

Sprego helye az informatika tantervekben

Csernoch Mária, Biró Piroska

{csernoch.maria, biro.piroska}@inf.unideb.hu

DE IK

Absztrakt. Sprego – Spreadsheet Lego – egy olyan programozási eszköz, amely alkalmazható a táblázatkezelés tanítására, ugyanakkor szolgálhat kezdő programozási nyelvként is. A nyelv egyszerűségéből következően a hangsúly a koncepció és az algoritmus megfogalmazásán van, szemben a kódolási részletekkel, tehát kezdőknek és végfelhasználóknak egyaránt kényelmes eszköz. Ezzel az egyetlen eszközzel az informatika tantervek két nagy témaköre is lefedhető, nincs szükség második nyelv megtanítására, ami a szűkös órakeretek között kiemelt fontosságú. A megközelítés további előnye, hogy közvetlen kapcsolatot teremthetünk a többi tantárggyal. A matematikához leginkább a függvény fogalmán keresztül kapcsolódik, míg az összes többi tantárgyhoz a feldolgozásra kerüli táblázati tartalmakon keresztül.

Kulcsszavak: Sprego, informatika tantervek, problémamegoldás, számítógépes gondolkodás

1. Bevezetés

Sprego – Spreadsheet Lego [1] – egy mély megközelítésű metakognitív módszer [2], [4]–[7], amely grafikus táblázatkezelői környezetben tanít funkcionális programozást. A módszer egyrészt felhasználja a 90-es években már bizonyított [8] és a későbbiekben megerősített [9], [10], [11] eredményeket, mely szerint a funkcionális nyelvek hatékonyan használhatóak kezdő programozási nyelvként. Másrészt kihasználja az időközben elterjedt grafikus felületek előnyeit.

Booth [8] mérései egyértelműen bizonyítják, hogy a minimális háttérismeret és a funkcionális nyelvek egyszerűsége lehetővé teszi, hogy a problémamegoldás során ne a kódoláson legyen a hangsúly, hanem a problémán, a koncepció megértésén. Így nem a programozási eszköz megtanulása köti le a kezdő programozók energiáját, figyelmét, hanem a koncepció és az algoritmus megfogalmazása. A módszer azonban bármennyire is ígéretesnek mutatkozott, nem terjedt el. Az okokat vizsgálva, arra a következtetésre jutottunk, hogy Booth korában még nem álltak rendelkezésre azok az eszközök, amelyek valóban megkönnyítették volna a kódolás menetét. Másrészt, a 90-es évek eleje még nem követelte meg a széles felhasználói közönség magas szintű számítógépes gondolkodását [12], tehát csak egy szűk réteg volt, aki számítógépet használt. Ezt a réteget, szinte kizárólagosan a programozók alkották, akiknek nem jelentett komolyabb gondot bármiféle programozási nyelv megtanulása.

Az utóbbi évtizedekben azonban megjelent egy széles alkalmazói közösség, amely nem feltétlenül tudja megfogalmazni a fejlett számítógépes gondolkodás szükségességére vonatkozó igényét [13], de aktív számítógép-használó. Napjainkra, ugyanakkor bizonyítottá vált, hogy hatékony számítógépes problémamegoldás ezen alapkészség nélkül nem lehetséges [12]. A funkcionális nyelvek táblázatkezelői felületen történő használatára is hangzottak el javaslatok az utóbbi időszakban [9], [10], de ezek inkább elméleti síkon maradtak, konkrét, teljesen kidolgozott módszerek nem születettek. Óriási lépést jelentett még a grafikus táblázatkezelői felületek megjelené-

se, amely biztosítani tudja a kódolás egyszerűsítését, tehát elérhetővé teszi ezt a programázási eszközt a széles felhasználói réteg számára is.

Ezeket az elméleti megfontolásokat [6]–[11] és az elérhető eszközöket [11], [14]–[17] felhasználva született a Sprego programozás [1], [19], [18]–[21], amely nem más, mint programozás funkcionális nyelven grafikus táblázatkezelői felületen. A megközelítés szolgálhat végfelhasználók hatékony táblázatkezelési és programozási eszközöként, valamint programozók számára kezdő programozási nyelvként.

A Sprego hatékonyság vizsgálatát [22]–[26] a Testing Algorithmic and Application Skills (TAaAS) projekt keretein belül kezdtük meg a 2011/2012-es tanévben [2], [25]–[35]. A mérési eredmények egyértelműen bizonyítják, hogy a Sprego módszerrel lényegesen hatékonyabban tanítható a táblázatkezelés, mint a korábban kizárólagosan alkalmazott felületi megközelítésekkel [2], [19], [26]. Másrészt, sikerült közvetlen kapcsolatot találnunk, bizonyítanunk, hogy a számítógépes gondolkodás fejlesztésében a Sprego és az imperatív programozási nyelvek hatással vannak egymásra, erősítik egymást [25].

A táblázatkezelő programok használatával kapcsolatban azonban nem hagyhatjuk figyelmen kívül a jelenleg leginkább alkalmazott felületi megközelítéseket [5] és ezek hatását. A felületi megközelítések elsődleges jellemzője, hogy a fókusz a felület objektumainak a kezelésén van; gombok, varázslók, azok elhelyezkedése és használata. Ennek következtében, nem a probléma megoldásához szükséges koncepció és algoritmus megfogalmazása van a középpontban. A felületi megközelítések következménye az alacsony hatékonyságú problémamegoldás, a rendkívül magas számú hibás dokumentum, a tartalmak létrehozásához és lekérdezéséhez szükséges, ám indokolatlanul magas emberi és erőforrásigény [36]–[45].

Mindezen eredményeket figyelembe véve, érdemes megvizsgálni a tantervi lehetőségeket, azt, hogy a Sprego hogyan illeszthető be a közoktatásba, hogyan kapcsolódik az informatika tantárgy különböző témaköreire, valamint a többi tantárgyhoz.

2. Sprego eszközök

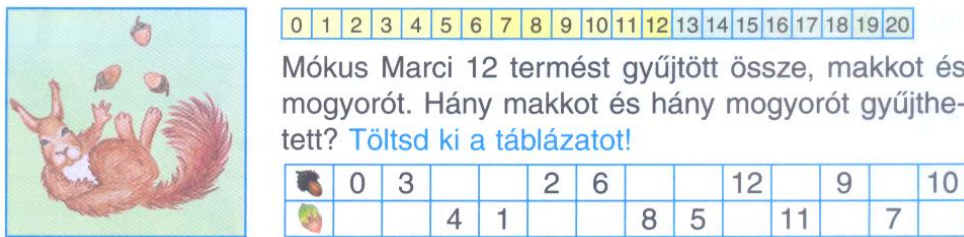
Sprego függvények

A Sprego keretein belül definiálunk egy szűk függvény halmazt, amely az ún. Sprego függvényeket tartalmazza [1]. A halmaz definiálásával célunk, hogy leszűkítsük, a szoftvergyártó cégek által javasolt, több száz függvényből álló készletet, amely a kezdők számára átláthatatlan dzsungel [35]. A Sprego alapkészlet egy tucat általános célú függvényt tartalmaz, amely szükség esetén tovább bővíthető. Fontos azonban hangsúlyozni, hogy a bővítéseknél az elsődleges szempont a függvények általános célú felhasználhatósága. Ennek megfelelően a Sprego alapkészlet a következő egy tucat függvényt definiálja: BAL(), JOBB(), HOSSZ(), SZÖVEG.KERES(), SZUM(), ÁTLAG(), MIN(), MAX(), HA(), HOL.VAN(), INDEX(), HIBÁS().

Függvény fogalom

A Sprego közvetlen kapcsolatot teremt az informatika és a matematika között [52]. Egyrészt a matematikai függvény fogalomra épít, másrészt alkalmazási teret biztosít az elméleti ismeretek gyakorlati megvalósításához, harmadrészt, a függvények programozási keretek közötti felhasználásával tovább bővíthetők a függvényekről szerzett ismeretek.

A táblázatokban tárolt diszkrét értékek, a tárolás és a hozzárendelés, szabályok megfogalmazása egybeesik a kisiskolás kortól alkalmazott és gyakorolt hozzárendelés fogalmával. Ennek ismeretében a táblázatkezelők használata, a hozzárendelések megfogalmazása számítógépes környezetben már korai iskolás korban elkezdhető.



Mókus Marci 12 termést gyűjtött össze, makkot és mogyorót. Hány makkot és hány mogyorót gyűjtött? Töltsd ki a táblázatot!

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

	0	3			2	6			12		9		10
	12	11	4	1			8	5		11		7	

1. ábra: Első osztályos matematika könyv feladata [46]

	0	3			2	6			12		9		10
	12	11	4	1			8	5		11		7	

2. ábra: Az első osztályos matematika feladat [46] mintájára készült táblázat. A piros karakter mutatja a felhasználó által gépelt értéket, a kék háttér pedig a helyes megoldást.

	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
9		12													
10		0	3	8	2	6			12		9		10		
11		12	9	4	1	4		8	5		11		7		
12															
13		1	1	1	0	0	0	0	0	0	0	0	0	0	0
14		=(H10:T10<>"")*(H11:T11<>"")*(H10:T10+H11:T11=G9)													

3. ábra: Az első osztályos matematika feladat [46] megoldásához készült képletről (H14), a képlet elrejtéséről (12-es sor) és a képlet outputjáról készült képernyő kép (13-as sor). (H12, H13 és H14 cellák ugyanazt a képletet tartalmazzák.) A feladat további általánosítása, hogy a konstans 12 helyett a G9 változó tartalmazza az összeget. Alapvető Sprego ismeretekkel bármelyik matematikatanár készíthet hasonló táblázatokat.

Eddigi tapasztalataink azt mutatják, hogy a tanulók megbízható, a gyakorlatban is használható alapismeretekkel rendelkeznek a függvények témakörben. A táblázatkezelői függvények alkalmazása magával hozza a többváltozós függvények bevezetését és rutinszerű használatát. (4. ábra, BAL() és JOBB() kétfváltozós függvények) A Sprego intenzíven használja és épít az összetett függvényekre. (4. ábra, E2 cella egymásba ágyazott függvényei. A JOBB() függvényhez képest a

BAL() kívül helyezkedik el, tehát a JOBB() függvény outputja a BAL() függvény egyik argumentuma.) Ez az az eszköz, amely szükséges ahhoz, hogy az egyszerű, általános célú függvények felhasználásával összetett problémák is megoldhatóak legyenek. Mindenképpen fontos megjegyezni, hogy az összetett és többváltozós függvények a matematika tananyagokban vagy egyáltalán nem jelennek meg, vagy csak marginális szerepet kapnak [47].

	B	C	D	E	F
1	Rating	Title	Votes		
2	9,2	A remény rabjai (1994)	727260	=BAL(JOBB(C2:C251;5);4)*1	
3	9,2	A keresztapa (1972)	544185	1972	A keresztapa
4	9	A keresztapa 2. (1974)	342859	1974	A keresztapa 2.
5	8,9	Ponyvaregény (1994)	571887	1994	Ponyvaregény
6	8,9	A jó, a rossz és a csúf (1966)	227552	1966	A jó, a rossz és a csúf
7	8,9	Tizenkét dühös ember (1957)	178266	1957	Tizenkét dühös ember
246	8	Nosferatu - Drakula (1922)	37943	1922	Nosferatu - Drakula
247	8	Egy gyilkosság anatómiája (1959)	22240	1959	Egy gyilkosság anatómiája
248	8	Erőszakik (2008)	155118	2008	Erőszakik
249	8	Viharsziget (2010)	250043	2010	Viharsziget
250	8	Három szín: piros (1994)	32485	1994	Három szín: piros
251	8	Laputa - Az égi palota (1986)	33802	1986	Laputa - Az égi palota

4. ábra: IMDB Top 250 Movies [48]

A Sprego megközelítés, természetesen alkalmazható a matematikai függvények ábrázolásához is. A módszerrel felgyorsítható az ábrázolás menete, a kézi függvényábrázolás kiváltható a gépi megjelenítéssel, valamint a képletekkel megvalósítható a függvények elemzése. Vizsgálhatóak a szélsőértékek, a szélsőértékhelyek, a zérus pontok, a függvény menete, szakadási pontok stb.

Tömbképletek

A Sprego kiemelten használja a tömbképleteket [15]–[17], amely a funkcionális programozási nyelvek egyik alapeszköze, de táblázatkezelői környezetben alig ismert. A tömbképletek használatának számtalan előnye van a GUI táblázatkezelői környezetekben. Egyrészt kiváltja a képletek másolását, amely műveletről bizonyításra került, hogy a táblázatkezelő dokumentumok egyik leggyakoribb hibaforrása [36], [37]. A másolás elhagyásával a kezdő felhasználóknál elkerülhető az abszolút és relatív hivatkozások bevezetésének szükségessége, ami tovább egyszerűsíti a kódolást, különös tekintettel a változók használatára. Ezzel a megoldással lényegesen csökkenthetőek a táblázatkezelői dokumentumok hiba forrásai. (A 3. ábra képlete tömbököt fogad inputként, és az outputként keletkező vektorokat szorozza össze. A szorzatvektor elemeit a táblázat 13-as sora tartalmazza. A 4. ábra képletében a JOBB() függvény az alapértelmezett egyetlen érték helyett egy vektort fogad be első argumentumként, majd a JOBB() függvény outputjaként keletkezett vektor lesz a BAL() függvény egyik argumentuma.)

További előnye a tömbképleteknek, hogy be tudjuk vezetni a tömb fogalmát – különös tekintettel a vektor fogalmára – egy olyan környezetben, ahol a tömb definiálása kódolás nélkül megvalósítható, a grafikus felület eszközeit használva. A vektorokkal bevezetjük az index, valamint az előre meghatározott lépésszámú ciklus fogalmát. Ezekkel az eszközökkel előkészíthetjük a deklaratív programozási nyelvek oktatását is.

A tömbképletek alkalmazásával az n-dimenziós vektor fogalma mindenképpen bevezetésre kerül, de jól megválasztott feladatok esetén az $n \times m$ dimenziós mátrix is. Ezen fogalmak ez idáig csak az emelt szintű matematikai, valamint programozási ismeretek között jelentek meg. A Spregoval találtunk tehát egy eszközt, amellyel olyan matematikai ismereteket tudunk közvetíteni, amelyekre eddig az általános és a középiskolai matematikaoktatási keretek között nem volt lehetőség. A programozási eszközökkel történő matematikatanítás az Angliában 2014-ben bevezetett informatika-tantervek egyik alapelgondolása [49]–[50] és [51].

Diszkusszió, debugging

A speciális célú, többváltozós beépített függvények óriási hátránya, hogy nem ellenőrizhető az algoritmus, a végrehajtás lépései. Ennek következtében a felhasználók többsége nem tud meggyőződni a képletek, az output helyességéről. Az így keletkezett hibák tovább csökkentik a táblázatkezelői dokumentumok megbízhatóságát.

A Sprego megközelítés egyrészt megköveteli a koncepció és az algoritmus megfogalmazását, másrészt az összetett függvények alkalmazása lehetővé teszi a megoldások lépésenkénti tesztelését. Ez a módszer hatékonyan fejleszti a felhasználók számítógépes gondolkodását és előkészíti a más programozási nyelveken történő problémamegoldást.

Az eredmények diszkussziója és a megoldások tesztelése mindenképpen csökkenti a lehetséges hibaforrások számát, ugyanakkor kialakítja, fejleszti az ellenőrzésre vonatkozó felhasználói igényt. Ezzel a módszerrel elkerülhetőek az olyan hibák, amelyet a frissen megjelent digitális matematika tankönyv tartalmaz [52], [53]. (A 6. ábra gyakorisági értékei azt mutatják, hogy 1-1 olyan király volt, aki $5 < x \leq 10$ és $10 < x \leq 15$ évig uralkodott. Ezen eredménynek azonban ellentmond az 5. ábra táblázata, amelyben 3 olyan király is található, akinek az $5 < x \leq 10$, valamint 4, akinek a $10 < x \leq 15$ intervallumba esett uralkodásának időtartama.) A példa azt mutatja, hogy a matematikában elfogadott és használt módszerek, hasonló problémák esetében sem kerültek át az informatikába. Ezen ellentmondásos helyzet feloldása érdekében kiemelt szerepe van az outputok és a forráskódok ellenőrzésének, a megoldások diszkussziójának és ellenőrzésének.

Ki az, aki két alkalommal is volt magyar király?							
Közismert név	Született	Trónralépés	Uralkodás vége	Elhunyt	Hány évesen lett király?	Uralkodásának időtartama	Hány évet élt?
Szent István	975	1001	1038	1038	26	37	63
Orseolo Péter	1011	1038	1041	1046	27	3	35
Aba Sámuel	988	1041	1044	1044	53	3	56
Orseolo Péter	1011	1044	1046	1046	33	2	35
I. András	1015	1046	1061	1061	31	15	46
I. Béla	1018	1061	1063	1063	43	2	45
Salamon	1053	1063	1074	1087	10	11	34
I. Géza	1039	1074	1077	1077	35	3	38
Szent László	1040	1077	1095	1095	37	18	55
Könyves Kálmán	1070	1095	1116	1116	25	21	46
II. István	1101	1116	1131	1131	15	15	30
Vak Béla	1108	1131	1141	1141	23	10	33
II. Géza	1130	1141	1162	1162	11	21	32
II. László	1131	1162	1163	1163	31	1	32
IV. István	1133	1163	1165	1165	30	2	32
III. István	1147	1162	1172	1172	15	10	25
III. Béla	1147	1172	1196	1196	25	24	49
Imre	1174	1196	1204	1204	22	8	30
III. László	1199	1204	1205	1205	5	1	6
II. András	1175	1205	1235	1235	30	30	60
IV. Béla	1206	1235	1270	1270	29	35	64
V. István	1239	1270	1272	1272	31	2	33
Kun László	1262	1272	1290	1290	10	18	28
III. András	1250	1290	1301	1301	40	11	51

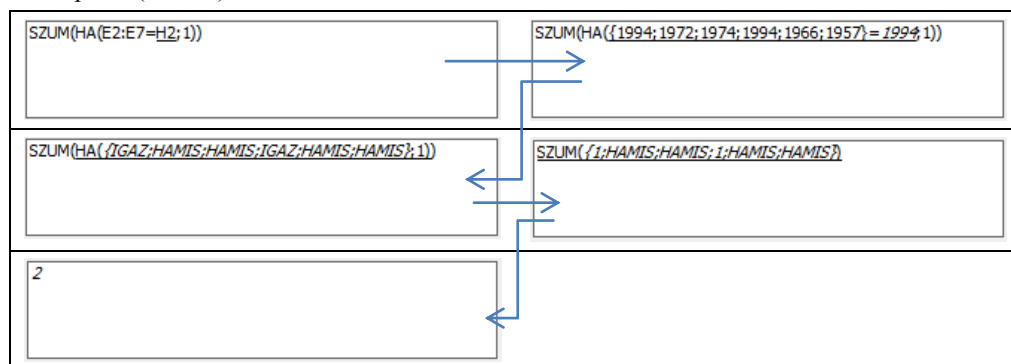
5. ábra: Árpád-házi királyok. Tények és következtetések (2. dia) [53]

Uralkodásuk időtartama szerint csoportosítva

Uralkodás időtartama	Királyok száma (Gyakoriság)	Relatív gyakoriság
<=5 év	9	9/24=38%
>5 és <=10	1	1/24=4%
<10 és <=15	1	1/24=4%
>15 és <=20	7	7/24=29%
>20 és <=25	3	3/24=13%
>25 és <=30	1	1/24=4%
>30 és <=35	1	1/24=4%
>35 és <=40	1	1/24=4%

6. ábra: Árpád-házi királyok. Tények és következtetések (6. dia) [53]

Megkönnyíti, valamint programozás-kompatibilissé teszi a tesztelést (debugging) az Excel *Képletkiértékelő* szolgáltatása. Sprego függvényeket használva és ezekkel összetett függvényeket létrehozva, ezen a felületen lépésről, lépésre követhető a kiértékelés menete. A *Képletkiértékelő* tipográfiai eszközöket használva megkülönbözteti az éppen kiértékelésre váró utasítást és az outputot: aláhúzással jelzi a következő kiértékelést és dőlttel a kiértékelés eredményét, az aktuális outputot (7. ábra).



7. ábra: A *Képletkiértékelő* lépései a H2 évben (H2 aktuális értéke 1994) készült filmek számának meghatározására készült `{=SZUM(HA(E2:E7=H2;1))}` tömbképlet alapján (4. ábra). A képlet helyettesíti a `DARABTELI()` és a `DARABHATÖBB()` beépített függvényeket (ZZZ fejezet).

Megnehezíti a *Képletkiértékelő* használatát a „fapados” megjelenítés, ami az utóbbi két évtizedben semmit nem változott. Az ablak nem méretezhető, a betűméret nem állítható, így egy hosszabb tömb elemzése, különösen, hosszabb szöveges értékekkel, rendkívül körülményes. A hatékonyság növelése érdekében, a programozásból jól ismert módszert, a tömb egyszerűsítését, lerövidítését érdemes elvégezni. A 7. ábra mintáján a tömb méretét csökkentettük, hogy a kiértékelés menete követhető legyen a rendelkezésre álló szűk területen.

Beépített függvények és Sprego képletek

Sprego függvények használatának további előnye, hogy a felhasználó, a program keretein belül, tetszőleges problémák megoldására képes algoritmusokat megfogalmazni, ennek következtében a beépített függvények korlátai, hiányosságai, hibái nem hátráltatják a végrehajtást [1]. A Sprego függvényekkel lényegesen szabadabban tudjuk elvégezni a kódolást, mint a speciális célú és igényű beépített függvényekkel. A szoftver eszközök mathability szintjeit [54], [55] tekintve a beépített függvények és a Sprego képletek létrehozása egyértelműen megfeleltethető a két nagy mathability osztálynak: reproducing és developing. A módszerek reproducing osztálya tartalmazza azokat a megközelítéseket, amelyek a meglévő szoftvereszközöket alkalmazzák – felületi megközelítések –, míg a developing osztálya a szoftvereszközöket felhasználva új eszközöket, programokat, módszereket fejleszt a probléma megoldásához – mély megközelítések [56], [56].

A főlegesen létrehozott és nehézkesen használható beépített függvények közül mindenképpen említést érdemelnek az *IF?() függvények – feltételeket befogadó függvények –, valamint a vektorban kereső FKERES() és VKERES() függvények [1] (8. ábra).

INDEX(HOL.VAN())	FKERES(), VKERES()
Keresési vektor	
keres oszlopban és sorban is	csak oszlopban keres – FKERES() csak sorban keres – VKERES()
Találati vektor	
bármelyik oszlop és sor	csak a keresési oszlop és attól jobbra –FKERES() csak a keresési sor és az alatt –VKERES()
Keresési vektor rendezettsége	
– növekvő sorrend (1, alapértelmezett) – csökkenő sorrend (-1) – nincs rendezés (0)	– növekvő sorrend (IGAZ, alapértelmezett) – csökkenő sorrend (nem lehetséges) – nincs rendezés (HAMIS)
Visszaadott érték	
– egyetlen érték (alapértelmezett) – vektor (sor vagy oszlop index 0)	– egyetlen érték (alapértelmezett) – vektor (nem lehetséges)

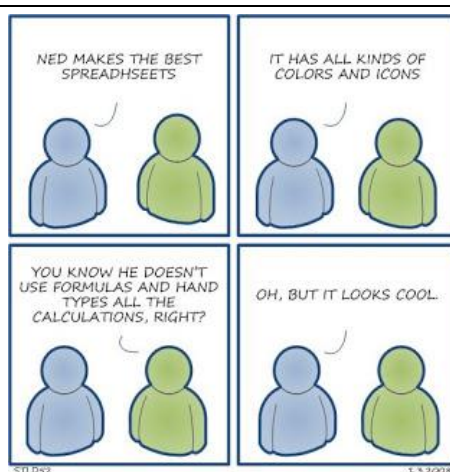
8. ábra: Az FKERES() és VKERES() függvények összehasonlítása az INDEX(HOL.VAN()) összetett függvénnyel [1]

Az *IF?() függvények problémái között meg kell említenünk, hogy számuk korlátozott, ennek következtében mindig jelentkezhet olyan probléma, amire nincs ilyen típusú beépített függvény. Igaz továbbá, hogy csak ÉS kapcsolat valósítható meg velük, nehezen kezelik az egyenlőtlenséget, a változót, valamint nem fogadnak el feltételként függvényt. Ezek a megkötések rendkívüli módon leszűkítik az *IF?() függvények használhatóságát. Találkozni olyan véleményekkel, amelyek azt fogalmazzák meg, hogy az *IF?() függvények helyettesíthetők az adatbázis függvényekkel. Ez azonban kezdő felhasználók esetén nem releváns, mivel a kezdők még nem rendelkeznek

megfelelő szintű adatbázis ismeretekkel. Továbbá, méréseink bizonyítják, hogy az adatbázis függvények még gyakorlott felhasználók esetén is rendkívül hibaérzékenyek. Használatuk, tehát nem javasolt.

A vektorban kereső FKERES(), VKERES() függvények helyettesíthetők az egymásba ágyazott INDEX(HOL.VAN()) összetett függvénnyel. A két említett speciális függvény és az INDEX(HOL.VAN()) összetett függvény összehasonlítását a 8. ábra tartalmazza.

Ha a problémákkal terhelt függvények tanítását kihagyjuk, akkor lényegesen több idő jut tényleges problémamegoldásra, mivel nincs szükség az értelmetlen korlátok megmagyarázására és elfogadtatására. Ezen túl, továbbra is érvényes, hogy az általános célú függvények használata lényegesen több probléma megoldására használható. A felhasználóbarát szlogen mögé rejtett értelmetlen részletek tanítását és alkalmazását, annak gyakorlati hasznát a 9. ábra párbeszéde remekül jellemzi.



9. ábra: Tartalmilag értékelhetetlen, indokolatlan formázásokat tartalmazó dokumentumok készítőit kifigurázó karikatúra (K: Ned készíti a legjobb táblázatokat. K: Tartalmaz mindenféle színeket és ikonokat. Z: Tudod ugye, hogy nem használ képleteket és minden egyes számítás eredményét kézzel viszi be? K: Oh, de azért jól néz ki.) [58]

Autentikus táblázatok

Az iskolai oktatásunk egyik sarkalatos pontja, hogy nem készíti fel a tanulókat az iskolán kívüli mindennapokra, ennek következtében elválík egymástól az iskolai követelményrendszer és a munkahelyek elvárásai. Ezzel magyarázható sok esetben az iskolai szoftverek iránt mutatott érdektelenség [59]. Nem látják a tanulók, hogy az iskolai alkalmazások hogyan adaptálhatók a valós világban.

A Sprego eszközeivel megvalósítható, hogy a „klasszikus” órai gépelések helyett már az első pillanattól kezdve autentikus táblázatokkal dolgozzunk tanórai keretek között is. Ennek a módszernek az előnye, hogy olyan táblázatokon tudunk adatlekérdezést végezni, amelynek témája érdekli a hallgatóságot, a tartalmak megfeleltethetők a hallgatói elvárásoknak, előképzettségnek és már a feldolgozások elején bizonyítani tudjuk a táblázatkezelő programok létjogosultságát,

ezek előnyeit a kézi adatfeldolgozással szemben. A módszer tehát alkalmas arra, hogy közvetlen kapcsolatot teremtsen más tantárgyak tartalmával, valamint iskolán kívüli tartalmakkal is (4. ábra, az IMDB adatbázis első 250 filmjének listája, 5 és 6. ábra, Árpád-házi királyok születési, halálzási és uralkodási idejét tartalmazó táblázat).

Informatikai szempontból további előnye az autentikus táblázatok tanórai felhasználásának, szemben a gépeléssel, hogy ezen táblázatok használatával lehetőség nyílik intenzív fájlkezelésre, fájlkonverzióra, az alkalmazások közötti átjárhatóság megismertetésére, gyakorlására [60]. Előkészítheti továbbá az adatbázis kezelést, példákat tudunk mutatni helytelen adatbázis tervezésre (4. ábra, a C oszlop két adatot tartalmaz, 5. ábra, *Uralkodásának időtartama* és a *Hány évet élt?* oszlopok redundáns adatokat tartalmaznak) és megoldásokat ezek javítására. Ezzel a megközelítéssel megmutatható, hogy a táblázatkezelés nem feltétlenül gyakorol negatív hatást az adatbázis kezelésre.

Verzió függetlenség

A Sprego előnye minden eddigi táblázatkezelői eszközzel szemben, hogy a két nagy táblázatkezelő program – Microsoft Excel és OpenOffice, LibreOffice Calc – között Sprego függvények használata esetén teljes az átjárás, és ugyanez igaz a programok különböző verziói esetén is. Nincs tehát szükség annak folyamatos ellenőrzésére, hogy a dokumentumokban használt függvények megtalálhatóak-e az éppen használt programban és verzióban.

3. Összegzés

A széles körű felhasználói réteg megjelenésével olyan mély megközelítésű metakognitív módszerek kidolgozása vált szükségessé, amelyekkel hatékonyan fejleszthető a nem-programozók számítógépes gondolkodása is. Jelenleg a felületi megközelítések uralják a felhasználói problémák megoldását, melynek következménye a rendkívül nagyszámú hibás dokumentum és a problémák megoldásához szükséges indokolatlanul magas emberi és gépi erőforrás igény.

Táblázatkezelői környezetben kidolgoztuk a Sprego programozást, amely egy mély metakognitív megközelítés. A módszer egyrészt szolgálhat a végfelhasználók programozási nyelveként, másrészt programozók kezdő programozási nyelveként. A Sprego mint bevezető nyelv előnye az imperatív programozási nyelvekkel szemben, hogy a táblázatkezelők grafikus felületén rendkívül egyszerű a kódolás, így a hangsúly nem a nyelv sajátosságain van, hanem a problémán. A végfelhasználókat tekintve a Sprego szolgálhat egy kényelmes programozási nyelvként, amely elegendő nem-professzionális programozási problémák megoldásához.

A Sprego további előnye, hogy eszközeit használva közvetlen kapcsolat teremthető más tantárgyakhoz, felhasználva azok tartalmait. A matematika kiemelt szerepet kap Sprego környezetben, mivel egyrészt a matematikában kialakított függvény fogalomra épít, másrészt gyakorló felületet biztosít a függvény fogalom megerősítéséhez, és korábban emelt szintűnek tekintett ismeretek bevezetésére nyílik lehetőség.

A Sprego jellegzetességeit összegezve megfogalmazhatjuk, hogy használatával teljes mértékben helyettesíthetőek, a korábban kizárólagosan alkalmazott, felületi megközelítések a táblázatkezelés-oktatásban. A Sprego ugyanakkor szolgálhat bevezető programozási nyelvként is, tehát két témakör is lefedhető egyetlen eszközzel. A rendkívül szűkös órakeretek között óriási előny, hogy csak egyetlen nyelvet kell megtanítani, megtanulni és az két környezetben is használható. A

Sprego további előnye, hogy bizonyítottan javítja a hallgatók számítógépes gondolkodását, ami feltétlenül szükséges a hatékony számítógépes problémamegoldáshoz.

Irodalom

- [1] Csernoch, M.: *Programozás táblázatkezelő függvényekkel*. Sprego. Műszaki Könyvkiadó, Budapest (2014)
- [2] Biró, P., Csernoch, M.: *Deep and surface structural metacognitive abilities of the first year students of Informatics*. 4th IEEE International Conference on Cognitive Infocommunications, Proceedings, Budapest (2013) 521–526
- [3] Ben-Ari, M.: *Bricolage Forever! PPIG 1999*. 11th Annual Workshop. 5–7 January 1999. Computer-Based Learning Unit, University of Leeds, UK. <http://www.ppig.org/papers/11thbenari.pdf> (2014. 04. 12.)
- [4] Ben-Ari, M., Yeshno, T.: *Conceptual models of software artifacts*. *Interacting with Computers* 18 (2006) 1336–1350
- [5] Csernoch, M., Biró, P.: *Számítógépes problémamegoldás*, TMT, Tudományos és Műszaki Tájékoztatás, Könyvtár- és információtudományi szakfolyóirat, Vol. 62(3) (2015) 86–94
- [6] *Informatics education: Europe cannot afford to miss the boat*. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education April 2013. <http://germany.acm.org/upload/pdf/ACMandIEreport.pdf>. (2014. 02. 02.)
- [7] *IEEE&ACM Report 2013. Computer Science Curricula 2013*. (2013) The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society. Letöltés: <http://www.acm.org/education/CS2013-final-report.pdf>. (2014. 04. 04.)
- [8] Booth, S.: *Learning to program: A phenomenographic perspective*. Gothenburg, Sweden: Acta Universitatis Gothoburgensis (1992)
- [9] Warren, P.: *Learning to program: spreadsheets, scripting and HCI*. in Proceedings of the Sixth Australasian Conference on Computing Education – vol. 30, Darlinghurst, Australia (2004) 327–333
- [10] Lister, R., Simon, B., Thompson, E., Whalley, J. L., Prasad, C. *Not seeing the forest for the trees: novice programmers and the SOLO taxonomy*, in Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education, New York, NY, USA (2006) 118–122
- [11] Sestoft, P.: *Spreadsheet technology*. Version 0.12 of 2012-01-31. IT University (2011)
- [12] Wing, J. M.: *Computational Thinking*. March 2006/Vol. 49, No. 3 Communications of the ACM (2006)
- [13] Soloway, E.: *Should we teach students to program?* Communications of the ACM. October 1993/Vol.36, No.10 (1993) 21–24
- [14] Wakeling, D.: *Spreadsheet functional programming*. JFP 17(1) Cambridge University Press. (2007) 131–143
- [15] Walkenbach, J., Wilcox, C.: *Putting basic array formulas to work*. <http://office.microsoft.com/en-us/excel-help/putting-basic-array-formulas-to-work-HA001087292.aspx?CTT=5&origin=HA001087290> (2012.05.08.)
- [16] Walkenbach, J.: *Excel2003 Formulas*. John Wiley & Sons (2003)
- [17] Wilcox, C., Walkenbach, J.: *Introducing array formulas in Excel*. <http://office.microsoft.com/en-us/excel-help/introducing-array-formulas-in-excel-HA001087290.aspx> (2013. 01. 18.)

- [18] Csernoch, M., Balogh, L.: *Algoritmusok és táblázatkezelés*. Tehetséggondozás a közoktatásban az informatika terén. Magyar Tehetségsegítő Szervezetek Szövetsége, Budapest (2011)
- [19] Biró, P., Csernoch, M.: *Táblázatkezelés algoritmikus megközelítése*. Kiss Árpád Emlékkonferencia Tanulmánykötete 2013, Debrecen (2014)
- [20] Csernoch, M., Biró, P.: *Sprego programming*. Spreadsheets in Education (eJSIE) Vol. 8, Iss. 1. <http://epublications.bond.edu.au/cgi/viewcontent.cgi?article=1175&context=ejsie>. (2015) (2015. 10. 12.)
- [21] Csernoch, M., Biró, P.: *Sprego programming*. LAP Lambert Academic Publishing. ISBN-13: 978-3-659-51689-4. (2015)
- [22] Csernoch, M., Biró, P.: *Button-up technikák hatékonyságának vizsgálata informatika szakos hallgatók táblázatkezelés-oktatásában*. Szerk: Kozma Tamás és Perjés István, Új kutatások a neveléstudományokban 2012, ELTE Eötvös Kiadó (2013) 369–392
- [23] Csernoch, M., Biró, P.: *Teachers' Assessment and Students' Self-Assessment on The Students' Spreadsheet Knowledge*. EDULEARN13 Proceedings July 1st-3rd, 2013 — Barcelona, Spain. Publisher: IATED (2013) 949–956
- [24] Csernoch, M., Biró, P.: *Spreadsheet misconceptions, spreadsheet errors*. Oktatáskutatás határon innen and túl. HERA Évkönyvek I., ed. Juhász Erika, Kozma Tamás, Publisher: Belvedere Meridionale, Szeged, 2014 (2014) 370–395
- [25] Csernoch, M., Biró, P., Máth, J., Abari, K.: *Testing Algorithmic Skills in Traditional and Non-Traditional Programming Environments*. Informatics in Education, Vilnius University, Vol. 14, No. 2. (2015)
- [26] Biró, P., Csernoch, M.: *Deep and surface metacognitive processes in non-traditional programming tasks*. In: 5th IEEE International Conference on Cognitive Infocommunications CogInfoCom 2014 Proceedings. IEEE Catalog Number: CFP1426R-USB, Italy (2014) 49–54
- [27] Biró, P., Csernoch, M.: *Elsőéves informatikushallgatók algoritmizáló készségei*. XXIII. Nemzetközi Számítástechnika és Oktatás Konferencia - SzámOkt 2013, EMT (2013) 154–159
- [28] Csernoch, M., Biró, P.: *Digital Competency and Digital Literacy is at Stake*, ECER 2014 Conference, 1–5. September, 2014, Porto, Portugal (2014)
- [29] Biró, P. Csernoch, M., Abari, K., Máth J.: *First year students' algorithmic skills in tertiary Computer Science education*, 9th International Conference on Knowledge, Information and Creativity Support Systems, 6–8. November, 2014, Cyprus, Limassol, Ed.: George Angelos Papadopoulos, Cyprus Library (2014) 301–306
- [30] Biró, P. Csernoch, M., Máth, J., Abari, K.: *Algorithmic skills transferred from secondary CSI studies into tertiary education*, World Academy of Science, Engineering and Technology International Journal of Social, Education, Economics and Management Engineering Vol. 9., No. 2 (2015) 292–298
- [31] Biró P., Csernoch M., Máth J., Abari K.: *Measuring the level of algorithmic skills at the end of secondary education in Hungary*. Procedia - Social And Behavioral Sciences 176 (2015) 876–883
- [32] Csernoch, M., Biró, P.: *Digital Competency and Digital Literacy is at Stake*, ECER 2014 Conference, 1–5. September, 2014, Porto, Portugal, The Past, Present and Future of Educational Research in Europe. University of Porto, pp. 1–4, 2014. <http://www.eera-ecer.de/ecer-programmes/conference/19/contribution/31885/> (2014)
- [33] Csernoch M., Biró P.: *The power in digital literacy and algorithmic skill*. Procedia - Social And Behavioral Sciences 174 (2015) 550–559

- [34] Csernoch, M. Biró, P., Máth, J., Abari, K.: *Mit tudok informatikából?* IF2014 Conference, 27–29 August, 2014, Debrecen. Ed.: Kunkli Roland, Papp Ildikó, Rutkovszky Edéné, University of Debrecen, Faculty of Informatics, Debrecen (2014) 217–230
- [35] Csernoch, M., Biró, P., Abari, K., Máth, J.: *Programozásorientált táblázatkezelői függvények*. In: Buda András (szerk.) XIV. Országos Neveléstudományi Konferencia. Oktatás és nevelés – gyakorlat és tudomány: tartalmi összefoglalók. Konferencia helye, ideje: Debrecen, Magyarország, Debreceni Egyetem Neveléstudományok Intézete, Poszter (2014)
- [36] Panko, R. R.: *What We Know About Spreadsheet Errors*. Journal of End User Computing's. Special issue on Scaling Up End User Development. (10)2 (2008) 15–21
- [37] Panko, R., Aurigemma, S.: *Revising the Panko-Halverson taxonomy of spreadsheet errors*. Decis. Support Syst. 49, 2 (2010) 235–244
- [38] Tort, F.: *Teaching Spreadsheets: Curriculum Design Principles*. In S. Thorne (Ed.), Proceedings of the EuSprIG 2010 conference: Practical steps to protect organisations from out-of-control spreadsheets (2010) 99–110
- [39] Tort, F., Blondel, F.-M., Bruillard É.: *Spreadsheet Knowledge and Skills of French Secondary School Students*. R.T. Mittermeir and M.M. Sysło (Eds.): ISSEP 2008, LNCS 5090 Springer-Verlag Berlin Heidelberg. (2008) 305–316
- [40] EuSprIG Horror Stories. <http://www.eusprig.org/horror-stories.htm> (2015. 07. 18.)
- [41] Powell, S. G., Baker, K. R., Lawson, B.: *A critical review of the literature on spreadsheet errors*. Decision Support Systems, 46(1) (2008) 128–138
- [42] Powell, S. G., Baker, K. R., Lawson, B.: *Errors in operational spreadsheets*. Journal of Organizational and End-User Computing, 1(3) (2009) 4–36
- [43] Powell, S. G., Baker, K. R., Lawson, B.: *Impact of errors in operational spreadsheets*. Decision Support Systems, 47(2) (2009) 126–132
- [44] *Report of JPMorgan Chase & Co. Management Task Force*. Regarding 2012. CIO Losses. http://files.shareholder.com/downloads/ONE/2272984969x0x628656/4cb574a0-0bf5-4728-9582-625e4519b5ab/Task_Force_Report.pdf (2014. 05. 17.)
- [45] Van Deursen, A., Van Dijk, J.: CTRL ALT DELETE. Lost productivity due to IT problems and inadequate computer skills in the workplace. Enschede: Universiteit Twente. http://www.ecdl.ch/fileadmin/ECDL/CH/Dokumente/Studie_CTRL-ALT-DELETE-en.pdf (2014. 05. 18.)
- [46] Scherlein, M., Hajdu, S. és Novák, L.: Matematika 1. osztály. Műszaki Könyvkiadó, Budapest (2007)
- [47] Kerettanterv. 51/2012. (XII. 21.) számú EMMI rendelet – a kerettantervek kiadásának és jóváhagyásának rendjéről. <http://kerettanterv.ofi.hu/> (2015. 06. 25.)
- [48] IMDb Charts. Top Rated Movies. Top 250 as voted by IMDb Users. <http://www.imdb.com/chart/top> (2012. 03. 19.)
- [49] Gove, M. (2012) Michael Gove speech at the BETT Show 2012. Published 13 January 2012. <https://www.gov.uk/government/speeches/michael-gove-speech-at-the-bett-show-2012> (2014. 04. 15.)
- [50] Gove, M.: *Michael Gove speaks about computing and education technology*. Published 22 January 2014. <https://www.gov.uk/government/speeches/michael-gove-speaks-about-computing-and-education-technology> (2014. 04. 15.)
- [51] *Computing programmes of study: key stages 3 and 4 National curriculum in England*. Published: September 2013

- https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/381754/SECOND_ARY_national_curriculum.pdf (2015. 08. 23.)
- [52] Csernoch M.: *Függvények a matematika tankönyveken túl*. MAFIOK XXXIX. ORSZÁGOS KONFERENCIA. Kaposvár, 2015. augusztus 24–26.
- [53] Kapcsolatok, változások, bepillantás a függvények világába - Statisztika, függvénytan. <http://etananyag.ofi.hu/konyvek/matematika-9-i-kotet/kapcsolatok-valtozasok-bepillantas-a-fuggvenyek-vilagaba-statisztika-fuggvenytan>. 34. fejezet. Táblázatok. Árpád-házi királyok. Tények és következtetések. http://etananyag.ofi.hu/tananyagok/3484_arpad-hazi-kiralyok (2015. 10. 10.)
- [54] Baranyi, P., Gilanyi, A.: *Mathability: Emulating and Enhancing Human Mathematical Capabilities*, CogInfoCom 2013, 4th IEEE International Conference on Cognitive Infocommunications, Budapest (2013) 555–558
- [55] Borus, G. Gy., Gilányi, A.: *Solving Systems of Linear Functional Equations with Computer*, CogInfoCom 2013, 4th IEEE International Conference on Cognitive Infocommunications, Budapest (2013) 559–562
- [56] Biró, P., Csernoch, M.: *The mathability of computer problem solving approaches*. In: Peter Baranyi (szerk.) Proceedings of 6th IEEE Conference on Cognitive Infocommunications. Konferencia helye, ideje: Győr, 2015.10.19-2015.10.21. Győr: IEEE (2015) 111–114
- [57] Biró, P., Csernoch, M.: *The mathability of spreadsheet tools*. In: Peter Baranyi (szerk.) Proceedings of 6th IEEE Conference on Cognitive Infocommunications. Konferencia helye, ideje: Győr, 2015.10.19-2015.10.21. Győr: IEEE (2015) 105–110
- [58] *Colors and Icons*. <http://std52.blogspot.ch/2008/01/colors-and-icons.html>. (2015. 10.19.)
- [59] Message, R.: *Programming for humans: a new paradigm for domain-specific languages*. Technical Report. UCAM-CL-TR-843. University of Cambridge, Computer Laboratory (2013)
- [60] Csernoch, M.: *Algoritmusalapú szöveg- és táblázatkezelés*. SZAKTÁRNET (közlésre elfogadva)
- [61] Vágner, A., Zsakó, L.: *Negative Effects of Learning Spreadsheet Management on Learning Database Management*. Volume 8 Number 2. Published: 24 August 2015. http://padi.psiedu.ubbcluj.ro/adn/article_8_2_1.pdf (2015)