy még az oktatási segédanyagok sem felelnek meg a megfelelően szerkesztett és formázott szöveg követelményeinek, és hibás példát mutatnak.

A jövőt tekintve, forrásokat, támogatókat szeretnénk találni a tanulók, tanárok, oktatók, irodai dolgozók képzéséhez. Ezeknek a képzéseknek a célja, hogy a végfelhasználók megtanulják, hogyan tudnak minőségi szövegkezelési munkát végezni. Ezzel együtt lehetőségünk nyílna az emberközpontú, feladatorientált fenntarthatósági fejlődés tanulmányozására. Szeretnénk felismertetni, hogy ha a felhasználók nem látják a komplex programok lényegét, továbbá nem ismerik az informatika fogalmait, a természetes nyelvű digitális szövegek feldolgozását, akkor az emberi és a gépi erőforrások pazarlásához vezet.

A változtatáshoz újra kell gondolni az oktatást, és olyan módszereket kell bevezetni, amelyekkel növelhető a minőség és csökkenthető a veszteség. Céljaink eléréséhez először kis lépéseket kell tenni [1][2][5][144][146] és lassan fejlődni. Ez egy időigényes folyamat, amelyet nem lehet siettetni. Nagy szükségünk van szakértő tanárookra [61] bármely tantárgyból és tudományból [44][59][60], akik képesek hatékonyan használni a digitális eszközöket az oktatásban, és képesek fejleszteni a tanulók számítógépes gondolkodását [29][105]–[107][157]. Tudják továbbá, hogy hogyan lehet hatékonyan alkalmazni a gyors és lassú gondolkodást a valós digitális problémamegoldásban [32][33].

11. Hosszútávú tervek

Hosszútávú célunk felhívni a figyelmet az ember szerepére a digitális fenntarthatóságban és megoldásokat kínálni az emberi és gépi erőforrások pazarlásának csökkentésére az oktatáson keresztül [33][61]. További célunk és/vagy reményünk, hogy képesek leszünk elérni egy olyan közösséget, amely nyitott a digitális fenntarthatóság problémáira és a helytelen és hibás végfelhasználói tevékenységekből származó veszteségek csökkentésére és eliminálására. Kutatási eredményeink alapján azt gondoljuk, hogy ezen cél elérése érdekében először is fejleszteni kell a tanárok számítógépes gondolkodási készségét – függetlenül a tantárgyuktól [61]. Amint képesek felismerni az információtartalom fontosságát és a digitális szövegek fenntarthatóságát, el tudják dönteni, hogy egy szöveg megfelelően van-e szerkesztve, meg tudják tanítani a módszert a diákjaiknak, majd elvárhatják tőlük a digitális szövegek helyes kezelését. Ebben a folyamatban a kognitív terhelés [29][32][45][105], a TPCCK (Technological Pedagogical Content Knowledge) [60], és a tantárgyi integráció [17][157] döntő szerepet játszanak.

Eredményeink alapján további vizsgálatokra van szükség, hogy bizonyítani tudjuk, mennyire igényesek a különböző szövegek és a módosítási igények, de az már most is nyilvánvaló, hogy a hibás szövegkezelés és az alapvető számítógépes gondolkodási készségek hiánya értelmetlenül felemészti erőforrásainkat. További kutatási kérdések, hogy a fenntarthatósági arány hogyan konvertálható minőségi oktatásra, hogyan számítható ki a tényleges pazarlás pénzben kifejezve, és hogyan lehet az oktatást és a veszteségeket egyensúlyba hozni, hogy az emberközpontú digitális irodai munka fenntartható legyen. A kiválasztott mintadokumentumok nem egyediek, számos hasonló dokumentum elérhető az interneten, amelyeket a szerzők megosztanak és azok az olvasók újra megosztanak, akik nem ismerik az ezekben a dokumentumokban található hibákat. Ez magában foglalja azt is, hogy a fenntarthatósági elemzések elvégzéséhez szükséges források óriási mennyiségben állnak rendelkezésre.

A bemutatott módszerrel objektív eszközt kínálunk az összehasonlító elemzéshez, valamint az oktatásban és az iparban (a való világban) keletkezett veszteségek méréséhez. Kiszámíthatjuk, hogy egy dokumentum módosításához mennyi időre, hány alkalmazottra, irodára, számítógépre stb. van szükség, illetve mennyi energiát pazarolnak el a helytelen szövegkezeléssel. Remélhetőleg a dolgozatban bemutatott megközelítés és módszer nem tűnik el észrevétel nélkül – ahogy az a korábbi hasonló eredményekkel történt –, és az eredmények alapján felhívhatjuk a figyelmet az emberközpontú digitális fenntarthatóság szerepére és fontosságára. Továbbá reméljük, hogy a jövőben olyan tankönyveket látunk majd, amelyek maguk mögött hagyják az eszközközpontú, toló

elvű számítógépes oktatási rendszereket, és áttérnek az emberközpontú, koncepció-alapú problémamegoldásra.

Fontosnak tartom még kiemelni, hogy a dolgozatban bemutatott eredményeket szövegszerkesztői dokumentumokon végzett mérések alapján fogalmaztuk meg. Ugyanakkor, a módszert alkalmazva, hasonló méréseket tudunk végezni annak ellenőrzése, hogy egyéb szöveges dokumentumok (pl. prezentációk, weblapok, hirdetések, online oktatási értékelő és ellenőrző rendszerek) mennyiben felelnek meg a helyesen szerkesztett dokumentum definíciójának, ezek létrehozásához és módosításához mekkora információtartalom szükséges és milyen a dokumentumok fenntarthatósági aránya, mennyi veszteséget termelünk a helytelen dokumentumkezelési gyakorlatokkal.

12. Köszönetnyilvánítás

Köszönettel tartozok témavezetőmnek Dr. Csernoch Máriának, aki közel 10 éve önzetlenül támogat, segít. Szakmai precizitásával, példamutató magatartásával irányt mutatott és vezetett szakmai utamon, minden felmerülő probléma esetén készséggel a rendelkezésemre állt.

Köszönetemet szeretném kifejezni a Debreceni Egyetem Informatikai Tudományok Doktori Iskola, Elméleti számítástudomány, adatvédelem és kriptográfia doktori program vezetőjének, Dr. Vaszil Györgynek, a Debreceni Egyetem Informatikai Karának, és az „EFOP-3.6.3-VEKOP-16-2017-00002 Integrált kutatói utánpótlásképzési program az informatika és számítástudomány diszciplináris területein” pályázat vezetőjének a kutatás megvalósításához nyújtott anyagi és erkölcsi támogatásáért.

Köszönöm a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatalnak, hogy a Kooperatív Doktori Program keretein belül nyújtott ösztöndíjjal hozzájárultak ahhoz, hogy a kutatás társadalmi, gazdasági hasznosításában részt vehessünk.

Köszönöm a mini-kompetenciatesztben részt vevő iskoláknak, pedagógusoknak és tanulóknak, akik hozzájárultak a mérés megvalósításához.

Köszönöm az ELMS Informatikai Zrt.-nek és a Karcagi Nagykun Református Gimnáziumnak, hogy lehetőséget biztosítottak az új módszerek alkalmazására és tesztelésére.

Hálával tartozom szüleimnek, férjemnek és minden családtagomnak, akik feltétel nélkül támogattak, bátorítottak és segítettek abban, hogy megvalósítsam terveimet.

13. Summary

Previous research proved that teaching spreadsheeting from a programming aspect [46][47][88][90] is much more effective [42][92][93] than the widely accepted tool-centred, low-mathability methods. The surface approach methods – focusing on the teaching of software interface, typing spreadsheet tables, browsing wizards and helps, coursebooks introducing as many functions as possible, and newer and newer functions and features of the softwares – do not support schema construction. Without schemata built up in the long-term memory, fast and slow thinking [32] cannot be applied effectively [198], which leads to erroneous spreadsheet documents [78][198], and ultimately to serious financial losses, both in human and machine resources [79].

On the one hand, my thesis presents the details of the spreadsheet section of a nationwide testing of Grades 7–10 students in Hungary. The test focuses on the students' problem-solving skills, how knowledge-transfer elements are activated, and which classroom activities form the students' self-assessment values. Most of the spreadsheet tasks are evaluated by the SOLO categories of understanding, which is a widely accepted method in teaching programming. In accordance with the programming nature of the tasks presented in the test, four levels of both spreadsheeting and self-assessment SOLO categories are set up. The self-assessment values provided by the students reveal a strong confidence in their spreadsheet knowledge, especially with those students who studied spreadsheeting in school. On the contrary, most of the students' results based on the three tasks of the test are pre-structural, which means that no recognizable pieces of knowledge are present.

The students' results demonstrate that the decontextualized, tool-centred, low-mathability approaches that primarily accepted in computer science education are inadequate for developing students' problem-solving and computational thinking skills [10][14][31][47][58][88][90][92][93][177][197]. Furthermore, in complete accordance with the Dunning-Kruger effect [199], the pre-structural level of understanding (missing the point) prevents students from seeing what they do not know.

Considering the activation of knowledge-transfer items, the results reveal that some pieces from mathematics allow students to reach the relational level of understanding in the specific, math-related task. However, hardly any traces of spreadsheet knowledge-transfer are evident, which prevents students from transforming pieces of knowledge from one task to another, or, in other words, from classroom activities to problem-solving, and from knowing the names of functions to applying them at the relational level of understanding.

At present, studying spreadsheeting seems to be only for the sake of learning how the user-interface works, without any meaningful further aims. We can conclude from the results of the test that spreadsheet education at this stage does not support the development of students' computational thinking and algorithmic skills. As a result, students cannot reach the relational level of understanding, not to mention the extended abstract category, which is one level above relational. One must also note that – contrary to Prensky's unproven claims considering digital generations [70] – even digital children are in great need of educational support. To provide this support, we must have expert teachers [33][45][61] who can break away from the widely accepted tool-centred, low-mathability teaching and testing approaches, and switch to high-mathability problem-solving approaches [11][13][48][49][92][93]. These results prove thesis [T1].

The next phase of the research revealed how Grades 7–10 students' spreadsheet competences and self-assessment are affected by the time spent on computers and mobile devices. The time spent on the digital devices was measured by four related questions, considering the computer and mobile use in school in a week and altogether in a day. The question was how the time spent on digital devices can affect students' spreadsheet skills. Considering students' spreadsheet knowledge, it is remarkable that around two-thirds of the students scored zero. The analyses revealed that in general computer use in school and mobile use altogether do not affect spreadsheet knowledge, while mobile use in school has a negative effect; the more they use these tools, the lower their results are. In our further analyses, where we only considered the results of those students who scored above zero, we found that these tools did not have a negative impact on students' results. A positive effect was found in the data analysis and formula completion tasks.

This study does not go into the details of classroom activities; however, it seems reasonable that students primarily use these devices for communication and data retrieval, including the extremely popular online voting systems. It seems that students do not have the motivation and the need to use digital tools for problem-solving and/or productive activities. It also seems that the school does not make students create valuable digital products, and the very same school cannot provide an objective measurement to evaluate students' digital activities beyond pointless clicking. These results clearly proved thesis [T2] and reveal that being born digital natives does not make digital knowledge, at least not in spreadsheeting. Students' digital competences, computational thinking skills, and the ability to communicate with a machine require conscious educational concepts and the development of teachers' related digital and methodological skills and abilities.

The second part of my thesis presents the results of testing text management knowledge and explains the entropy and sustainability rate of digital texts. The results of the mini competence test reveal that most of the students in grades 7–10 cannot distinguish between manual and automated numbering in MS Word samples. They do not know that one bit of information put on the GUI – the position of the cursor – can be enough to decide which sample has manual or automated numbering. Our research revealed that the supporting documents focus exclusively on tools, paying no attention to the information put on the GUI, the semantics of the commands of the word processor [195], or the information that the non-printing characters carry [147]–[150]. Based on the test result, it is found that the entropy of handling automated numbering is 3.0961. This means that at least three bits are needed to transfer the information from the GUI of the word processor to end-users. To gain the level of knowledge at which one bit – the position of the cursor – is enough to tell whether the numbering is automated, three bits of information must be put on the channel. This implies that, in the teaching-learning process, saying that automated numbering is nothing more than one click on one of the numbering buttons is not enough. The messages of the GUI must be taught and learned consciously. End-users must gain the knowledge that the developers of the word processors put on the GUI. This result proves thesis [T3].

To prove thesis [T4], two erroneous documents were analysed, corrected, formatted, and then modified in both the original and corrected documents. It was found that it takes significantly more resources to modify the original incorrect documents than the correct documents. In general, we can conclude that handling erroneous documents is more difficult, complicated, complex, and demanding than handling correct ones. We have found that the sustainability rate of a task can be derived from the duration, the information content, and the entropy of the atomic steps of the task. This value is in the $[0, 1]$ range: 0 indicating collateral corrections, and 1 the planned modifications of a correct document. In an erroneous document, this range is $]0, 1[$. The steps of the measurement process were recorded with the logging application ANLITA [193], dedicated to tracking end-user input actions. In the sample erroneous document, we found the sustainability rates to be close to 0 for both modification tasks, meaning that the waste of human and machine resources is extremely high, because of end-users' weak computational thinking skills and ignorance of quality digital work. This result proves thesis [T5].

Many believe that the less developed a document is, the easier it is to handle, modify, or manage it. This misconception originates from confusing the underdevelopment of user-skills with those of digital products [199][200]. However, we found proof that the more properly structured and built texts are, the less demanding they are to maintain. This paradox suggests that those end-

users create erroneous documents whose computational thinking skills are low, even though handling such content demands more expertise than is needed when handling correct ones. This is a vicious circle that we can only escape by making end-users' computational thinking skills more developed. The results make it obvious that the amount of data required to complete a task does not necessarily imply the quality of the outcome, and low-quality data requires more data. It is also proven that correcting the errors first is highly profitable before applying repeated modifications. These findings are in complete accordance with productivity theories in industry [5][146].

Millions of erroneous documents can be found in our close environment, in our private collections, and on the internet. In general, these issues and problems go almost unnoticed by those involved in education, research, and industry. The situation is further complicated by the fact that even the teaching aids do not meet the requirements of a properly edited and formatted text and set erroneous example.

Thinking forward, we should find resources to train our students, teachers, advisors, clerks, and office workers, to learn how to do quality work in text management, together with a research centre, where the development of human-centred sustainability could be studied. It must be recognized that not seeing the essence of complex programs and not finding the concepts of informatics in handling natural language texts with a word processor leads to a waste of human and machine resources. To make changes, education must be rethought, and methods should be introduced which can increase quality and decrease waste. To achieve our goals, small steps should be taken first and develop slowly [1][2][5][144][146]. This is a time-consuming process which cannot be rushed.

Our long-term goal is to call attention to the human role in digital sustainability and offer solutions to reduce the waste of human and machine resources through education. This being crucial in the development process, as has been proven by industry. Our further target condition and/or hope is that we will be able to reach an audience sensitive to digital sustainability and eliminate waste generated by negligent end-user activities.

To accomplish this goal, first the computational thinking skills of teachers – regardless of their subjects – should be developed [61]. Once they are able to recognize the importance of information content and the sustainability of digital texts, they can decide whether a text is properly edited, they can teach the method to their students, and then we can expect them to handle digital texts correctly. In this process, cognitive load [29][32][45][105], TPCK (Technological Pedagogical Content Knowledge) [60], and subject integration play crucial roles [17][157].

Based on these results, more studies are required to find out how demanding various texts and other modification tasks are, but it is already obvious that erroneous text management and the lack of fundamental computational thinking skills use up our resources senselessly. Further research questions are how the sustainability rate can be converted into quality education, how the actual waste can be calculated in currency, and how education and waste can be balanced to make human-centred digital office work sustainable.

The selected sample document is not unique, several similar – both in length and content – are available on the internet, shared by authors and re-shared by readers who are unaware of the errors which these documents carry. With the presented method, an objective tool is offered to make comparative analyses and measure the losses generated in both education and industry (in the real world). We can calculate how much time is needed to fulfil a modification task, how many computers, how much workspace and energy are needed and wasted due to negligent text handling.

Hopefully, this approach will not end up unnoticed as its predecessors, and based on these results, we can call attention to the role and importance of human-centred digital sustainability. Furthermore, we hope to see course books which leave behind the tool-centred push production systems, and switch to human-centred, concept-based problem-solving.

It is also important to note that the results presented in this dissertation are based on measurements of word processing documents. However, by applying the method, similar measurements can be made to verify the extent to which other text documents correspond to the definition of a correctly edited document, the amount of information required to create and modify them, and the sustainability rate of documents, and how much loss is generated through inappropriate document management practices.

14. Irodalomjegyzék

- [1] T. Ohno, *Toyota Production System: Beyond Large-Scale Production*. New York: Productivity Press, 1988.
- [2] J. F. Krafcik, Triumph of the Lean Production System, *Sloan Management Review*, vol. 30, no. 1, pp. 41–52, 1988.
- [3] N. Modig and P. Åhlström, *This Is Lean. Resolving the Efficiency Paradox*. Stockholm: Rheologica Publishing, 2018.
- [4] J. P. Womack and D. T. Jones, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. London: Simon & Schuster, 2003.
- [5] M. Rother, *Toyota Kata: Managing People for Continuous Improvement, Adaptiveness, and Superior Results*. New York: McGraw-Hill, 2010.
- [6] C. Wolfram, *The Math(s) FIX: An Education Blueprint for the AI Age*. Wolfram Media, Inc., 2020.
- [7] M. Dondi, J. Klier, F. Panier, and J. Schubert, Defining the skills citizens will need in the future world of work, Future-citizen skills McKinsey, European Court of Human Rights, *European Convention on Human Rights*. Strasbourg, UE: Council of Europe, 1950. Available at: https://www.echr.coe.int/documents/convention_eng.pdf. (letöltés dátuma: 2024 január).
- [8] Future of jobs report 2023 - World Economic Forum, WEF_Future_of_Jobs_2023.pdf, https://www3.weforum.org/docs/WEF_Future_of_Jobs_2023.pdf (letöltés dátuma: 2024 január).
- [9] T. Nagy, Az informatika kerettanterv elemzése, *3th International Interdisciplinary Conference 2018*. Available at: https://detep.unideb.hu/sites/default/files/upload_documents/kotet_interdisz_3.pdf. (letöltés dátuma: 2024 január).
- [10] M. Csernoch, P. Biró, J. Máth and K. Abari, Testing algorithmic skills in traditional and non-traditional programming environments. *Informatics in Education*, 14(2), 175–197, 2015. <https://doi.org/10.15388/infedu.2015.11>
- [11] P. Biró and M. Csernoch, The mathability of computer problem solving approaches, *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 111–114, 2015. doi:10.1109/coginfo-com.2015.7390574
- [12] M. Csernoch and P. Biro, First year students’ attitude to computer problem solving, *2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 2017. doi:10.1109/coginfo-com.2017.8268247

- [13] P. Biro and M. Csernoch, The mathability of Spreadsheet Tools, *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfo-Com)*, Oct. 2015. doi:10.1109/coginfocom.2015.7390573
- [14] P. Biró and M. Csernoch, Maths problems in pseudo-codes compared to computer usage, *END2018: International conference on Education and New Developments*, pp. 341–346, 2018.
- [15] L. Zsakó, Informatika Nemzeti Alapanterv 2020, *NFODIDACT 2015*, 2015. Available at: <https://people.inf.elte.hu/szlavi/InfoDidact15/Manuscripts/ZsL.pdf>. (letöltés dátuma: 2024 január).
- [16] W. Gander, A. Petit, G. Berry, B. Demo, J. Vahrenhold, A. McGettrick, R. Boyle, M. Drechsler, A. Mendelson, C. Stephenson, C. Ghezzi and B. Meyer, *Informatics Education: Europe Cannot Afford to Miss the Boat Report of the Joint Informatics Europe & ACM Europe Working Group on Informatics Education April 2013*. Informatics Europe & ACM Europe Working Group on Informatics Education, 2013.
- [17] J. Vahrenhold, E. Nardelli, C. Pereira, G. Berry, M. Caspersen, J. Gal-Ezer, M. Kölling, A. McGettrick and M. Westermeier, *Informatics Education in Europe: Are We All In The Same Boat? Report by The Committee on European Computing Education (CECE)*. New York, NY: Association for Computing Machinery, 2017.
- [18] 130/1995. (X. 26.) Korm. Rendelete a Nemzeti Alapanterv Kiadásáról, Bevezetéséről és Alkalmazásáról; Korona Kiadó: Budapest, Magyarország, 1995; ISBN 9638153873.
- [19] 202/2007. (VII. 31.) Korm. Rendelete a Nemzeti Alapanterv kiadásáról, bevezetéséről és alkalmazásáról. http://www.nefmi.gov.hu/letolt/kozokt/nat_070926.pdf
- [20] 110/2012. (VI. 4.) Korm. rendelete a Nemzeti alapanterv kiadásáról, bevezetéséről és alkalmazásáról, http://njt.hu/cgi_bin/njt_doc.cgi?docid=149257.388616. (letöltés dátuma: 2024 január).
- [21] 5/2020. (I. 31.) Korm. Rendelete a Nemzeti Alapanterv kiadásáról, bevezetéséről és alkalmazásáról Szóló 110/2012. (VI. 4.), http://ofi.hu/sites/default/files/attachments/mk_nat_20121.pdf. (letöltés dátuma: 2024 január).
- [22] OFI 2020a, Kerettanterv Az általános Iskola 5–8. évfolyama számára. https://www.oktatas.hu/koznevelis/kerettantervek/2020_nat/kerettanterv_alt_isk_5_8. (letöltés dátuma: 2024 január).
- [23] OFI 2020b, Kerettanterv a gimnáziumok 9–12. évfolyama számára, 2020. https://www.oktatas.hu/koznevelis/kerettantervek/2020_nat/kerettanterv_gimn_9_12_evf. (letöltés dátuma: 2024 január).

- [24] OFI 2013a. Kerettanterv az általános iskola 5–8. évfolyamára. Kötelező tantárgyak 2.2.15 http://kerettanterv.ofi.hu/02_melleklet_5-8/2.2.15_informat_5-8.doc. (letöltés dátuma: 2024 január).
- [25] OFI 2013b. Kerettanterv a gimnáziumok 9–12. évfolyama számára. Kötelező tantárgyak 3.2.16 http://kerettanterv.ofi.hu/03_melleklet_9-12/3.2.16_informat_9-12.doc. (letöltés dátuma: 2024 január).
- [26] G. Hildreth, *Learning the three R's (2nd ed.)*, 1947. doi:10.1037/14928-000
- [27] J. M. Wing, Computational thinking, *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006. doi:10.1145/1118178.1118215
- [28] P. Baranyi and A. Gilányi, Mathability: Emulating and enhancing human mathematical capabilities, *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*, 2013. doi:10.1109/coginfo-com.2013.6719309
- [29] G. Polya, *How to Solve It*. Princeton University Press, 1945.
- [30] K. Freiermuth, J. Hromkovič, and B. Steffen, Creating and testing textbooks for secondary schools, *International Conference on Informatics in Secondary Schools - Evolution and Perspectives ISSEP 2008: Informatics Education - Supporting Computational Thinking*, pp. 216–228, Jul. 2008. doi:10.1007/978-3-540-69924-8_20
- [31] P. Papp and M. Csernoch, A táblázatkezelés a problémamegoldás? *InfoÉra 2018*, pp. 187–201, Available at: https://people.inf.elte.hu/szlavi/InfoDidact18/Infodidact2018.pdf?fbclid=IwAR38Vk3h2w_81Iv61C76V6xkEr-pLdxX4Ubc96P4VuR4EXYmIFW0b5-Jj0Z4. (letöltés dátuma: 2024 január).
- [32] D. Kahneman, *Thinking, Fast and Slow*. Penguin books, 2011.
- [33] M. Csernoch, Thinking fast and slow in computer problem solving, *Journal of Software Engineering and Applications*, vol. 10, no. 01, pp. 11–40, 2017. doi:10.4236/jsea.2017.101002
- [34] D. H. Dolmans, S. M. Loyens, H. Marcq, and D. Gijbels, Deep and surface learning in problem-based learning: A review of the literature, *Advances in Health Sciences Education*, vol. 21, no. 5, pp. 1087–1112, Nov. 2015. doi:10.1007/s10459-015-9645-6
- [35] M. Gove, Digital Literacy Campaign – Michael Gove’s speech in full, *The Guardian*, <http://www.theguardian.com/education/2012/jan/11/digital-literacy-michael-gove-speech> (letöltés dátuma: 2023 január).
- [36] T. Bell, and H. Newton, Unplugging Computer Science, *Improving Computer Science Education*, pp. 66–81, 2013.

- [37] W. Gander, Informatics in Schools? - Urgently Needed!, ECSS 2012, Barcelona, Nov. 21, 2012. Available at: <http://www.inf.ethz.ch/personal/gander/talks/GanderECSS2012.pdf>. (letöltés dátuma: 2024 január).
- [38] J. Hromkovič and B. Steffen, Why teaching informatics in schools is as important as teaching mathematics and Natural Sciences, *Lecture Notes in Computer Science*, pp. 21–30, 2011. doi:10.1007/978-3-642-24722-4_3
- [39] P. Baranyi, A. Csapo, and G. Sallai, *Cognitive Infocommunications (COGINFOCOM)*. Springer, 2015.
- [40] E. Soloway, Should we teach students to program?, *Communications of the ACM*, vol. 36, no. 10, pp. 21–24, 1993. doi:10.1145/163430.164061
- [41] J. M. Carroll, Evaluation, description and invention: Paradigms for human-computer interaction, *Human Computer Interaction*, pp. 1–22, Aug. 1988. doi:10.21236/ada204617
- [42] S. Booth, *Learning to Program: A Phenomenographic Perspective*. Göteborg: Acta Universitatis Gothoburgensis, 1992.
- [43] M. Ben-Ari, Non-myths about programming, *Communications of the ACM*, vol. 54, no. 7, pp. 35–37, 2011. doi:10.1145/1965724.1965738
- [44] P. Mishra and M. J. Koehler, Technological Pedagogical Content Knowledge: A framework for teacher knowledge, *Teachers College Record: The Voice of Scholarship in Education*, vol. 108, no. 6, pp. 1017–1054, 2006. doi:10.1177/016146810610800610
- [45] J. Chen, D. Morris, and N. Mansour, Science Teachers’ Beliefs: Perceptions of Efficacy and the Nature of Scientific Knowledge and Knowing, *International Handbook of Research on Teachers’ Beliefs*, pp. 382–398, Nov. 2014. doi:10.4324/9780203108437-31
- [46] P. Warren, Learning to program: spreadsheets, scripting and HCI, *Proceedings of the Sixth Australasian Conference on Computing Education*, vol. 30, pp. 327–333, 2004.
- [47] P. Sestoft, *Spreadsheet Technology*. Copenhagen: IT University of Copenhagen, 2011.
- [48] M. Csernoch, *Programozás Táblázatkezelő Függvényekkel; Sprego: Táblázatkezelés Csupán Egy Tucát Függvénnyel*. Budapest: Műszaki Könyvkiadó Kft., 2014.
- [49] M. Csernoch and P. Biró, Sprego Programming, *Spreadsheets in Education*, vol. 8, 2015. doi:https://sie.scholasticahq.com/article/4638-sprego-programming

- [50] A. V. Aho, Computation and computational thinking, *The Computer Journal*, vol. 55, no. 7, pp. 832–835, Jun. 2012. doi:10.1093/comjnl/bxs074
- [51] C. Selby and J. Woollard, Computational Thinking: The Developing Definition. ITiCSE Conference, Canterbury, UK, 1–3 July 2013.
- [52] L. Mannila, V. Dagiene, B. Demo, N. Grgurina, C. Mirolo, L. Rolandsson and A. Settle, Computational thinking in K-9 education. Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference, Uppsala, Sweden, 21–25 June 2014; pp. 1–29.
- [53] M. Tedre and P. J. Denning, The long quest for computational thinking. 16th Koli Calling International Conference on Computing Education Research, Koli, Finland, 24–27 November 2016; pp. 120–129.
- [54] P. J. Denning, Remaining trouble spots with computational thinking, *Communications of the ACM*, vol. 60, no. 6, pp. 33–39, May 2017. doi:10.1145/2998438
- [55] M. Bower, L. N. Wood, J. W.M. Lai, C. Howe and R. Lister Improving the computational thinking pedagogical capabilities of School Teachers, *Australian Journal of Teacher Education*, vol. 42, no. 3, pp. 53–72, Mar. 2017. doi:10.14221/ajte.2017v42n3.4
- [56] V. J. Shute, C. Sun, and J. Asbell-Clarke, Demystifying computational thinking, *Educational Research Review*, vol. 22, pp. 142–158, Nov. 2017. doi:10.1016/j.edurev.2017.09.003
- [57] N. Garrett, Textbooks for responsible data analysis in Excel, *Journal of Education for Business*, vol. 90, no. 4, pp. 169–174, 2015. doi:10.1080/08832323.2015.1007908
- [58] P. Biró, M. Csernoch, K. Abari and J. Máth, Testing Algorithmic and Application Skills. *Turk. Online J. Educ. Technol. Spec.* 2015, pp. 536–543.
- [59] D. M. Kadijevich, C. Angeli, and C. Schulte, Teaching Spreadsheets: A TPACK Perspective, in *Improving computer science education*, New York: Routledge, 2013, pp. 132–146.
- [60] C. Angeli and N. Valanides, *Technological Pedagogical Content Knowledge: Exploring, Developing, and Assessing TPACK*. Boston: Springer, 2016.
- [61] J. Hattie, *Visible Learning for Teachers: Maximizing Impact on Learning*. New York, New York: Routledge, 2012.
- [62] OECD-PISA, Pisa 2009 results: Students on line, *PISA*, 2011. doi:10.1787/9789264112995-en

- [63] NJSzT, Nemes Tihamér Nemzetközi Informatikai Tanulmányi Verseny, Programozás. 2020. Available at: <http://nemes.inf.elte.hu>. (letöltés dátuma: 2024 január).
- [64] NJSzT, Nemes Tihamér Nemzetközi Informatikai Tanulmányi Verseny, Alkalmazás Kategória. 2020. Available at: <http://tehetseg.inf.elte.hu/nemesa/index.html>. (letöltés dátuma: 2024 január).
- [65] NJSzT: Nemes Tihamér Országos Alkalmazói Tanulmányi Verseny. <http://tehetseg.inf.elte.hu/nemesa/index.html>. (2023) Letöltés dátuma: 2023. 12. 03.
- [66] NJSzT: Nemes Tihamér Nemzetközi Programozási Verseny. <http://nemes.inf.elte.hu/>. Letöltés dátuma: 2023. 12. 03.
- [67] P. Biro and M. Csernoch, Deep and surface metacognitive processes in non-traditional programming tasks, *2014 5th IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Nov. 2014. doi:10.1109/coginfo-com.2014.7020507
- [68] R. Lister, After the gold rush: toward sustainable scholarship in computing, in *Proceedings of the tenth conference on australasian computing education*, 2008, vol. 78, pp. 3–17.
- [69] M. Csernoch, and P. Biró. Are digital natives spreadsheet natives? In *Proceedings of the EuSpRIG 2019 Conference Spreadsheet Risk Management*, Nice, France, 27–31 May 2019; pp. 1–12.
- [70] M. Prensky, Digital Natives, digital immigrants, *On the Horizon*, vol. 9, no. 5, pp. 1–6, 2001. doi:10.1108/10748120110424816.
- [71] P. A. Kirschner, J. Sweller, and R. E. Clark, Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching, *Educational Psychologist*, vol. 41, no. 2, pp. 75–86, 2006. doi:10.1207/s15326985ep4102_1
- [72] P. A. Kirschner and P. De Bruyckere, The myths of the digital native and the Multitasker, *Teaching and Teacher Education*, vol. 67, pp. 135–142, 2017. doi:10.1016/j.tate.2017.06.001
- [73] K. Sebestyén, Te Is Lehetsz Informatikatanár (?). *INFODIDACT 2014: Informatika Szakmódszertani Konferencia*, 2014. Available at: <http://people.inf.elte.hu/szlavi/InfoDidact14/Manuscripts/SK.pdf> (letöltés dátuma: 2021 május). (letöltés dátuma: 2024 január).
- [74] *Spreadsheet Competency Framework: A Structure for Classifying Spreadsheet Ability in Finance Professionals*. London: ICAEW, 2016.

- [75] Twenty principles for good spreadsheet practice - icaew, ICAEW THOUGHT LEADERSHIP, <https://www.icaew.com/-/media/corporate/files/technical/technology/excel-community/20-principles-of-good-spreadsheet-practice-2018..> (letöltés dátuma: 2024 január)
- [76] R. R. Panko, The Cognitive Science of spreadsheet errors: Why thinking is bad, *2013 46th Hawaii International Conference on System Sciences*, 2013. doi:10.1109/hicss.2013.513.
- [77] R. R. Panko, What we know about spreadsheet errors, *Journal of Organizational and End User Computing*, vol. 10, no. 2, pp. 15–21, May 1998. doi:10.4018/joeuc.1998040102.
- [78] R. R. Panko, Introduction to end user computing: The dark matter (and dark energy) of corporate it use and user (U2) studies Minitrack, *2012 45th Hawaii International Conference on System Sciences*, pp. 1–24, 2012. doi:10.1109/hicss.2012.374.
- [79] Horror stories, European Spreadsheet Risk Interest Group, 2021. <https://eusprig.org/research-info/horror-stories/>. (letöltés dátuma: 2024 január).
- [80] T. Rattenbury, J. M. Hellerstein, J. M. Heer, S. Kandel, and C. Carreras, *Principles of Data Wrangling: Practical Techniques for Data Preparation*. Sebastopol: O’Reilly, 2017.
- [81] K. Rajalingham, D. R. Chadwick, and B. Knight, Classification of spreadsheet errors, <https://eusprig.org/wp-content/uploads/0805.4224.pdf>. (letöltés dátuma: 2024 január).
- [82] R. J. Casimir, Real programmers don’t use spreadsheets, *ACM SIGPLAN Notices*, vol. 27, no. 6, pp. 10–16, 1992. doi:10.1145/130981.130982
- [83] G. Lovászová and J. Hvorecký, On Programming and Spreadsheet Calculations, *Spreadsheets in Education*, vol. 1, no. 1, 2003.
- [84] J. Cunha, *Model-Based Spreadsheet Engineering*. Universidade do Minho. Escola de Engenharia. LAP LAMBERT Academic Publishing, 2012.
- [85] D. D. Hoang, H. Paik, and B. Benatallah, An Analysis of Spreadsheet-Based Services Mashup, *21st Australasian Database Conference*, pp. 141–149, 2010. Available at: <https://dl.acm.org/doi/pdf/10.5555/1862242.1862261>. (letöltés dátuma: 2024 január).
- [86] M. Burnett, J. Atwood, R. Walpole Djang, J. Reichwein, H. Gottfried, and S. Yang, Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm, *Journal of Functional Programming*, vol. 11, no. 2, pp. 155–206, 2001. doi:10.1017/s0956796800003828.

- [87] M. Burnett, End-User Development, in *The Encyclopedia of Human-Computer Interaction*, 2nd ed, C. Scaffidi, Ed. Interaction Design Foundation, 2012.
- [88] P. Hubwieser, Functional modelling in secondary schools using spreadsheets, *Education and Information Technologies*, vol. 9, no. 2, pp. 175–183, 2004. doi:10.1023/b:eait.0000027929.91773.ab
- [89] D. Wakeling, Spreadsheet functional programming, *Journal of Functional Programming*, vol. 17, no. 1, pp. 131–143, 2007. doi:10.1017/s0956796806006186.
- [90] R. Abraham, M. M. Burnett, and M. Erwig, Spreadsheet programming, *Wiley Encyclopedia of Computer Science and Engineering*, pp. 2804–2810, 2009. doi:10.1002/9780470050118.ecse415
- [91] M. Csernoch, P. Biró, and J. Máth, Developing computational thinking skills with algorithm-driven spreadsheeting, *IEEE Access*, vol. 9, pp. 153943–153959, 2021. doi:10.1109/access.2021.3126757
- [92] G. Csapó, M. Csernoch, and K. Abari, Sprego: Case study on the effectiveness of teaching spreadsheet management with schema construction, *Education and Information Technologies*, vol. 25, no. 3, pp. 1585–1605, 2019. doi:10.1007/s10639-019-10024-2
- [93] G. Csapó, K. Sebestyén, M. Csernoch, and K. Abari, Case study: Developing long-term knowledge with Sprego, *Education and Information Technologies*, vol. 26, no. 1, pp. 965–982, 2020. doi:10.1007/s10639-020-10295-0
- [94] J. Walkenbach, *Excel 2010 Bible*. Indianapolis, IN: Wiley, 2010.
- [95] J. Walkenbach, *Excel 2016 Bible*. Indianapolis, IN: Wiley, 2016.
- [96] G. Harvey, *Excel 2010 for Dummies*. John Wiley & Sons, Inc, 2010.
- [97] D. Gookin, *Word 2019 for Dummies*. Hoboken, NJ: John Wiley & Sons Inc., 2019.
- [98] T. K. Nagy, M. Csernoch, and J. Máth, Cross-curricular connections and knowledge-transfer elements in data management, *Acta Polytechnica Hungarica*, vol. 19, no. 1, pp. 133–149, 2022. doi:10.12700/aph.19.1.2022.19.9
- [99] T. Nagy, M. Csernoch, and P. Biró, The comparison of students’ self-assessment, gender, and programming-oriented spreadsheet skills, *Education Sciences*, vol. 11, no. 10, 2021. doi:10.3390/educsci11100590 Article number: 590

- [100] M. Csernoch, From webtables to datatables, in *Proceedings of the EuSpRIG 2019 Conference Spreadsheet Risk Management*, EuSpRIG European Spreadsheet Risks Interest Group, 2019, pp. 1–22
- [101] P. Biró and M. Csernoch, Unplugged tools for building algorithms with Sprego, *Mafalda, Carmo. Education and New Developments*, pp. 401–405, 2017.
- [102] K. Sebestyén and G. Csapó, Visualising Sprego Inequality Problems With 2d Representations, *Turkish Online Journal of Educational Technology*, pp. 1081–1091, 2018.
- [103] N. Dienes and Á. Gulácsi, Játékosított Sprego programozás, *TDK dolgozat*, 2021.
- [104] M. Csernoch, Beaufort-Skála Digitális Feldolgozása: Online tanulás, ProSuli, <https://hipersuli.webuni.hu/kepzes/beaufort-skala-digitalis-feldolgozasa> (letöltés dátuma: 2023 december).
- [105] R. Low, P. Jin, and J. Sweller, Cognitive load theory, attentional processes and optimized learning outcomes in a digital environment, *Human Attention in Digital Environments*, pp. 93–113, 2011. doi:10.1017/cbo9780511974519.004
- [106] L. F. Malmi, J. Sheard, P. Kinnunen, Simon, and J. Sinclair, Computing Education Theories: What Are They and How Are They Used?, *Proceedings of the 2019 ACM Conference on International Computing Education Research*, pp. 187–197, 2019. doi:10.1145/3291279.3339409
- [107] L. F. Malmi, J. Sheard, P. Kinnunen, Simon, and J. Sinclair, Development and use of domain-specific learning theories, models, and instruments in computing education, *ACM Transactions on Computing Education*, vol. 23, no. 1, pp. 1–48, 2022. doi:10.1145/3530221
- [108] W.-W. Jiang, W. Chen, and Y.-C. Chen, Important computer competencies for the nursing profession, *Journal of Nursing Research*, vol. 12, no. 3, pp. 213–226, 2004. doi:10.1097/01.jnr.0000387505.98877.6d
- [109] R. Catrambone and J. M. Carroll, Learning a word processing system with training wheels and guided exploration, *ACM SIGCHI Bulletin*, vol. 17, no. SI, pp. 169–174, 1986. doi:10.1145/30851.275625
- [110] K. Sebestyén, G. Csapó, M. Csernoch, and B. Aradi, Error recognition model: High-mathability end-user text management, *Acta Polytechnica Hungarica*, vol. 19, no. 1, pp. 151–170, 2022. doi:10.12700/aph.19.1.2022.1.10

- [111] J. M. Carroll, R. L. Mack, C. H. Lewis, N. L. Grischkowsky, and S. R. Robertson, Exploring exploring a word processor, *Human–Computer Interaction*, vol. 1, no. 3, pp. 283–307, 1985. doi:10.1207/s15327051hci0103_3
- [112] J. M. Carroll and M. B. Rosson, *Paradox of the Active User*. San Jose etc.: IBM Thomas J. Watson Research Division, 1986.
- [113] J. M. Carroll, *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Cambridge, MA: MIT Press, 1987.
- [114] *WordPerfect Workbook: For IBM Personal Computers and PC Networks, Version 5.1*. Orem, UT: WordPerfect Corp., 1989.
- [115] S. Kinn and T. Siann, Introduction to word processing, *Computers and Clinical Audit*, pp. 91–99, 1998. doi:10.1007/978-1-4899-6639-1_8
- [116] A. L. Kamouri, J. Kamouri, and K. H. Smith, Training by exploration: Facilitating the transfer of procedural knowledge through analogical reasoning, *International Journal of Man-Machine Studies*, vol. 24, no. 2, pp. 171–192, 1986. doi:10.1016/s0020-7373(86)80047-5
- [117] K. Sebestyén, G. Csapó, and M. Csernoch, The effectiveness of the web-table-datatable conversion approach, *Annales Mathematicae et Informaticae*, vol. 56, pp. 109–121, 2023. doi:10.33039/ami.2022.12.010
- [118] I. Tsimperidis and G. Chrysomalidis, Timeless and Multi-Platform Teaching of Word Processors, *Journal of Modern Education Review*, vol. 11, pp. 1287–1293, 2021.
- [119] I. M. Joos, D. M. Wolf, and R. Nelson, Introduction to Word Processing, Chapter 5, in *Introduction to computers for Healthcare professionals*, Burlington, MA: Jones & Bartlett Learning, 2021
- [120] J. Lambert, Microsoft Word 2016 Step By Step, Microsoft Word 2016 Step By Step, Microsoft Press Store,
- [121] A. Philo and M. Angstadt, *MICROSOFT WORD 2016 STEP-BY-STEP GUIDE*. Norristown, PA: MC-NPL Computer Lab, 2020.
- [122] L. A. Bucki, *Microsoft Word 2013 Bible*. Wiley, 2013.
- [123] F. Wempfen, *Microsoft PowerPoint 2013 Bible*. Indianapolis, IN: John Wiley & Sons, Inc., 2013.
- [124] E. Smyrnova-Trybulska, *Effective Development of Teachers' Skills in the Area of ICT and e-Learning: Monograph*. Katowice: Studio Noa, 2017.
- [125] L. Bucki, J. Walkenbach, F. Wempfen, D. Kusleika, and M. Alexander, Creating Documents with Word 2013, in *Microsoft Office 2013 bible*, Indianapolis: Wiley, 2013, pp. 101–398

- [126] A. Lénárd, G. Sarbó, M. Tarné Éder, and N. Turzó-Sovák, *Digitális Kultúra 3*. Budapest, Hungary: Oktatási Hivatal, Budapest, 2022.
- [127] A. Lénárd, N. Turzó-Sovák, M. Tarné Éder, and G. Sarbó, *Digitális Kultúra 4*. Budapest, Hungary: Oktatási Hivatal, Budapest, 2023.
- [128]] A. Lénárd, A. Abonyi-Tóth, N. Turzó-Sovák, and P. Varga, *Digitális Kultúra 5*. Budapest, Hungary: Oktatási Hivatal, Budapest, 2020.
- [129] A. Abonyi-Tóth, C. Farkas, N. Turzó-Sovák, and P. Varga, *Digitális Kultúra 6*. Budapest, Hungary: Oktatási Hivatal, Budapest, 2021.
- [130] A. Abonyi-Tóth, C. Farkas, and P. Varga, *Digitális Kultúra 7*. Budapest, Hungary: Oktatási Hivatal, Budapest, 2022.
- [131] A. Abonyi-Tóth, C. Farkas, and P. Varga, *Digitális Kultúra 8*. Budapest, Hungary: Oktatási Hivatal, 2023.
- [132] P. Varga, K. Jeneiné Horváth, Z. Reményi, Cs. Farkas, I. Takács, G. Siegler, Abonyi-Tóth: *Digitális Kultúra 9*. Budapest, Hungary: Oktatási Hivatal, Budapest, 2020.
- [133] A. Abonyi-Tóth, Cs. Farkas, Z. Reményi, and K. Jeneiné Horváth, *Digitális Kultúra 10*. Budapest, Hungary: Oktatási Hivatal, 2021.
- [134] A. Abonyi-Tóth, Cs. Farkas, Zs. Fodor, K. Jeneiné Horváth, Z. Reményi, G. Siegler, P. Varga, *Digitális Kultúra 11*. Budapest, Hungary: Oktatási Hivatal, 2022.
- [135] Qualification component and record of learner achievement. Word processing software skills E3 (L/616/1318). Jul. 13, 2017 https://www.aim-group.org.uk/clientfiles/files/units/learner_achievements/Word%20Processing%20Software%20Skills%20E3%20CV2%20AIM%20Awards%20Component%20V1-1.docx. (letöltés dátuma: 2024 január).
- [136] Building Institutional Capacity from the Heart of Africa, in *BEST PRACTICE AWARDS 2016 ECDL/ICDL*, ECDL Foundation, 2016, pp. 31–32
- [137] Word processing, Word Processing Unit 2: Manipulating Text With Word Lesson 4: Basic Text Formatting, <http://www.webmedia-workshop.com/Classes/cs100/mod.wp/unit.2/lesson.4/paraformat.html>. (letöltés dátuma: 2024 január).
- [138] R. McNally and S. O’Neill, Word Processing Uses and Features, in *For students of Monaghan Institute*, Dec. 29, 2023 https://miitnotes.weebly.com/uploads/9/8/4/1/9841023/lo1_word_processing_uses_and_features.pdf. (letöltés dátuma: 2024 január).

- [139] E. Goodwin, Word Processing basics, in *Business Technology Applications*, Dec. 29, 2023 https://slideplayer.com/slide/6126983/#google_vignette. (letöltés dátuma: 2024 január).
- [140] A. Prestage, Word Processing, *Encyclopedia of Information Systems*. Academic Press, Amsterdam, 2003 <https://www.sciencedirect.com/topics/computer-science/word-processing>. (letöltés dátuma: 2024 január).
- [141] M. Ben-Ari, Bricolage Forever!, in *Psychology of Programming Interest Group*, 1999
- [142] M. Ben-Ari and T. Yeshno, Conceptual models of software artifacts, *Interacting with Computers*, vol. 18, no. 6, pp. 1336–1350, 2006. doi:10.1016/j.intcom.2006.03.005
- [143] R. Maurer, *One Small Step Can Change Your Life: The Kaizen Way*. New York, NY: Workman Pub., 2014.
- [144] R. Maurer, *The Spirit of Kaizen: Creating Lasting Excellence One Small Step at a Time*. New York: McGraw-Hill, 2013.
- [145] T. Narusawa and J. Shook, *Kaizen Express: Fundamentals for Your Lean Journey*. Cambridge, MA: Lean Enterprise Institute, 2009.
- [146] J. K. Liker, *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. New York, New York: McGraw-Hill, 2004.
- [147] S. R. Dodlapati, P. Lakkaraju, N. Tulluru, and Z. Zeng, Non Printable & special characters: Problems and how to overcome them, <https://www.lexjansen.com/nesug/nesug10/ff/ff04.pdf> (letöltés dátuma: 2023 január).
- [148] C. K. Kenyon, Show/hide non-printing formatting characters, <https://www.addbalance.com/word/nonprinting.htm> (letöltés dátuma: 2023 január).
- [149] S. Barnhill, Word's nonprinting formatting marks, Nonprinting Formatting Marks, <http://wordfaqs.ssarnhill.com/NonprintChars.htm> (letöltés dátuma: 2023 január).
- [150] E. Curts, How to show non-printing characters in google docs, LaptrinhX, <https://laptrinhx.com/how-to-show-non-printing-characters-in-google-docs-989631243/>
- [151] P. Virágvolgyi, *A Tipográfia Mestersége: Számítógéppel*. Budapest: Osiris, 2004.
- [152] D. Jury, *What Is Typography?* Brighton: Ivy Press, 2006.
- [153] D. Jury, *About Face: Reviving the Rules of Typography*. Brighton: Rotovision, 2004.

- [154] G. Reynolds, *Presentation Zen: Simple Ideas on Presentation Design and Delivery*. Berkeley: New Riders, 2008.
- [155] ECDL base: Word processing training course & certification information, ECDL Training, <http://www.ecdlonline.uk/ecdl-courses/certification/ecdl-basic-word-processing.html> (letöltés dátuma: 2023 január).
- [156] ECDL, Foundation. Best Practice Awards; Digital Literacy AG: Basel, Switzerland, 2015
- [157] M. Caspersen, I. Diethelm, J. Gal-Ezer, A. McGettrick, E. Nardelli, D. Passey, B. Rován, M. Webb, Informatics Reference Framework for School, informatics for all, <https://www.informaticsforall.org/wp-content/uploads/2022/03/Informatics-Reference-Framework-for-School-release-February-2022.pdf> (letöltés dátuma: 2024 január).
- [158] A. Smalley, *Four Types of Problems: From Reactive Troubleshooting to Creative Innovation*. Cambridge, MA, US: Lean Enterprise Institute, 2018.
- [159] T. Brown, *Change by Design, Revised and Updated: How Design Thinking Transforms Organizations and Inspires Innovation*. New York: HarperCollins Publishers, 2019.
- [160] M. Csernoch, J. Máth, and T. Nagy, The interpretation of graphical information in word processing, *Entropy*, vol. 24, no. 10, p. 1492, 2022. doi:10.3390/e24101492
- [161] M. Csernoch, Teaching word processing – the practice, *Teaching Mathematics and Computer Science*, vol. 8, no. 2, pp. 247–262, 2010. doi:10.5485/tmcs.2010.0252
- [162] M. Csernoch, Teaching word processing - the theory behind, *Teaching Mathematics and Computer Science*, vol. 7, no. 1, pp. 119–137, 2009. doi:10.5485/tmcs.2009.0212
- [163] M. Stuermer, G. Abu-Tayeh, and T. Myrach, Digital Sustainability: Basic conditions for sustainable digital artifacts and their ecosystems. *Sustainability Science*, vol. 12, no. 2, pp. 247–262, 2016. doi:10.1007/s11625-016-0412-2
- [164] P. Messerli and E. Murniningtyas, Eds., *The Future Is Now: Science for Achieving Sustainable Development: Global Sustainable Development Report 2019*. UN, Dept. of Economic and Social Affairs, 2019.
- [165] L. Jensen, Ed., *Sustainable Development Goals Report 2022*. New York, NY: United Nations, 2022.

- [166] IBM Cloud Education, What is green computing?, <https://www.ibm.com/cloud/blog/green-computing> (letöltés dátuma: 2024 január).
- [167] V. Forti, C. P. Baldé, R. Kuehr, and G. Bel, *The Global E-Waste Monitor 2020. Quantities, Flows, and the Circular Economy Potential*. UNU/UNITAR and ITU, 2020. https://ewastemonitor.info/wp-content/uploads/2020/11/GEM_2020_def_july1_low.pdf. (letöltés dátuma: 2024 január).
- [168] O. Hatamleh and G. Tilesch, *BetweenBrains; Taking Back Our AI Future*. GTPublishDrive, 2020.
- [169] J. Tucker, Facing the challenge of digital sustainability as humanities researchers, *Journal of the British Academy*, vol. 10, pp. 93–120, 2022. doi:10.5871/jba/010.093
- [170] M. Csernoch and E. Dani, Do you speak, write and think in Informatics?, *Acta Polytechnica Hungarica*, vol. 19, no. 1, pp. 113–131, 2022. doi:10.12700/aph.19.1.2022.1.8
- [171] European spreadsheet Risk Interest Group, <https://eusprig.org/> (letöltés dátuma: 2024 január 4).
- [172] C. Wolfram, Evidence: Let’s promote not stifle innovation in Education, Conrad Wolfram, <https://www.conradwolfram.com/writings/2015/5/21/role-of-evidence-in-education-innovation> (letöltés dátuma: 2024 január).
- [173] OECD – PISA 2018 Results (Volume I). What Students Know and Can Do. Dec., 2019. Available at: <https://read.oecd.org/10.1787/5f07c754-en?format=pdf>. (letöltés dátuma: 2024 január).
- [174] OECD – PISA 2022 Results (Volume I), The State of Learning and Equity in Education. Dec., 2023. doi:10.1787/53f23881-en
- [175] T. K. Nagy, The Paradox of the Hungarian Frame Curricula in Informatics, *Turkish Online Journal of Educational Technology*, pp. 910–922, 2018.
- [176] J. B. Biggs and K. F. Collis, Origin and description of the solo taxonomy, *Evaluating the Quality of Learning*, pp. 17–31, 1982. doi:10.1016/b978-0-12-097552-5.50007-7
- [177] R. Lister, B. Simon, E. Thompson, J. L. Whalley, and C. Prasad, Not seeing the forest for the trees, *ACM SIGCSE Bulletin*, vol. 38, no. 3, pp. 118–122, 2006. doi:10.1145/1140123.1140157
- [178] J. Sheard, A. Carbone, R. Lister, B. Simon, E. Thompson, J. Whalley, Going solo to assess novice programmers, *ACM SIGCSE Bulletin*, vol. 40, no. 3, pp. 209–213, 2008. doi:10.1145/1597849.1384328

- [179] D. Gross, F. Akaiva, K. Nordquist, *Succeeding in Business with Microsoft Excel 2013: A Problem-Solving Approach*. Cengage Learning (New Perspectives). Stamford, CT: Cengage Learning, 2014.
- [180] Top 250 youtubers sorted by SB score - socialblade youtube stats ..., <https://socialblade.com/youtube/top/250>.
- [181] MTA NYTI, A Magyar Helyesírás Szabályai (Akh.), A magyar helyesírás szabályai (AkH.), 2012. <https://helyesiras.mta.hu/helyesiras/default/akh12> (letöltés dátuma: 2024 január).
- [182] Qualifications Framework. Description of the Eight EQF Levels. Available at: <https://europa.eu/europass/en/description-eight-eqf-levels>. (letöltés dátuma: 2024 január)
- [183] S. Carretero, R. Vuorikari, and Y. Punie, *DigComp 2.1 the Digital Competence Framework for Citizens with Eight Proficiency Levels and Examples of Use*. Luxembourg: Publications Office, 2017.
- [184] C. E. Shannon, Prediction and entropy of printed English, *Bell System Technical Journal*, vol. 30, no. 1, pp. 50–64, 1951. doi:10.1002/j.1538-7305.1951.tb01366.x
- [185] C. Shannon, A Mathematical Theory of Communication, *The Bell System Technical Journal*, pp. 623–656, 1948.
- [186] C. Shannon, A mathematical theory of communication, *The Bell System Technical Journal*, pp. 379–423, 1948.
- [187] R. M. Gray, *Entropy and Information Theory*. Springer, 2011.
- [188] Information Entropy (Foundations of Information Theory: Part 2), Matthew Bernstein. Available at: <https://mbernste.github.io/posts/entropy/>. (letöltés dátuma: 2024 január).
- [189] C. O’Neil and R. Schutt, Entropy, in *Doing data science*, Sebastopol (CA): O’Reilly Media, 2014, pp. 186–187
- [190] W. Kinsner, Is entropy suitable to characterize data and signals for Cognitive Informatics?, *Novel Approaches in Cognitive Informatics and Natural Intelligence*, pp. 28–51. doi:10.4018/978-1-60566-170-4.ch002
- [191] A. Rényi, On measures of information and entropy, *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 547–561, 1961.
- [192] E. Tóthné Laufer, *Az Információ- És Kódelmélet Alapjai*. Budapest: Óbudai Egyetem, 2019.

- [193] K. Nagy and M. Csernoch, Pre-testing erroneous text-based documents: Logging end-user activities, *Frontiers in Education*, vol. 7, Mar. 2023. doi:10.3389/educ.2022.958635
- [194] M. Csernoch, Methodological Questions of Teaching Word Processing, in *3rd International Conference on Applied Informatics*, 1997, pp. 375–382
- [195] M. Csernoch and P. Biró, Wasting Human and Computer Resources, *International Journal of Social, Education, Economics and Management Engineering*, vol. 9, no. 2, pp. 573–581, 2015.
- [196] M. Csernoch, K. Nagy, and T. Nagy, The entropy of digital texts—the mathematical background of correctness, *Entropy*, vol. 25, no. 2, p. 302, 2023. doi:10.3390/e25020302
- [197] K. Hankiewicz and M. Butlewski, Efficiency in performing basic tasks using word processing programs by the elderly as a measure of the ergonomic quality of software, *Lecture Notes in Computer Science*, pp. 481–488, 2014. doi:10.1007/978-3-319-07233-3_44
- [198] E. Osztíán, Interaktív, Többérvékszerves Elektronikus Oktatási Környezet a Számítógépes Gondolkodás Fejlesztéséhez. Ph.D. Dissertation, University of Debrecen, Debrecen, Hungary, 2021. Available at: <https://dea.lib.unideb.hu/dea/handle/2437/302091> (letöltés dátuma: 2024 január).
- [199] D. Kruger and D. Dunning, Unskilled and unaware of it: How difficulties in recognizing one’s own incompetence lead to inflated self-assessments., *Journal of Personality and Social Psychology*, vol. 77, no. 6, pp. 1121–1134, 1999. doi:10.1037/0022-3514.77.6.1121
- [200] S. Staub and R. Kaynak, Is an unskilled really unaware of it?, *Procedia - Social and Behavioral Sciences*, vol. 150, pp. 899–907, 2014. doi:10.1016/j.sbspro.2014.09.099

15. Függelék

1. függelék: A mini kompetenciateszt első oldala

Lány Fiú Osztály: Életkor:

Alapinformációk

- G1. Milyen típusú iskolába jársz? (egyet jelölhetsz meg)
- Általános iskola Szakgimnázium Gimnázium AMI
 Szakközépiskola Szakiskola egyéb:.....
- G2. Informatika órán kívül, más tanórákon HETENTE mennyi időt szoktál SZÁMÍTÓGÉP (asztali számítógép, laptop) használatával tölteni? Húzd alá a választ!
- kevesebbet kb. 1 óra kb. 2 óra kb. 3 óra kb. 4 óra kb. 5 óra többet
- G3. Informatika órán kívül, más tanórákon HETENTE mennyi időt szoktál MOBIL eszközök (táblagép (tablet), okostelefon, e-book olvasó) használatával tölteni? Húzd alá a választ!
- kevesebbet kb. 1 óra kb. 2 óra kb. 3 óra kb. 4 óra kb. 5 óra többet
- G4. Összesen mennyi időt szoktál SZÁMÍTÓGÉP (asztali számítógép, laptop) használatával tölteni NAPONTA? Húzd alá a választ!
- kevesebbet kb. 2 óra kb. 4 óra kb. 6 óra kb. 8 óra kb. 10 óra többet
- G5. Összesen mennyi időt szoktál MOBIL eszközök (táblagép (tablet), okostelefon, e-book olvasó) használatával tölteni NAPONTA? Húzd alá a választ!
- kevesebbet kb. 2 óra kb. 4 óra kb. 6 óra kb. 8 óra kb. 10 óra többet
- G6. Heti hány órában tanuld/tanultad az informatikát az iskolában?

1. oszt.	2. oszt.	3. oszt.	4. oszt.	5. oszt.	6. oszt.	7. oszt.	8. oszt.	NYEK	9. oszt.	10. oszt.

- G7. Mivel foglalkoztok informatika órán? (többet is megjelölhetsz)
- ECDL feladatok megoldása. Táblázatokba adatok gépelése.
 Önállóan dolgozunk, feladatsor alapján. Algoritmusok leírása, rajzolása.
 Érettségi feladatok megoldása. Keresés eredményének értelmezése.
 Programozás. Szöveg gépelése.
 Tankönyvi, munkafüzeti feladatok. Adatbázis gépelése.
 Prezentációtervezés, diaminta. Játék, szabad foglalkozás.
 Algoritmusok megfogalmazása. Hibafelismerés szöveges dokumentumokban.
 Források megnevezése. Elektronikus könyvtárban keresés.
 Fájlkonverziók. Tömítési módok és eljárások megismerése.
 Speciális keresés böngésző programokban. Elektronikus adatbázisban keresés.
 Számítógép részeinek működési elve. Függvények egymásba ágyazása.
 Dokumentumtervezés. Eredmények helyességének ellenőrzése.
 Egyéb:

2. függelék: A mini kompetenciateszt második oldala

G8. Milyen nyelven tanultok/ tanultatok programozni? (többet is megjelölhatsz)

- | | | |
|----------------------------------|---|--|
| <input type="checkbox"/> C | <input type="checkbox"/> C++ | <input type="checkbox"/> C# |
| <input type="checkbox"/> Logo | <input type="checkbox"/> Python | <input type="checkbox"/> Java |
| <input type="checkbox"/> Scratch | <input type="checkbox"/> robot nyelvek | <input type="checkbox"/> blokk nyelvek |
| <input type="checkbox"/> PHP | <input type="checkbox"/> HTML | <input type="checkbox"/> CSS |
| <input type="checkbox"/> Pascal | <input type="checkbox"/> Nem tanultunk programozni. | <input type="checkbox"/> egyéb:..... |

G9. Milyen tankönyvek/ munkafüzetek van informatikából? (többet is megjelölhatsz)

- Nincs tankönyvünk/ munkafüzetünk. Egyéb:.....
- Nem tudom milyen könyvet/ munkafüzetet használunk.
- Informatika munkatankönyv 7-8. osztály (Műszaki Könyvkiadó)
- Informatika 9-10. (Műszaki Könyvkiadó)
- Informatika 7. (Mozaik Kiadó) Informatika 7. mf. (Mozaik Kiadó)
- Informatika 8. (Mozaik Kiadó) Informatika 8. mf. (Mozaik Kiadó)
- Informatika középiskolásoknak (Mozaik Kiadó) Informatika középiskolásoknak mf. (Mozaik Kiadó)
- Informatika 9-10. a gimnáziumok számára (Eszterházy Károly Egyetem)
- Nyolcadikos informatika (Pedellus Tankönyvkiadó)
- Informatikai ismeretek a középiskolák részére (Jedlik Oktatási Stúdió)
- Informatikai feladatgyűjtemény (Jedlik Oktatási Stúdió)

G10. Amennyiben van, milyen gyakran használjátok a tankönyvet/ munkafüzetet informatika órán? Karikázd be a megfelelő választ! (0=soha, 5=minden órán)

0 1 2 3 4 5

G11. Hogyan értékelnéd ismereteidet az alábbi témakörökben? (0=egáltalán nem tudom, 5=nagyon jól tudom)

	Mennyire ismered a témakört?					Iskolában tanultad?		
	Karikázd be a megfelelő számot!					tanultam	nem tanultam	
fájlkezelés	0	1	2	3	4	5	<input type="checkbox"/>	<input type="checkbox"/>
szövegkezelés	0	1	2	3	4	5	<input type="checkbox"/>	<input type="checkbox"/>
táblázatkezelés	0	1	2	3	4	5	<input type="checkbox"/>	<input type="checkbox"/>
adatbáziskezelés	0	1	2	3	4	5	<input type="checkbox"/>	<input type="checkbox"/>
algoritmizálás, programozás	0	1	2	3	4	5	<input type="checkbox"/>	<input type="checkbox"/>
források kezelése, hitelessége	0	1	2	3	4	5	<input type="checkbox"/>	<input type="checkbox"/>

G12. Sorold fel azokat a táblázatkezelő függvényeket (maximum 15), amelyekről úgy gondolod, hogy ismeretükre feltétlenül szükség van!

1.	4.	7.	10.	13.
2.	5.	8.	11.	14.
3.	6.	9.	12.	15.

3. függelék: A mini kompetenciateszt harmadik oldala

Informatikai ismeretek

F1. Karikázd be a nem perifériás eszközök betűjelét! A perifériás eszközök esetén jelöld meg a képek alatt, hogy bemeneti (B), kimeneti (K) vagy be- és kimeneti (BK)!



Z B K BK



Y B K BK



X B K BK



V B K BK



U B K BK

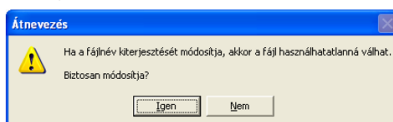


T B K BK

F2. Mennyire ismered ezen eszközök működési elvét? (0=egyáltalán nem ismerem, 5=nagyon jól ismerem)
Karikázd be a megfelelő számot!

Z	0	1	2	3	4	5
Y	0	1	2	3	4	5
X	0	1	2	3	4	5
V	0	1	2	3	4	5
U	0	1	2	3	4	5
T	0	1	2	3	4	5

F3. Mit jelent az alábbi üzenet adatfájlok esetén?



- Ha a kiterjesztés változik a gép képtelen lesz felismerni a fájlt.
- Kiterjesztés módosításnál adatok elveszhetnek.
- Megváltozik a társítás és a fájl használhatatlan lesz.
- Megváltozik a társítás, de a fájl továbbra is használható lesz.
- Ha rossz formátumra írjuk át a fájlt, akkor az megsérül és használhatatlan lesz.
- Az új kiterjesztés más kódolása miatt a tartalom a felhasználó által olvashatatlan lesz.

F4. Mit csinál a következő algoritmus?

A, B, C

Ha $A^2=B^2+C^2$ vagy $B^2=A^2+C^2$ vagy $C^2=A^2+B^2$

akkor "igen"

egyébként "nem"















Ha vége.

.....

.....

4. függelék: A mini kompetenciateszt negyedik oldala

- F5. Barbara két pecsétet kapott. Az egyik virágot, a másik napot nyomtat. Elgondolta, hogy tudná a nevét csak a virágokkal és napokkal lepecsételni. A különböző betűket virágok és napok különböző sorozatával határozta meg.

Betű	B	A	R	E	Y
Sorozat		 	  	   	   

A saját nevét, „Barbara” tehát így pecsételte le:



Ezután lepecsételte angol barátja nevét is:



Hogy hívják a barátját?

- F6. Mi történik, ha duplán kattintunk egy dokumentum fájlra (például: zz.jpg, zz.html, zz.ods, zz.xls)?

.....

- F7. Melyik számozás helyes? Karikázd be a helyes válaszokat! (többet is megjelölhetsz)

- A 1.A-növény-részei
- B 3.Termodinamikus-kölcsönhatás
- C b.)-Gabonafélék
- D Petőfi-szerelmi-költészete
- E 4.-Magyarázza-el-a-ciklusok-működését!

- F8. Találd meg a helyesírási hibákat a szövegben! Karikázd be ezeket a mintán!
 Hány hibát találtál? Karikázd be a megfelelő számot!

A gyermekek kitarással a repülő méhecskét utánozzák, méhecske hangját hangoztatják: „z” szaladgálnak, ha leoltjuk a villanyt, leszállnak virágpont gyűjteni (leguggolnak). Felkapcsoljuk a villanyt, folytatódik tovább a játék.

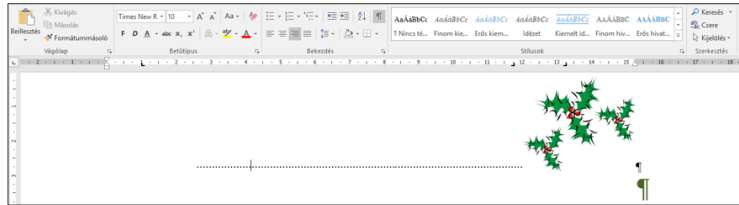
0 1 2 3 4 5 6 több

- F9. Helyesek-e a mintán (F8 feladat) a nyitó és a záró zárójelekhez tartozó belső szóközők?

- A nyitó zárójelnél helyes a belső szóköz. A záró zárójelnél nem, mert nincs belső szóköz.
- Egyik zárójelnél sem helyesek a belső szóközők.
- A záró zárójelnél helyes, hogy nincs belső szóköz. A nyitó zárójelnél nem, mert van belső szóköz.
- Mindkét zárójelnél helyesek a szóközők.

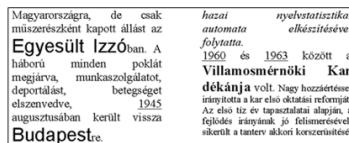
5. függelék: A mini kompetenciateszt ötödik oldala

F10. Az aktuális bekezdésben milyen bekezdés-formázás olvasható le a mintáról?



- Nem látszik, hogy melyik az aktuális bekezdés, mert nincs szöveg a mintán.
- Jobbra igazítás és 3 tabulátor.
- Szóközökkel a kép jobbra igazítása.
- Bal behúzás, ami onnan látszik, hogy a szóközök nem a margónál kezdődnek.

F11. Mi okozza a nagy lyukakat („utcákat”) a szavak között?



- Tabulátor.
- Sorkizárt igazítás.
- Sok szóköz.
- A szövegszerkesztő program hibája, mert széthúzta a szöveget.

F12. Dönts el az alábbi szövegrészekről, hogy melyik lehetett az eredeti dokumentum!

A

B

- A, mert az az interneten található.
- B, mert az egy Word dokumentum.
- Nem tudjuk megmondani, mert egyik sem tartalmaz forrásmegjelölést.
- Mindkettő, mert a mondókák szájhagyomány útján terjedhetnek.

F13. Olvasd el a szövegrészletet (F12 feladat)! Milyen tartalmi (szemantikai) hibát találsz a szövegrészletben?

- Kézi számozás.
- Minden sor végén extra szóköz.
- Nem-törhető szóköz a mondóka utolsó sorában.
- A név nincs aláhúzva szaggatott vonallal.

6. függelék: A mini kompetenciateszt hatodik oldala

- F14. Hogyan egészítenéd ki az A1 cella képletét úgy, hogy ne kapijunk hibáüzenetet (szintaktikailag helyes legyen)? Javítsd ki a képleteket az ábrán, majd írd az ábra alá, hogy mely tartományokra hivatkozol!

	A
1	SZUM(A2 A20

	A
1	SZUM(A2 A20

- F15. Írd le, hogy milyen sorrendben hajtja végre az alábbi képletben szereplő műveleteket a táblázatkezelő!

=HA(ÁTLAG(D2:D58)-50<A5;"a";"")

1. lépés	
2. lépés	
3. lépés	
4. lépés	
5. lépés	

- F16. Hogyan tudnál egy táblázatkezelő dokumentumot szövegfájlá (.csv vagy .txt) alakítani? (egyet jelölhatsz meg)

- Konvertálás. Importálás. Exportálás.
 Társítás. Átírjuk a kiterjesztést. Mentés másként, átírjuk a kiterjesztést.
 Google-ben rákeresek. Online konverter. Mentés másként, kiválasztjuk az új típust.

- F17. A következő kérdések az alábbi MAGYAR nyelvű táblázatra vonatkoznak. Milyen típusú adatok szerepelnek az egyes oszlopokban a 2. sortól kezdődően? (többet is megjelölhatsz)

	A	B	C	D
1	Felhasználó	Feltöltés	Feliratkozó	Megtekintés
2	VamosART	484	1,107,555	226,195,766
3	Videómánia	338	833,23	254,545,702
4	PamKutya	120	809,866	223,441,355
5	LetsGoMartin	176	725,638	162,798,559
6	TheVR	1,062	592,675	213,550,948
7	luckeY	1,183	561,13	150,341,428
8	Peter Gergely	100	548,241	79,713,757
9	Scribble Netty	159	546,049	74,234,471
248	Szilvaglam	87	61,899	3,918,538
249	rance flow	524	61,863	53,275,385
250	KIS GRÓFO (official)	9	61,65	30,712,031
251	KODIAK	736	61,467	14,599,194

	Egész szám	Valós szám	Szöveg	Logikai	Dátum	Egyik sem
A oszlop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B oszlop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C oszlop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D oszlop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Hány adatsort (rekordot) tartalmaz a táblázat?

Írd le a B oszlopban látható LEGNAGYOBB számot!

Írd le a B oszlopban látható LEGKISEBB számot!

Mit csinál a képlet? {=SZUM(HA(BAL(A2:A251)="L";1))}

16. Publikációs jegyzék

Idegen nyelvű tudományos közlemények nemzetközi folyóiratban

M. Csernoch, K. Nagy, and T. Nagy, The entropy of digital texts—the mathematical background of correctness, *Entropy*, vol. 25, no. 2, p. 302, 2023. doi:10.3390/e25020302

M. Csernoch, J. Máth, and T. Nagy, The interpretation of graphical information in word processing, *Entropy*, vol. 24, no. 10, p. 1492, 2022. doi:10.3390/e24101492

T. K. Nagy, M. Csernoch, and J. Máth, Cross-curricular connections and knowledge-transfer elements in data management, *Acta Polytechnica Hungarica*, vol. 19, no. 1, pp. 133–149, 2022. doi:10.12700/aph.19.1.2022.19.9

T. Nagy, M. Csernoch, and P. Biró, The comparison of students’ self-assessment, gender, and programming-oriented spreadsheet skills, *Education Sciences*, vol. 11, no. 10, 2021. doi:10.3390/educsci11100590 Article number: 590

Idegen nyelvű tudományos nemzetközi konferencia közlemények

T. K. Nagy, The Paradox of the Hungarian Frame Curricula in Informatics, *Turkish Online Journal of Educational Technology*, pp. 910–922, 2018.

Magyar nyelvű tudományos közlemények magyar folyóiratban

K. Sebestyén, T. K. Nagy, M. Csernoch, Tanulók szövegkezelés ismeretei az általános iskola elvégzése után, *Információs Társadalom*, 2020. https://infstars.infonia.hu/pub/PUB_2023-May-03-13-35-19_39568.pdf.

Magyar nyelvű tudományos konferencia közlemények

T. Nagy, M. Csernoch, Számítógépes problémamegoldás mérése az informatikaórán. *INFODIDACT 2018: Informatika Szakmódszertani Konferencia (ed. Szlávi Péter, Zsakó László)*, <https://people.inf.elte.hu/szlavi/InfoDidact18/Manuscripts/NTKCsM.pdf>

T. Nagy, Az informatika kerettanterv elemzése, *3th International Interdisciplinary Conference 2018*. Available at: https://detep.unideb.hu/sites/default/files/upload_documents/kotet_interdisz_3.pdf

Beküldött, közlésre elfogadott tudományos közlemények

M. Csernoch, T. Nagy, K. Nagy, J. Csernoch and C. Hannusch, Human-centered digital sustainability: handling enumerated lists in digital texts. *IEEE Access*.

T. Nagy, M. Csernoch, One tool does not make a difference – Spreadsheet competences in the function of time spent on digital tools, IEEE Education.

Szakmai konferenciák, előadástartások idegen nyelven

Időpont	Előadás adatai
2018. július	The Paradox of the Hungarian Frame Curricula in Informatics, International Conference on New Horizons in Education nemzetközi tudományos konferencia, Párizs

Szakmai konferenciák, előadástartások magyar nyelven

Időpont	Előadás adatai
2020. december	Programozás tanulás támogatása SPREGO-val, ADA 2020
2018. november	Számítógépes problémamegoldás mérése az informatika órán, InfoDidact Konferencia, Zamárdi
2019. május	7–10. évfolyamos tanulók tudástranszfer alapú számítógépes gondolkodásának elemzése, ADA 2019 tudományos konferencia, Debrecen
2019. április	7–10. évfolyamos tanulók tudástranszfer alapú számítógépes gondolkodásának elemzése reprezentatív felmérés alapján, XXXIV. OTDK Konferencia Tanulás- és Tanításmódszertan, Esztergom
2019. április	Digitális Témahét: Informatika didaktika módszertani bemutató szakórák, Szlovák Tanítási Nyelvű Óvoda, Általános Iskola, Gimnázium és Diákotthon, Budapest
2018. október	7–10. évfolyamos tanulók tudástranszfer alapú számítógépes gondolkodásának elemzése reprezentatív felmérés alapján, TDK 2018, Debrecen
2018. november	Számítógépes problémamegoldás mérése az informatika órán, InfoDidact Konferencia, Zamárdi
2018. május	Az informatika kerettanterv elemzése, 3th International Interdisciplinary Conference 2018, Debrecen.