



**Módszerek és eszközök az
informatikaoktatás hatékonyságának
növelésére**

Doktori (PhD) értekezés *tézisei*

Kátai Zoltán

Debreceni Egyetem
Természettudományi Kar
Debrecen, 2006

Tartalomjegyzék

1 Bevezetés	1
2 Algoritmustervezés felülnézetből	2
3 A látás, hallás és kinezetikus érzékelés bevonása elemi algoritmusok tanításába	5
4 „Legyél te is eminens” - Értékelési módszer és szoftver	8
Irodalomjegyzék	11
Publikációs jegyzék	14

1 Introduction.....	15
2 „Upperview” Algorithm Design	16
3 Involvement of the sight, hearing and the phenomenon of kinaesthesia in teaching- learning process of elementary algorithms	19
4 “Who wants to be eminent” – Assessment method and software.....	22
References.....	25
Publications.....	28

1 Bevezetés

A XX. század második felétől az oktatás bizonyos didaktikai vonatkozásai egyre nagyobb hangsúlyt kaptak, és számos didaktikai szempont új megvilágításba került. Ehhez elsősorban a digitális számítógép feltalálása, az internet létrehozása, valamint az utóbbi években az agykutatás területén elért eredmények járultak hozzá. Mindhárom nyugodtan nevezhetjük forradalminak, ami az oktatásra gyakorolt hatásukat illeti.

A számítógép megjelenése, majd az internet kialakítása nyomán a figyelem középpontjába új tanítási, tanulási és értékelési formák kerültek. Ami az emberi agyat illeti, a kutatások feltárták, hogy az egy rendkívül rugalmas biológiai szerkezet [1]. Két fő tényező határozza meg, hogy agyunk miként fejlődik az életünk folyamán: amit érzékszerveinkkel agyunkba beengedünk, és amit gondolkodásunk témájául választunk. Mindkét tényező kitüntetett módon van jelen a tanítási-tanulási folyamatban.

Az agykutatásnak és a számítógépnek az oktatás folyamatára kifejtett hatása úgy is megnyilvánul, hogy hangsúlyt kapnak régóta ismert didaktikai alapelvek, vagy új lehetőségek tárulnak fel az alkalmazásukra. Comenius [6] például a következő megállapítást tette: „A tanulás legyen teljesen gyakorlatias, teljesen szórakoztató, ... olyan, hogy általa az iskola valóban a játék helyévé, vagyis az egész élet előjátékává váljon.” Másik jól ismert, és a modern kutatások által erőteljesen alátámasztott didaktikai alapelv, hogy a tanulás annál hatékonyabb, minél több érzékszervet vonunk be. A számítógépek megjelenése, illetve a számítástechnika fejlődése új távlatokat nyitott e több száz éve megfogalmazott alapelvek hatékony alkalmazásához.

A fejlett társadalmak jelenlegi és a közeljövőben bekövetkező szellemi, gazdasági és politikai átalakulásai egyértelmű változásokat idéznek elő az iskolarendszerben is. A tudás korszakában a súlypont a technológiáról az információra helyeződik át [3, 4]. E változások szempontjából határkőnek számít a Bolognai Konferencia. A bolognai oktatási reform alapvető építőkövei a tanulási eredmények, azon ismeretek, készségek, képességek és kompetenciák, amelyeket az egyén a tanulási folyamat végére elsajátít. Az igazi fordulópontot az információ tudássá alakításának és alkotó alkalmazásának a képessége jelenti [5]. A bolognai reformprogramban is megfogalmazódó új irányzatok a tanítás-tanulás-értékelés közötti szoros kapcsolatot hangsúlyozzák, mint amelyek egy visszacsatolós folyamat szorosan összefüggő elemei [9].

E forradalmi előrelépésekkel, illetve társadalmi szintű változásokkal egyidejűleg egy paradigmaváltásnak vagyunk a szemtanúi: a tanítástól a tanulás felé, a tanárközpontú oktatástól a diákközpontú felé. E paradigmaváltással megváltozott a tanár szerepe az oktatás folyamatában. A modern tanár már nem az ismeretelméleti tudás központja, hanem a tanulási folyamat segítője és előmozdítója. Ma már minden hatékony oktatási gyakorlatnak egyik alapvető tényezője egyértelműen, hogy a tanulók optimális tanulási folyamataikhoz szükséges feltételekre összpontosít. Az aktív tanulás jelentősége azon a felismerésen alapszik, hogy az intelligencia elválaszthatatlanul összekapcsolódik a cselekvéssel (pszichológiai aspektus), és hogy a szervezet aktivitása az idegsejtek növekedésének és összekapcsolódásának alapvető feltétele (neurofiziológiai aspektus). Ha a tanulási tartalmakkal való aktív foglalkozásnak biztosítunk feltétlen előnyt, akkor ennek hatása van az oktatási folyamat tervezésére, a teljesítmények ellenőrzésére és a

teljesítmény értékelésére is. Mindennek a tanuló személyiségéhez szükséges igazodni [10].

Ahhoz, hogy az imént felvázolt új irányzatok kifejthessék jótékony hatásukat a tanulókra, többre van szükség a tanulásméletek átértékelésénél. A tanároknak égetően szükségük van olyan konkrét didaktikai módszerekre és eszközökre, amelyek átviszik a gyakorlatba a legújabb kutatások eredményeit, és kihasználják a számítógépek nyújtotta lehetőségeket.

Bár az irányelvek egyre világosabbak, és a lehetőségek egyre elérhetőbbek, gyakran továbbra is hiányoznak a konkrét didaktikai módszerek és a hatékony didaktikai eszközök. A jelen dolgozat az informatikaoktatás e területein igyekszik előrelépést tenni. A dolgozat anyagának kidolgozásában 15 éves informatikaoktatási tapasztalatokra, és az e területen végzett kutatásokra támaszkodtunk.

A dolgozatban három új didaktikai módszert mutatunk be, amelyekkel a tanítási-tanulási-értékelési folyamat mindhárom elemének hatékonysága növelhető. Az elsőt (**Algoritmustervezés felülnézetből**) az informatikaoktatás egy sajátos területe, az algoritmustervezési stratégiák oktatásához dolgoztuk ki. A módszer alkalmazásához szükséges tartalom biztosítása érdekében a szerző kiadott – ugyanezzel a címmel - egy könyvet is. [11] Bár elsősorban *tanítási* módszer, hangsúlyt fektet a tanulók aktivizálására.

A második módszer (**A látás, hallás, valamint a kinesztetikus érzékelés bevonása elemi algoritmusok tanításába**) az elemi algoritmusok *tanításának és tanulásának* hatékonyságát hivatott növelni. Ehhez a módszerhez egy szoftvert is kidolgoztunk, amely mint didaktikai eszköz lehetővé teszi - mind a tanárnak, mind a tanulóknak - a módszer hatékony alkalmazását a tanítás, illetve a tanulás során. Amint a módszer neve is utal rá, a tanulók szinte teljes lényükkel részt vesznek a tanulásban. A diákok a szoftvert mint e-learning eszközt önállóan, akár otthon is használhatják.

Végül a harmadik módszerrel, amely egy hálózati szoftver („**Legyél te is eminens**”) köré épül, az *értékelés* fázisát szeretnénk hatékonyabbá tenni. Egy olyan tesztalapú „feleltető szoftverről” van szó, amely szinte minden tantárgy keretén belül jól használható, és lehetővé teszi a tanítás-tanulás-értékelés folyamatára, mint egészre való rálátást.

2 Algoritmustervezés felülnézetből

„Tanítani szinte nem is jelent mást, mint megmutatni, miben különböznek egymástól a dolgok a különböző céljukat, megjelenési formájukat, és eredetüket illetően. ... Ezért aki jól megkülönbözteti egymástól a dolgokat, az jól is tanít.” [6] Az ebben a fejezetben bemutatott didaktikai módszer elsősorban erre az alapelvre épül. Egy olyan tanítási, illetve tanulási módszerről van szó, amely segít a tanulóknak, úgymond felülnézetből látni öt alapvető algoritmustervezési stratégiát (visszalépéses keresés, oszd meg és uralkodj, mohó, dinamikus programozás, elágazás korláttal). A módszer célja az, hogy e programozási technikák bemutatásán túl, olyan nézőpontba juttassuk a tanulót, amelyből feltárulnak előtte a technikák közötti elvi, alapvető, sőt árnyalatbeli különbségek, illetve hasonlóságok. A fentebb idézett comeniusi alapelvvel összhangban ez nélkülözhetetlen, ha uralni szeretnénk a programozás e területét.

A szakirodalomban számos példát találunk a programozási technikák összehasonlítására. Cormen, Leiserson és Rivest például a referenciamunkájukban [17] összehasonlítják a dinamikus programozás és mohó stratégiákat. Más kiadványok [18] azt tárgyalják, hogy miként egészíthetik ki egymást a visszalépéses keresés és mohó technikák. Razvan Andone és Ilie Garbacea a dinamikus programozás és az oszd meg és uralkodj stratégiákról közöl összehasonlító elemzést [19].

Másik alkalmazott módszer a technikák párhuzamos bemutatására, hogy ugyanazon feladatokat különböző technikákkal oldják meg [37]. Ezt a gondolatot vittük tovább, kidolgozva egy olyan módszert, amely lehetővé teszi mind az öt technika módszeres, párhuzamos vizsgálatát [11, 63]. Célunk olyan felülnézet kialakítása a tanulók fejében, amelyből egyidejűleg láthatják a bemutatott stratégiákat. A módszer másik erőssége, hogy látva a teljes képet, a tanulók képesek lesznek felismerni az egyes technikák egymáshoz való viszonyát, és így a „nehezebb” stratégiák is elérhetőbbé válnak számukra.

Felülnézet alatt ebben a dolgozatban azt értjük:

1. „Egymás mellett” látjuk a vizsgálat tárgyát képező entitásokat.
2. Csak az látszik, ami a vizsgálat szempontjából lényeges.
3. Nyilvánvalóak a hasonlóságok és a különbségek, szembetűnők a kapcsolatok.

A felülnézet kialakításához szükséges egy úgynevezett „absztrakt platform”, amelyen az elemzett entitások úgy helyezhetők egymás mellé, hogy szembetűnővé váljanak a vizsgálat szempontjából lényeges tulajdonságok és kapcsolatok.

A visszalépéses keresés és az oszd meg és uralkodj technikák általában rekurzívan közelítik meg a feladatot: A feladatot visszavezetik hasonló, egyszerűbb részfeladatokra, majd ezeket további hasonló, még egyszerűbb részfeladatokra, egészen addig, míg triviális részfeladatokhoz nem jutnak. Ez a fajta lebontás azt feltételezi, hogy a feladat felépítésében fastruktúrájú legyen. A mohó, a dinamikus programozás és az elágazás korláttal technikák egyik közös vonása, hogy általában olyan feladatok esetében alkalmazzuk őket, amelyek döntéssorozatként foghatók fel. Ez ismét egy fastruktúrához, egy döntési fához vezet. Tehát az összes tárgyalásra kerülő módszert elsősorban olyan feladatok esetében használjuk, amelyek valamilyen értelemben fastruktúrával rendelkeznek. A technikák szempontjából ez azt jelenti, hogy mindegyik úgy tekinti a feladatot, mint egy fát. Ez a fastruktúra az az absztrakt platform, amelyen a technikák egymás mellé helyezhetők, és amely a felülnézet kialakításához szükséges.

A felülnézet módszer alkalmazásához egy tanmenetet javasolunk. A 2005/2006-os tanévtől a Sapientia Erdélyi Magyar Tudományegyetem marosvásárhelyi karán az alábbi tanterv szerint történik az algoritmustervezési stratégiák oktatása egy teljes féléven (14 hét) keresztül:

0. A rekurzió átisméltése, a fastruktúrák és azok bejárásainak bemutatása.
1. Egy feladaton keresztül, amely mindegyik technikával megoldható, általános és átfogó képet nyújtunk a technikákról (1. hét).
2. Bemutatjuk a visszalépéses keresés technikát (2.-3. hét).
3. Bemutatjuk az oszd meg és uralkodj technikát (4. hét).
4. Oszd meg és uralkodj vagy visszalépéses keresés (5. hét).

5. Bemutatjuk a mohó algoritmust (6. hét).
6. *Visszalépéses keresés és mohó algoritmus (7. hét).*
7. Bemutatjuk a dinamikus programozás módszerét (8.-10. hét).
8. *Oszd meg és uralkodj vagy dinamikus programozás (11. hét).*
9. *Mohó módszer vagy dinamikus programozás (11. hét).*
10. *Beágyazzuk az elágazás korláttal technikát az előbbi módszerek alkotta képbe (12.-13. hét).*
11. *Határesetek a programozási technikák világában (14. hét).*

Az egyes technikákat bemutató órákon hangsúlyozni kell, mit jelent az illető stratégia szempontjából fastruktúráként felfogni egy feladatot. A felülnézet órákra (dőlt betűvel jeleztük a tanmenet e leckéit) olyan feladatokat választunk, amelyek „megoldhatók” mindkét összehasonlítható módszerrel. Ezek azok az órák, amelyeken kihangsúlyozzuk a stratégiák közti hasonlóságokat, különbségeket és kapcsolatokat. Az utolsó fejezet olyan algoritmusokkal foglalkozik, amelyekben két vagy több technika jellegzetes vonásai keverednek.

A dinamikus programozás az egyik leghatékonyabb programozási technika, csakhogy a legelmélyültebb gondolkodást feltételezi. A nehézség többek között a dinamikus programozási feladatok sokszínűségében áll. A diáknak nagyon mélyen át kell látnia a dinamikus programozás alapelveit, hogy a legkülönbözőbb helyzetekben alkalmazni tudja. *A felülnézet módszer egy másik erőssége, hogy kiterjeszhető a dinamikus programozási feladatok területére, lehetővé téve ezek egyfajta osztályozását [61].*

Ugyanazt az absztrakt platformot használjuk, a feladatok mögött lévő döntési fát, és a feladatokat e döntési fa különböző típusai szerint osztályozzuk. Minden egyes feladatosztályhoz a dinamikus programozásnak mint algoritmustervezési stratégiának egy sajátos változata rendelhető. Az adott feladatnak a megfelelő osztályba sorolása után már sokkal elérhetőbb a technika valamelyik (az illető feladatosztályt megoldó) sajátos változatának alkalmazása. A felülnézet módszer kivetítése a dinamikus programozás területére azzal az előnnyel is jár, hogy árnyaltabbá teszi a dinamikus programozás és a mohó algoritmus, illetve az oszd meg és uralkodj technika közti különbségeket, hasonlóságokat és kapcsolatokat.

Bevezettük az első (minden döntéssel a feladat egy kisebb méretű hasonló feladattá redukálódik, amelyet az aktuális csomópont valamelyik részfája ábrázol), illetve második (minden döntéssel a feladat két vagy több kisebb méretű hasonló feladatra esik szét, amelyeket az aktuális csomópont megfelelő részfái ábrázolnak) típusú döntési fák elnevezést, valamint az összevont döntési fa (egy súlyozott irányított gráf, amelyet úgy kapunk, hogy egymásra csúsztatjuk a döntési fa identikus állapotokat képviselő csomópontjait) fogalmát.

A dinamikus programozással megoldható feladatok a szerint osztályozhatók, hogy a mögöttük meghúzódó döntési fa első vagy második típusú, illetve, hogy az első esetben az összevont döntési fa körmentes-e vagy sem, és amennyiben tartalmaz kört, van-e negatív éle vagy nincs.

Az 1. típusú döntési fájú feladatok esetében három dinamikus programozásos stratégiát különítettünk el:

1. *Topológikus sorrend alapú dinamikus programozás: Az összevont döntési fa körmentes.*
2. *Dijkstra algoritmus alapú dinamikus programozás: Az összevont döntési fa tartalmaz ugyan kört, de nincs negatív éle.*
3. *Belmann-Ford algoritmus alapú dinamikus programozás: Az összevont döntési fának van negatív éle is, de nincs a gyökből elérhető negatív összsúlyú köre.*

A 2. típusú döntési fájú feladatokhoz tartozó stratégia azonosítására az „optimális megosztás – optimális uralom” elnevezést javasoljuk [59].

Egy kísérletet végeztünk, amellyel lemértük, hogyan járul hozzá a felülnézet módszer és a javasolt tanmenet az algoritmustervezési stratégiák hatékony tanításához-tanulásához. A kísérletet a marosvásárhelyi Bolyai Farkas Elméleti Líceumban végeztük el a 2003/2004-es tanévben. Ebben az iskolában minden évfolyamon (IX.-XII.) három párhuzamosan működő informatika tagozatos osztály létezik. A hivatalos tananyag X. osztályban írja elő a technikák tanítását. A kísérletbe mindhárom osztályt bevontuk: *X.g*, *X.h*, *X.i*. A *h* és *i* osztályokat ugyanaz a tanárnő, a *g* osztályt pedig egy másik tanárnő tanította. A *g* és *i* osztályokat választottuk kísérleti csoportnak, a *h* osztályt pedig kontrollcsoportnak. A kísérleti csoportokban a tanárok felülnézet módszerrel, a javasolt tanmenet szerint oktattak, a kontrollosztályban pedig a klasszikus módon (a technikákat különálló egységekként kezelték). A kísérlet nyomán mindkét kísérleti osztályban az átlagok (az 1-10 skálán) körülbelül másfél jeggyel magasabbak lettek, ami szignifikáns különbséget jelent. A módszer kimagasló eredményekhez vezetett a tehetség gondozásban is. A 2002/2003-as tanévben az informatikai olimpia megyei szakaszán X. osztályban az első, a második, illetve a harmadik helyezett is abból az osztályból került ki, amelyben először alkalmazták a bemutatott módszert a programozási technikák oktatásában.

A felülnézet módszer a gondolkodási műveletek széles skáláját mozgósítja, ami elméleti magyarázattal szolgál a hatékonysága mellett. Jelentősen hozzájárul továbbá a diákok általános problémamegoldó gondolkodásának javulásához [20].

Mivel a különböző fogalmakat egymáshoz kapcsoltn mutatja be, ezért tartósabb mentális reprezentációt eredményez. Ez összhangban van azzal, ahogy a modern tudomány az emberi memóriát látja [37]. A tanmenet végére a diákok fejében az algoritmustervezési stratégiákról kialakult kép nemcsak átfogó, tiszta és mély lesz, hanem egyben tartósan eltárolt is.

3 A látás, hallás és kinesztetikus érzékelés bevonása elemi algoritmusok tanításába

Ebben a fejezetben felsorakoztatott kutatási eredményekből legalább öt érvelési vonal emelhető ki, amelyek mind ugyanahhoz a következtetéshez vezetnek: a tanítás-tanulás folyamata annál hatékonyabb, minél több érzékszervet vonunk be.

- Több érzék, több információt jelent.
- Az, hogy mindenkinek más a tanulási stílusa összecseng azzal a felismeréssel, hogy az egyes tanulóknak más-más lehet a domináns érzékszerve. Minél több érzéküket veszik igénybe a diákok, annál valószínűbb, hogy hatékonyan fogják találni az illető módszert [28, 31].

- Howard Gardner munkássága a nyolc „intelligenciáról”, és más kutatások arra is rámutattak, hogyan növeli a tanulás hatékonyságát, ha nemcsak a domináns érzékszervünkre támaszkodunk. Más szóval: az érzékszervek a kombinált alkalmazásukkal erősíthetik egymást [29].
- A kettőskódolás elmélete rámutat annak előnyére, ha a terhelés megoszlik a különböző érzékszervek között [39].
- Minél több érzékszervet vonunk be a tanulásakor, annál több úton érhető el az adott információ a visszaemlékezéskor.

A számítógépek programozásának elsajátítása legalább háromszorosan elvont folyamat. A programozási nyelv maga is absztrakció a valósághoz képest, a nyelvből megteremtett számítógépes program pedig kétszeres absztrakció a valósághoz viszonyítva. Mindezeket figyelembe véve, a számítógépek programozásának hatékony tanítása, illetve tanulása olyan módszereket és eszközöket feltételez, amelyeket átítat a szemléletesség didaktikai alapelve.

Számos új-mexikói iskolában a matematikaoktatást összekapcsolják a zene- és tánc tanítással. Tapasztalataik azt igazolják, hogy e látszólag teljesen különböző két terület sokat nyújthat egymásnak [30].

Bármely algoritmus esetében beszélhetünk annak ciklusvázáról (csontvázáról), amely alatt az algoritmus ciklusutasításainak a szerkezetét értjük. A ciklusok magját képező utasításokat pedig úgy tekinthetjük, mint az algoritmus „húsos részeit” [64]. Az alábbi kétlépéses módszert ajánljuk elemi algoritmusok tanításához, illetve tanulásához:

1. *A feladat elemzési szakaszában a megoldási algoritmus ciklusvázának a meghatározása.*
2. *A ciklusváznak a megfelelő utasításokkal való feltöltése.*

Kidolgoztunk egy szoftvert, amely javítja a diákok ciklusváz-felismerő képességét. A szoftver célja, lehetővé tenni a látás, hallás, valamint a kinesztetikus érzékelés bevonását e folyamatba. Az alkalmazásnak négy fő modulja van. A `cod_creator`, `cod_beautifier`, `cod_buherator` és a `run_code` modulok.

A `cod_creator` modul különböző ciklusvázú programok automatikus létrehozását teszi lehetővé, és két üzemmódban használható:

- *A ciklusváz paramétereinek megadásával: Megadhatjuk, hogy hány első szintű ciklust szeretnénk, és hogy ezekbe hány második, illetve harmadik szintű ciklus legyen beágyazva. Az is beállítható, hogy az egyes ciklusok hányszor legyenek végrehajtva.*
- *A ciklusváz „bedobolásával”: „Begépeljük” a program ciklusvázát, mintha eldobalnánk a ritmusát, mintáját. A módszer alkalmazásának ennél a pontjánál kerül sor a kinesztetikus érzékelés bevonására.*

A `code_beautifier` modullal bármely C/C++ program forráskódja megszépíthető, azaz a szoftver oly módon rendezi át a kódot, hogy jól nyomon követhető legyen a ciklusváza. Különösen a látás bevonása szempontjából fontos ez a művelet.

A `code_buherator` modul – átírva a forráskódot - az algoritmus minden ciklusmagjába hangokat megszólaltató, illetve a programfutást késleltető rutinokat épít be. Olyan,

mintha kihangosítanánk az algoritmus ciklusvázát. A ciklusmagok minden egyes lefutásakor megszólal egy zongorahang. A külső ciklusok mélyebb, a belsők magasabb hangokon „szólalnak meg”. A különböző szintű ciklusok különböző mértékű késleltetésével elértük, hogy a külső ciklusok lefutását jelző hangjegysorok hangjegyei kisebb, a belsőkéi pedig nagyobb frekvenciával kövessék egymást.

A run_code modul párbeszéd ablakában megjelenik a vizsgálat alatt álló algoritmus megszerkesztett C/C++ kódja. A Run gomb lenyomására a háttérben elindul a cod_buherator modul által átdolgozott (kihangosított és lelassított) program futtatása. Miközben a tanuló hallja – futó program révén - az algoritmus ciklusvázát szemléltető hangjegysort, a run_code ablakban szemével követheti, hol tart a végrehajtás.

A szoftver alkalmazásához egy tanmenetet is javasolunk:

- *A tanár különböző ciklusvázakat „hallgattat meg” a tanulókkal, miközben a diákok szemükkel követik az algoritmus végrehajtását.*
- *A tanulókat megkérjük, hogy próbálják meg hallás után felismerni a „lejátszott” ciklusvázakat.*
- *„Kihangosított” alapvető elemi algoritmusok végrehajtásának nyomon követése füllel és szemmel.*
- *A diákok különböző ciklusvázakat „gépelnék be”. Miközben a tanulók ujjai ritmikus, zongorázásra emlékeztető mozdulataival újabb és újabb ciklusvázakat gépelnék be, végül már bennük cseng az algoritmusok ritmusa. A módszer e szakaszát elméleti órákon is alkalmazhatjuk, még ha nem is áll rendelkezésre számítógép és a szoftver. Először a tanár, majd a diákok, a kezeiket és lábaikat használva, szó szerint eldobolhatják a katedrán, illetve a padjaikon a ciklusvázak ritmusát.*
- *Tipikus hibákat tartalmazó algoritmusok ciklusváz-hibáinak hallás utáni felismerése.*

A bemutatott módszer és szoftver hatékonyságának empirikus bizonyítása végett egy kísérletet végeztünk. A marosvásárhelyi Bolyai Farkas Elméleti Líceum két IX. osztályát (h és g) vontuk be. Mindkét osztály a 2005/2006-os tanévben kezdett programozást tanulni C/C++ nyelven. A kísérletre az első félév végén került sor. A két osztály ugyanazon tanterv szerint tanulja a programozást, de más-más tanár tanítja őket. Mindkét osztályt két-két egyenlő erősségű csoportba (egy kísérleti és egy kontroll) osztottuk. A kísérleti csoportokban két héten át a fentebb leírt tanmenet és kétlépéses módszer szerint történt az oktatás. A kontrolles csoportokban a hagyományos módszereket használták anélkül, hogy törekedtek volna az érzékszervek bevonására, és persze anélkül, hogy használták volna a szoftvert. A kísérlet befejeztével felmérőt írtunk. Az alábbi szignifikáns eredmények születtek:

- *A h osztály eredményei azt mutatják, hogy az új módszer alkalmazása várhatóan (0,9804 valószínűséggel) 0,76 jeggyel fogja emelni az osztályátlagot.*
- *A g osztály esetében ez az érték 1,35, 0.9897 valószínűséggel.*
- *Ha a két mintát együtt vizsgáljuk, akkor 1,05 értékű átlagos javulásra számíthatunk 0,9957 valószínűséggel.*

A szoftver használata lehetővé teszi, hogy a diákok úgymond le tudják fogni az algoritmusok pulzusát. Tényleg egész testtel való tanulást biztosít. A tanulók láthatják, hallhatják és – a ciklusvázak bedobolása révén - érezhetik is az algoritmusok lüktetését. Mindez egyértelműen hatékonyabb tanulást jelent. A több érzékszerv nemcsak több

információt, jobb megértést és jobb memorizálást jelent, hanem hozzájárul a más-más domináns érzékszervű diákok „esélyegyenlőségéhez” a tanítás-tanulás folyamatában.

Az algoritmusnak a kód alapján történő elemzésekor a ciklusszerkezetek mint szöveges kódolású információk, a bal agyfélteke dolgozza fel. Ezzel párhuzamosan a különböző hangmagasságú és frekvenciájú hangjegyek, valamint az aktuális programsornak a hangjegysorral összehangolt kiemelése révén a jobb agyfélteke jut információhoz az algoritmus ciklusvázának felépítéséről. E kettős kódolás nemcsak hatékonyabb tanulási folyamatot, hanem tartósabb mentális reprezentációt is eredményez.

Ha megkérjük a „zenész diákokat” (akik játszanak valamilyen hangszeren), hogy különböző ciklusváz-ritmusokkal készüljenek a „következő órára”, amelyeket az osztálynak majd hallás után kell azonosítani, akkor ez gazdagítani fogja az órát affektív síkon is.

4 „Legyél te is eminens” - Értékelési módszer és szoftver

A modern felfogásban az értékelés már nemcsak nevelési és oktatási módszer, hanem rendszerszabályozó elem. Az értékeléssel szerzett információk felhasználásával szabályozható mind a tanítói, mind a tanulói tevékenység. Előnyös, ha gyakran keletkezik visszacsatolt információ, hiszen így a tanítás-tanulás folyamata finomabban szabályozható [44, 45, 46].

Ma már az oktatásban alkalmazott módszerek sokfélék. Részben tovább élnek a hagyományos értékelési formák, részben kibővültek a tudományos kutatás módszereivel, és az azokhoz kötött módszertani követelményekkel (objektivitás, validitás, megbízhatóság). A számítógépek, valamint a számítógépes hálózatok megjelenése a tesztre irányították a figyelmet, mint ami kiküszöböli a hagyományos módszerek számos hiányosságát, de sajnos újakat teremt.

Az ideális tanítás-tanulás folyamatot gyakorlatiasnak és egyben élvezetesnek kell lennie. [6] A kérdés, amelyet elemeztünk az, hogy ki lehetne-e terjeszteni e comeniusi alapelvet az értékelés fázisára is?

Törekvésünk semmiképpen sem számít úttörő munkának, hiszen számos kutató és pedagógus ért már el szép eredményeket e területen. A számítógépek, majd a számítógépes hálózatok megjelenése új lehetőségeket teremtett. A másik segítségnek éppen a média bizonyulhat. Számos olyan TV-játék vált híressé, amely a játékon túl arra hivatott, hogy oktasson, és ezzel egy időben lemérje a játékosok tudásszintjét. E TV-játékok sikere azt mutatja, hogy nem lehetetlen izgalmassá tenni a feleltetést sem. Persze, az osztálytermek és a stúdiók között van egy alapvető különbség: a játékos legfeljebb nem nyer semmit, de a diák számára a felelés rossz jeggyel, a vizsga pedig bukással is végződhet. Ezért e TV-játékok által inspirált didaktikai szoftvereket inkább a tanult anyagrész átismétlésére használják, mint vizsgáztatásra.

Ilyen irányú sikeres alkalmazások a Jeopardy népszerű TV-játék osztálytermi implementációi [49, 50]. Másik népszerű TV-játék a „Legyen ön is milliós”. Bár nagyon sok pedagógust ihletett már meg e TV-játék is, és dolgoztak ki számítógépes tananyagot ismétlő, illetve feleltető szoftvereket, úgy gondoljuk, hogy az általunk

készített „Legyél te is eminens” hálózati feleltető szoftver sok szempontból úttörőnek számít e területen, és egyben az említett TV-játék egyik legsikeresebb osztálytermi implementációjának tekinthető. Sok alkalmazás, bár jól szemlélteti, hogy különböző játékok osztálytermi implementációja miként növelheti a tanítási-tanulási folyamat hatékonyságát, mégis messze kihasználatlanul hagyja a számítástechnika, informatika biztosította lehetőségeket [51, 52, 57].

A „Legyél te is eminens” szoftver felhasználóbarát hálózati alkalmazás, amihez interneten keresztül is kapcsolódhatunk. A szoftver három felhasználói minőségben használható: feleltető tanárként, felelő diákként vagy „néző diákként”. A TV-játéktól eltérően több felelő is lehet.

A szoftver különböző üzemmódokban működtethető. Implementáltuk a hagyományos tesztet is, amikor a diákok egymásután kapják a kérdéseket, és a szoftver végül összesíti a jó válaszokat. Az alapértelmezett „eminens” üzemmódban a diákoknak kilenc kérdést kell megválaszolniuk. Ha minden kérdésre helyesen válaszolnak, akkor a hivatalból kapott 1 ponttal együtt ez 10-es osztályzatot jelent nekik (összhangban a Romániában használatos 1 – 10 jegyrendszerrel). Két küszöbértéket építettünk be: az 5-ös és 8-as osztályzatokat. A kérdéseket három nehézségi kategóriába soroltuk. A szoftver az első négy kérdést véletlenszerűen az első (legkönnyebb) kategóriából választja (5-ös osztályzatig), a következő hármat a második kategóriából (8-as osztályzatig), végül az utolsó két kérdést a harmadik kategóriából (10-es osztályzatig). A felelő diákok bármikor megállíthatják a saját kiértékelésük folyamatát, ha elégedettek az összegyűjtött pontszámmal, és nem biztosak a következő kérdésre adandó válaszukban. Ilyenkor átkerülnek a nézők közé. Téves válasz esetén osztályzatuk visszaesik a legközelebbi küszöbértékre. Minden felelőnek rendelkezésére áll három segítség: kérhet felelést, kérheti a nézők válaszainak a statisztikáját, illetve chat-elhet valamelyik nézővel.

A „Legyél te is eminens” szoftver jól használható vizsgáztatásra olyan tantárgyak esetén, amelyek lehetővé teszik a tesztrendszeres értékelést. Az alábbi vizsgamenetet javasoljuk:

- *A szoftvert kétszer egymás után eminens üzemmódban futtatjuk le. Először az osztály egyik feléhez tartozó diákok a felelők és a többiek a nézők, majd fordítva.*
- *Ezután a program hagyományos üzemmódban fut le.*
- *Végül a két osztályzat közül a jobbat írjuk be a vizsgajegyzőkönyvbe.*

Azzal, hogy a vizsga a hagyományos teszttel fejeződik be (ami már általánosan elfogadott, és amire mindenképpen sor kerülne) és a jobb osztályzatot írjuk be, biztosítani tudjuk, hogy az eminens üzemmódból (mint a TV-játékból) csak nyerhessen a diák.

Azért, hogy osztálytermi feleltetésre is használható legyen a szoftver, kidolgoztunk egy ötvözött üzemmódot is. Ilyenkor az első négy kérdést hagyományos tesztrendszerben kapják a diákok. A negyedik kérdés megválaszolását követően a szoftver kijelzi az elért pontszámot, ami első küszöbnek számít az ebben a pillanatban induló eminens teszthez (a második küszöb automatikusan három ponttal magasabb). A „kiesett” felelők nézőkként továbbra is részt vesznek a tesztben: „post eminens” teszt.

A „Legyél te is eminens” didaktikai szoftver megírásával az volt a célunk, hogy olyan módszert és eszközt biztosítsunk az értékelés folyamatához, amely: képes izgalmat vinni a feleltetésbe; aktivizálja az egész osztályt; a mérésen túl tanít és nevel; rendelkezik a teszt előnyeivel a hagyományos módszerekkel szemben; maximálisan kihasználja a

számítástechnikai nyújtotta jelenlegi lehetőségeket; olyan adatbázist eredményez, ami mélyreható pedagógiai kutatásokat tesz lehetővé. Arra is odafigyeltünk, hogy az eredeti ötletek mellett beépítsük a szoftverbe a már létező alkalmazások erősségeit. Az alábbi összefoglalása a módszer előnyeinek azt igazolja, hogy elértük célunkat. (Reméljük, hogy a további kísérletek és kutatások megerősítik majd ezt a következtetést.)

- Versenyhelyzetet alakít ki és izgalmas, amik további motivációforrások a tesztre való készüléskor.
- A feleltetési üzemmód azt a tanárt „modellezi”, amelyik tekintettel van a kezdeti fokozott stresszre, segítséget ad, amikor a felelő elakad, és kész figyelembe venni azt a tudást is, ami a feleltetés során nem kerül felszínre.
- Egyszerre több diák is „felelhet”. Ha az első négy kérdésre egy-egy percet szánunk, a következő háromra kettőt-kettőt, és az utolsó kettőre egyenként két és fél percet, akkor 15 perc alatt be is fejeződik a feleltetés.
- A szoftver lehetővé teszi az eminens teszt erősségeinek kamatoztatását vizsgáztatás alkalmával.
- Az egész osztályt foglalkoztatjuk, hiszen mindegyik tanuló képernyőjén megjelennek a kérdések, és válaszolniuk kell. Továbbá, a segítség-opciók révén, az egész osztály aktívan részt vesz a felelők megmértésében. Az eminens üzemmód „törvényes” keretet biztosít ahhoz, ahogy az osztály a felelők segítségére siethet. Mindezek plusz motivációforrást jelentenek a rendszeres tanuláshoz.
- A szerver folyamatosan megjeleníti a tanulók tevékenységét. Ez kizárja annak lehetőségét, hogy észrevétlenül maradjon a felelést megúszó tanulók felkészületlensége, vagy a felelésről lemaradók felkészültsége.
- A „felelés” befejeztével azonnal számos statisztikai kimutatás áll a tanár rendelkezésére nyomtatható formában.
- Mivel a kérdésszerkesztő modul önállóan is használható, a tanulók bevonhatók a tesztkérdések összeállításába, ami számos előnnyel jár.
- Elektronikus jegyzőkönyv készül feleltetések minden mozzanatáról. Ezen adatbázis lehetővé teszi a tanítás-tanulás folyamatának egy későbbi, igen átfogó tanulmányozását.
- Tekintettel e tesztforma kis időigényére, gyakori alkalmazásával lehetővé válik a tanítási-tanulási folyamat finomabb szabályozása.
- Az értékelés mellett nemcsak a tanítást és tanulást teszi lehetővé, hanem a nevelést is.
 - Olyan tulajdonságokat alakít ki a tanulóknál, mint az önzetlenség, segítőkészség, együttérzés. A néző diákok úgy mutatják ki e tulajdonságokat, hogy teljesen beleélik magukat a kérdésekbe, mintha ők felelnének, hogy a lehető legjobbat tudják nyújtani, ha a segítségüket kéri.
 - Előtérbe kerül az önbizalom, illetve az osztálytársakban, valamint az osztályban, mint egészben való bizalom kérdése.
 - Hozzájárulhat az osztályszellem megerősödéséhez.
- A rossz válasz szigorú büntetése, a megállás lehetősége, a küszöbértékek, valamint a segítségforrások együttes jelenléte gyakran olyan választások elé állítja a felelőket, amelyek próbára teszik, de egyben csiszolják is a bölcsességüket.

Irodalomjegyzék

- [1] R. Kotulak, *Inside the Brain*, Andrews McMeel, Kansas City, 1996
- [2] R. L. Atkinson, R. C. Atkinson, *Pszichológia*, Osiris, Budapest, 1997
- [3] P. Drucker, *Knowledge-worker Productivity: The Biggest Challenge*, California Management Review 41, S. 79–94, 1999
- [4] P. Drucker, *Beyond the Information Revolution*, The Atlantic, 1999. 12
www.theatlantic.com/issues/99oct
- [5] F. Schaffhauseri, *Társadalom-iskolaelmélet – iskolaszervezet*, Bábosik I., R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [6] Comenius, *Orbis sensualium pictus*, 1653
- [7] Z. Krygowska, *A matematikadidaktika jelenkori kutatásainak főbb irányzatai és problémái*, Dydaktyka Matematyki, Varsó, 1/1982
- [8] H. Braun, *Reflections on the future of assessment*, Konferenciaelőadás, EARLI (European Association for Research on Learning and Instruction), Assessment and evaluation, 2000. szeptember 13–15., Maastricht
- [9] A. Stephen, *Hogy kell a tanulás eredményeit használni? „Csoportkapcsolatok”*, Bolognai Konferencia, Budapest, 2006. január 25.
- [10] K. Garnitschnig, G. Khan-Svik, *Aktív tanulás. A sikeres oktatásszervezés aspektusai*, Bábosik I., R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [11] Kátai Z., *Algoritmustervezés felülnézetből*, Sapientia Kiadó, Kolozsvár, 2006
- [12] B. S. Bloom, *Taxonomie von Lernzielen im kognitiven Bereich*, Weinheim, Base, 1972
- [13] B. S. Bloom, D. R. Krathwohl, *Taxonomie von Lernzielen im affektiven Bereich*, Weinheim, Base, 1975
- [14] E. Persy, *Motiváció a tanítás-tanulás folyamatában*, I. Bábosik, R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [15] E. Ch. Wittmann, *Grundfragen des Mathematikunterrichts*, Vieweg, 1981
- [16] L. Eliot, *What's Going On In There? How the Brain and Mind Develop in the First Five Years of Life*, Bantam Doubleday Dell Pub, 2000
- [17] T. H. Cormen, C. E. Leiserson, R. L. Rives, *Introduction to Algorithms*, by The Massachusetts Institute of Technology, 1990, 266-270, 287-289.
- [18] I. Odagescu, C. Copos, D. Luca, F. Furtuna, I. Smeureanu, *Metode și tehnici de programare*, Intact, București, 1994, 95-108.
- [19] R. Andone, I. Garbacea, *Algoritmi fundamentali. O perspectivă C++*, Libris, Cluj-Napoca, 1995, 185-187, 219-221.
- [20] Revákné Markóczi I., Máthé J., *A természettudományos problémamegoldó gondolkodás fejlesztése a középiskolában*, Új Pedagógiai Szemle, 2002/10
- [21] J. Dewey, *How we think*, New York, Holt, Rinehart and Winston, 1910
- [22] J. Rossman, *The psychology of the inventor*, Washington, Inventor's Publishing Co, 1931
- [23] A. Osborne, *Applied imagination*, New York, Scribner, 1963
- [24] Pólya Gy., *A problémamegoldás iskolája*, Budapest, Tankönyvkiadó, 1979
- [25] Lénárd F., *A problémamegoldó gondolkodás*, Budapest, Akadémiai Kiadó, 1963
- [26] J.Holt, *How Children Fail*, New York, Pitman, 1994.
- [27] *Research on the Brain*, Classroom Compass, 1997, Volume 3, Number 2

- [28] D. Dickinson, Washington State Art Commission, winter 2003 issue
- [29] H. Gardner, *Frames of Mind*, 1985
- [30] *Rhythm of Mathematics*, Classroom Compass, 1998, Volume 4, Number 2
- [31] M. D. Levine, *A Mind at a Time*, Simon & Schuster, 2002
- [32] H. C. Hughes, *Sensory exotica – a world beyond human experience*, Bradford Book, 2001
- [33] Bábosik I., R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [34] D. Laird, *Approaches to training and development*, Addison-Wesley, Reading, Mass., 1985
- [35] *Navy Instructor Manuals*, www.tpub.com/content/administration
- [36] *Out of Memory*, <http://library.thinkquest.org/C0110291>
- [37] <http://www.cis.upenn.edu/~matuszek/cit594-2004/Lectures/44-dynamic-programming.ppt>
- [38] P. Russel, *The Brain Bokk*, Dutton, New York, 1979
- [39] Komenczi B., *Orbis sensualium pictus. Multimédia az iskolában*, Iskolakultúra, 1997/1
- [40] Paivio, *Mental representations. A dual coding approach*, New York, Oxford University Press, 1986
- [41] D. L. Nelson, V. S. Reed - J. R. Walling, *Pictorial superiority effect*, Journal of Experimental Psychologi: Human Learning and Memory, 1976/2, 523-528
- [42] M. Paechter, *Sprechende Computer in CBT: eine didaktische Konzeption*, Arbeiten aus der Seminar für Pädagogik, Bericht 1/93. TU, Braunschweig, 1993
- [43] M. Pyter, *Textrepresentation in Hypertext. Empirische Analyse von visuellen versus audiovisuellen Sprachdarbietungen in Hypertext*, Papier zum Kongress der DGPs, Hamburg, 1994
- [44] Báthory Z., *Értékelés a pedagógiában*, Pedagógiai Szemle, 1972/3, 212-220
- [45] Golnhofer E., *A pedagógiai értékelés*, Falus Iván (szerk.), Didaktika. Nemzeti Tankönyvkiadó, Budapest, 392-414.
- [46] H. Brückner, *Számítógépek az oktatásban, Számítógépes oktatás*, KSH Nemzetközi Számítástechnikai Oktató és Tájékoztató Központ, Budapest 1978
- [47] Bálya D., *Az informatika kihívása a teszt-technológiában*, TDK dolgozat, BME TIO 1997
- [48] A. Binet, *A propos de la mesure de l' intelligence*, Paris, 1905
- [49] J. Benek-Rivera, V.E. Mathews, *Active learning with Jeopardy: Students ask the questions*, Journal of Management Education, 28 (1), 104-118, 2004
- [50] J. J. Grabowski, Michelle L. Price, *Simple HTML Templates for Creating Science Oriented Jeopardy! Games for Active Learning*, Department of Chemistry, University of Pittsburgh, 2002
- [51] D. Marsh, *Who wants to be a maths millionaire?*, <http://ferl.becta.org.uk>, 26 April 2002
- [52] J. J. Cochran, „*Who Wants To Be A Millionaire: The Classroom Edition*”, INFORMS Conference, San Antonio, TX., 2000
- [53] G. Brown, M. Atkins, *Effective Teaching In Higher Education*, Methuen, London, 1988
- [54] J. Hartley, I. K. Davies, *Note Taking: A Critical Review*, Programmed Learning and Educational Technology, Vol. 15, 207-224, 1978
- [55] W. J. McKeachie, *McKeachie's Teaching Tips: Strategies, Research and Theory for College and University Teachers*, Houghton-Mifflin, Boston, MA, 1999

- [56] T. Sutherland, C. Bonwell, *Using Active Learning In College Classes: A Range Of Options For Faculty*, New Directions for Teaching and Learning, Vol. 67, 1996
- [57] J. M. Collins, *Who wants to be a millionaire? An educational game for learning enhancement*, Innovative Teaching Practices at Bloomsburg University, Teaching and Learning Enhancement Center, 2004
- [59] Kátai Z., *Dynamic programming and d-graphs*, Studia - Informatics, Cluj, 2006
- [60] T. Angelo, P. Cross, *Classroom assessment techniques: A handbook for college teachers* (2nd edition), San Francisco, 1993
- [61] Kátai Z., *Dynamic programming strategies on the decision tree hidden behind the optimizing problems*, Informatics in Education, Institute of Mathematics and Informatics, Lithuania, 2006 (elfogadva)
- [62] Kátai Z., *Involvement of the sight, hearing, touching and moving in teaching basic algorithms*, Computers & Education, 2006 (*benyújtva*) (Társ szerzők: Nyakóné Juhász Katalin, Adorjáni Alpár Károly)
- [63] Kátai Z., "Upperview" algorithm design in teaching computer science in high schools, Teaching Mathematics and Computer Science, 3 (2005) 2
- [64] Kátai Z., *Programozás C nyelven*, Sapientia-EMTE, Scientia Kiadó, Kolozsvár, 2004

Publikációs jegyzék

Referált nemzetközi folyóiratban megjelent cikkek:

- [1] "Upperview" algorithm design in teaching computer science in high schools, Teaching Mathematics and Computer Science, 3 (2005) 2, 221-241
- [2] Dynamic programming and d-graphs, Studia Universitatis Babeş-Bolyai - Series Informatica, 2006 (elfogadva)
- [3] Dynamic programming strategies on the decision tree hidden behind the optimizing problems, Informatics in Education, Institute of Mathematics and Informatics, Lithuania, 2006 (elfogadva)
- [4] Involvement of the sight, hearing, touching and moving in teaching basic algorithms, Computers & Education, 2006 (benyújtva) (Társszerzők: Nyakóné Juhász Katalin, Adorjáni Alpár Károly)

Lektorált egyetemi jegyzetek:

- [1] Programozás C nyelven, Sapientia-EMTE, Scientia Kiadó, Kolozsvár, 2004
- [2] Algoritmustervezés felülnézetből, Sapientia-EMTE, Scientia Kiadó, Kolozsvár, 2006

Tudományos konferenciákon elhangzott előadások:

- [1] Hogyan tanítsuk a programozási technikákat, XIV Számítástechnika az oktatásban nemzetközi konferencia, EMT, Kolozsvár 2004. március 25-28.
- [2] Programozási technikák felülnézetből, XV Számítástechnika az oktatásban nemzetközi konferencia, EMT, Kolozsvár 2005. március 17-20.
- [3] Algoritmus tervezés (Didaktikai szempontok), Informatika a felső oktatásban nemzetközi konferencia, Debrecen, 2005, augusztus 24-26.

Egyéb publikációk:

Tudományos konferencia kiadványok:

- [1] Hogyan tanítsuk a programozási technikákat?, XIV Számítástechnika az oktatásban nemzetközi konferencia, EMT, Kolozsvár, 2004, 50-56
- [2] Programozási technikák felülnézetből, XV Számítástechnika az oktatásban nemzetközi konferencia, EMT, Kolozsvár, 2005, 139-146

Egyéb tudományos folyóiratokban megjelent cikkek:

- [1] Programozási technikák felülnézetből (I rész), Firka, 2003/2004-4, EMT, Kolozsvár, 145-148
- [2] Programozási technikák felülnézetből (II rész), Firka, 2003/2004-5, EMT, Kolozsvár, 190-192

Szakmaspecifikus produktumok

- [1] "Legyél te is eminens", Hálózati értékelési szoftver, (Társszerző: Máthé Szabolcs)
- [2] "Elemi algoritmusok anatómiája", Didaktikai szoftver, (Társszerző: Adorjáni Alpár Károly)

1 Introduction

Since the second half of the 20th century certain didactical connections of education have been highlighted and several didactical points of view have received a different new light. This was due first to the invention of the computer, the internet and in the recent years to the results obtained in the field of brain research. We can call all the three as revolutionary, considering their effect on education, respectively in the role of education in the society.

Because of the appearance of the computer and the development of the internet, new teaching, learning and evaluation forms have appeared in the centre of attention. Concerning the human brain, the researches have revealed that it is a very flexible biological structure. [1] Two main factors define the way our brain develops throughout our life: what we let into our brain with our senses, and what we choose as subject of our thinking. Both factors are favoured in the process of teaching-learning.

The effect of the brain research and of the computer on the process of education becomes evident as well-known educational methods are stressed or new possibilities for their use appear. For example Comenius [6] made the following statement: "Learning should be completely practical, completely entertaining, ..., such that school really could become the place for games, thus the prelude for the whole life." Another well-known, and by the modern research completely supported basic educational principle is that the more we involve the senses into learning, the more effective it becomes. The appearance of the computers, the development of the computer sciences has opened new perspectives in the efficient use of the basic principles stated several hundreds of years ago.

The present and yet to come spiritual, economic and political transformations of the developed societies are causing unequivocal alterations in the school system as well. In the age of knowledge, the centre of attention is shifted from technology to information. [3, 4] In the light of these changes, the Conference of Bologna is considered a landmark. The basic building blocks of the reform of Bologna are the learning outcomes, those skills, abilities, knowledge and competencies the individual acquires until the end of the learning process. The real turning point is the ability to transform the information into knowledge and apply it in a creative way. [5] The new guidelines drafted also in the reform of Bologna stress the tight connection between teaching-learning-assessment, as those being the interlinked elements of a feedback process. [9]

In the same time with these revolutionary headlines, respectively changes at the level of society, we are witnesses of a paradigm-change as well: from teaching towards learning, from a teacher oriented education towards one where the student is in the centre. With this paradigm-change, the role of the teacher in the process of education has changed. The modern teacher is not any more the centre of an epistemological knowledge, but the assistant of the learning process and its promoter. Today one of the basic factors of an efficient education is unanimously the fact that it concentrates on the conditions necessary for the optimal learning process of the students. The importance of the active learning is based on the realization that the intelligence is inseparably linked with activity (psychological aspect) and that the activity of the body is the basic condition for the growth and interconnection of the nerve cells (physiologic aspect). If we ensure top priority for the active drill with learning contents, this will have an effect on the planning

of the educational process, the control of the achievements and their assessment. All this has to be adjusted to the personality of the learner. [10]

In order that the above drafted guidelines display their generous effect on the students, more than the reevaluation of the learning methods is necessary. It is of utmost importance that the teachers have such concrete educational methods and tools, which translate into practice the results of the latest researches and exploit the possibilities given by computers.

Although the guidelines are clearer and clearer and the possibilities easier to reach, the concrete educational methods and efficient tools are often missing. The present work is trying to make steps forward in these fields of the teaching of computer sciences. We have relied on 15 years of experience in teaching computer sciences and on researches in this field while working on the present study.

In this work we are going to present three methods, which can help to increase the efficiency of all the three elements of the process of teaching-learning-assessment. We have developed the first one ("**Upperview**" **Algorithm Design**) for the teaching of a specific field of the computer sciences, the teaching of algorithm design strategies. In order to ensure the necessary content for applying the method, the author has edited – with the same title – a book as well. [11] Though it is first of all a *teaching* method, it stresses the activating of the learner, too.

The second method (**Involvement of the sight, hearing and the phenomenon of kinaesthesia in the teaching-learning process of elementary algorithms**), is meant to increase the efficiency of *teaching and learning* the elementary algorithms. For this method, we have developed software as well, which makes the use of the method possible, as an educational tool, for the teacher, but also for the student, in the process of teaching and learning. As the name of the method suggests, the students are almost involved in the learning with their whole being. The students can use this software as an e-learning tool at home, independently.

Finally with the third method, which is built around a network software ("**Who wants to be eminent**"), we would like to increase the efficiency of the assessment phase. This is test-based software, usable within almost any subject and makes it possible to have an overview of the process of teaching-learning-assessment as a whole.

2 „Upperview” Algorithm Design

„To teach means scarcely anything more than to show how things differ from one another in their different purposes, forms, and origins. . . . Therefore, he who differentiates well teaches well.” [6] In this chapter we are going to present a teaching – learning method and suggest a syllabus that help the high school students look at the algorithm design strategies from a so called „uppreview”: greedy, backtracking, divide and conquer, dynamic programming, branch and bound. The goal of the suggested syllabus is, beyond the presentation of the techniques, to offer the students a view that reveals them the basic and even the slight differences and similarities between the strategies. In consensus with the Comenius principle this is essential, if we want to master this field of programming.

Several issues of the bibliography compare the techniques. For example, the authors Cormen, Leiserson and Rivest in their book „Introduction to Algorithm” [17] compare the dynamic programming and the greedy strategy. Some books (for example [18]) discuss how the backtracking and greedy techniques can complement each other. In other books (for example [19]) we can find a comparing analysis concerning the strategies of divide and conquer and dynamic programming.

In the present chapter we have developed this idea and we are presenting a method which makes it possible to discuss uniformly all the above-mentioned techniques. [37] We tried to establish such an „upperview” where each technique can be seen in the same time next to each other. [11, 63] By this means it becomes possible to integrate all the four techniques into a frame which forms a whole. If the students recognize the position of certain techniques related to the others, then the so called „more difficult” strategies become available for them.

We use the notion of „upperview” in the following sense:

- 1. We can see the entities being analyzed „next to each other”.*
- 2. Only those elements can be seen which are essential for the analysis.*
- 3. The similarities and differences are obvious, the connections are striking.*

In order to carry out an „upperview”, a so called „abstract platform” might be necessary, where the entities being analyzed can be laid down next to each other in such a manner that the features and connections essential for the analysis become obvious.

The backtracking and the divide and conquer techniques usually approach the problem in a recursive way. The sequence of ideas in a recursion is the following: How can the problem be reduced to similar, simpler subproblems, then later again reduce these ones to similar, even simpler subproblems, until we get trivial subproblems? This type of break-down presumes that the problem – in its construction – should have the structure of a tree. The common characteristic of the greedy and dynamic programming techniques is that we usually apply them for problems that can be regarded as sequences of decisions. This leads again to a tree structure, a decision tree. Considering all this, we can say that we apply all the techniques we are going to present especially in the case of problems which in some consideration have the structure of a tree. From the point of view of the techniques this means that each one considers the problem as a tree. Well, this common tree structure is that common plane or abstract platform – necessary for the „upperview” – where the techniques can be laid next to each other.

As an application of the „upperview” method we suggest the following syllabus:

- 0. Revision of the recursion, presentation of the tree structures and their traversal.*
- 1. Through a demo problem solvable with each of the strategies, we offer a general and comprehensive image of the techniques.*
- 2. We present the backtracking technique.*
- 3. We present the divide and conquer technique.*
- 4. Divide and conquer and backtracking from „upperview”.*
- 5. We present the greedy technique.*
- 6. Backtracking and greedy from „upperview”.*
- 7. We present the technique of the dynamic programming.*

8. *Divide and conquer and dynamic programming from „upperview”.*
9. *Greedy and dynamic programming from „upperview”.*
10. *We embed the “branch and bound” technique in the picture given by the above-mentioned methods.*
11. *Boundary cases in the world of programming techniques.*

At the classes presenting certain techniques we have to emphasise what it means from the point of view of the respective strategy to perceive a problem like a tree. For the „upperview” classes we choose such problems, which can be „solved” with each of the respective techniques. These are the classes where the similarities, differences and connections between the strategies are stressed.

In this chapter we also present a study and classification of the dynamic programming strategies. By presenting the characteristics of certain dynamic programming strategies on the decision tree hidden behind the optimizing problems, we offer such a clear tool for their study and classification, which can help in the comprehension of the essence of this programming technique. [61]

We introduced the notions: I. type decision tree (By each decision the problem is reduced to a similar problem of smaller size, represented by one of the subtrees of the current node); II. type decision tree (By each decision the problem is divided into two or more smaller sized subproblems, represented by the corresponding subtrees of the current node); contracted decision tree (We obtain this data structure, if we overlap the nodes representing the identical states of the tree.)

In case of I. type decision tree we have distinguished three different dynamic programming strategies, for each of which there are well-known graph-algorithms:

1. *The contracted decision tree is cycle free. In this case, there is an algorithm based on the topological order of the nodes.*
2. *The contracted decision tree contains circles, but it has no negative edges. For this case, there is the Dijkstra algorithm, which determines the minimal weight paths with a priority queue in the increasing order of the optimal values.*
3. *The contracted decision tree has negative edges, but has no negative total weight circle reachable from the root. The Belmann-Ford algorithm solves this problem.*

In case of II. type decision tree we have called the specific dynamic programming strategy „Optimal division – Optimal conquest”. [59]

In the following, we related about an experiment where we assessed how the presented method and the suggested syllabus contributed to the more efficient teaching of the algorithm design strategies. We conducted the experiment at Bolyai Farkas Highschool from Targu Mures (Romania), in the school year 2003/2004. At this school there are at the moment for each grade (IX – XII) three classes of Computer sciences functioning in parallel. The official curriculum foresees the teaching of the techniques in the Xth grade. We involved all three classes in the experiment: X.g., X.h. and X.i. Classes h and i had the same teacher, but class g a different one. We chose classes g and i as experimental classes and class h for the control group. The teachers in the experimental classes were teaching using the „upperview”-method, according to the suggested syllabus, and using the classical method in the control class (we treat the techniques as separate units). The

average marks in both experimental classes (although the methods were applied by different teachers), were approximately 1.5 marks higher (on the 1-10 scale), which represents a significant difference.

The overview method activates the wide range of thinking operations, which offers a theoretical explanation for its efficiency. Furthermore, it contributes considerably to the improvement of the general problem-solving thinking of the students. [20]

As it presents the different concepts interlinked, it results in a longer lasting mental representation. This is in harmony with the way the modern science sees the human memory. [37] At the end of the curriculum the picture of the algorithm design strategies will be comprehensive, clear and deep in the mind of the students, but also stored for a long term.

3 Involvement of the sight, hearing and the phenomenon of kinaesthesia in teaching-learning process of elementary algorithms

From the presented research results in this chapter we can emphasize at least five reasoning lines, which all lead to the same conclusion: the process of teaching-learning is more efficient the more senses are involved in it:

- more senses - more information
- different students - different dominant senses [28, 31]
- different students - different "intelligences" [29]
- multiple senses - more pathways of locating the stored information
- multiple senses - distributed loading [39]
- combined senses – more efficient learning process

Learning computer programming, just like mathematics, is a multiple abstract process. The programming language itself can be considered as a first level abstraction. The computer program will be a second abstraction level. The algorithm behind the program can be regarded as a third level abstraction. In conclusion it is obvious that computer programming, first of all, is a left brain activity. This recognition indicates another reason why the multiple senses involvement -if they activate the right side of the brain- in teaching-learning elementary algorithms is especially efficient.

For example in several New Mexico schools teaching mathematics is combined with teaching music and dance. Their experience shows that these two apparently different areas have much to offer to each other. [30]

Any algorithm has a "loop-skeleton", its structure of loops. The instructions that represent the nucleus of the loops can be seen as the algorithm's "meat-parts". [64] In what follows, we recommend two-steps method for teaching and learning elementary algorithms:

1. *Analyzing the task we establish the loop-skeleton of the algorithm that solves the problem.*
2. *We fill up the loop-skeleton with the adequate instructions.*

We have developed a new didactical software tool to help the in establishing the loop-skeleton. The software has four main modules: `cod_creator`, `cod_beautifier`, `cod_buherator`, `run_code`.

The `cod_creator` module makes it possible to create program-skeletons with different loop structure in an automatic way. It runs in two modes:

- *Giving the loop-skeleton's parameters: We introduce how many loops we want on the first, second and third level, and which is subordinate to which. Additionally we have to give the number of iterations of each loop.*
- *Drumming the loop-skeleton in: This mode makes it possible to type the program's loop-skeleton in, as if we have drummed its rhythm-pattern in. The phenomenon of kinaesthesia is involved especially at this stage of the learning process.*

By the `code_beautifier` module every C/C++ source file can automatically be reorganized ("beautified") in such a way that its cycle skeleton should easily be noticed. This operation gets an important role because of the sight involvement.

The `code_buherator` module – rewriting the source file- plants sound and delay procedures in nucleuses of each loop instruction. This will have such an effect upon the algorithm as if its loop-skeleton would have been spoken up. A piano sound will be heard every time when a loop's nucleus is traversed. The outer loops will be audible in lower, and the inner ones in higher sounds. Additionally, applying different length delays in case of the loops, which are situated on different levels, the result will be that the outer loops will have smaller frequency sound-sequences than the inner ones.

In the dialog box of the `run_code` module the "beautified C/C++ cod" of the analyzed algorithm appears. Pushing the Run button starts the slow motion running of the program. While the students "are listening to the algorithm's cycle-skeleton" represented by its sound-sequence, they can keep their eyes on the program's running (as we can see the instruction which is being executed is highlighted).

We suggest the following syllabus as application of the presented software.

- *The students "are listening to" several loop-skeletons while they keep their eyes on the slow motion running of the program generated by the `code_creator` module.*
- *The students are invited to recognize some unknown loop-skeletons, only by hearing their "piano accompaniment".*
- *The students are following, with their eyes and ears, the running of some well-known elementary algorithms in the `run_cod` module's dialog box.*
- *The students are invited to "drum in" divers loop-skeletons. As the students in New Mexico schools feel the fractions in their bones because of the music and dance, our students should get to the point where the algorithms' rhythm rings in them. They reach this stage because of using their fingers in a pianist manner to "type" repeatedly the loop-skeleton of the different algorithms. This phase of the method can be applied even if no computers are accessible. Firstly the teacher, then the students can "drum" the loop-skeletons' rhythm, using their hands and legs.*

- *The students are listening to the “piano accompaniment” of algorithms that contain typical errors. Then they are invited to compare these sound-sequences with the “piano accompaniment” of the correct algorithms.*

In order to prove empirically the efficiency of the above presented didactical method we performed an experiment. Two 9 grader classes (IX. g and IX. h) were involved in the experiment (with 24 and 26 students) from the Bolyai Farkas High School (Targu Mures, Romania). Both classes had started to learn C/C++ programming language at the beginning of the 2005/2006 school year. The experiment took place at the end of the first term. Different teachers teach the two classes but conform to the same syllabus. We divided both classes in two equivalent groups. Therefore, we had an experimental group and a control group in both classes. The division of each class was made on the basis of the students’ performance during the whole term. During the two-week experiment, in the control groups of each classes, the elementary algorithms subject was taught according to the above presented two steps method and syllabus. In the control groups, evidently, the students were taught according to the classical methods without making any effort to involve the senses in the teaching-learning process. At the end of the experiment, we made the students write a test. The following significant results have been obtained:

- The results of class *h* show that the application of the new method will probably (with a probability of 0.9804) increase the class average with 0.76 marks.
- In case of class *g* the value is 1.35, with a probability of 0.9897.
- If we analyze the two samples together, we can expect an average 1.05 improvement, with a probability of 0.9957.

The use of this software in the teaching-learning process of elementary algorithms has made it possible for students to feel the algorithm’s pulse. It has really been learning with the “whole body”. The students could see, hear and – drumming the loop-skeleton’s in – feel the pulsation of algorithms. The more senses involved doesn’t only mean more information, better perception and more efficient memorising, but ensures the same chance for students with different dominant senses.

The high-level right-brain involvement in the teaching-learning process represents a theoretical explanation for the efficiency of this didactic method. The code-based analyse of the implemented algorithm takes place mostly on the left side of the brain. Our method, by involving the mentioned senses in the learning process, activates the right side of the brain too. For example, during the slow-motion execution of the algorithms, the loop statements of the beautified code are processed as text-based information on the left side of the brain. In the same time, the tones synchronised with the highlight of the current line of code supply information about the loop-skeleton to the right side of the brain. Through this dual coding, the information is transmitted to the brain in three different forms. On one hand it is textual (verbal) and on the other hand it is auditive and visual (non-verbal by images and sensations). While our ears hear the sounds, the highlight of the current line guides our eyes over running code, so we can say that these three forms bind together in a harmonic way. All of this contributes to the effectiveness of the learning process and memorising.

We can ask the “musician students” (those, who play musical instruments) to prepare a “piece” for the next class. The colleagues will have to identify the heard algorithms. It is

not hard to imagine the positive affective effect on the students when they hear their own colleagues playing the guitar or the flute in computer-science classes.

4 “Who wants to be eminent” – Assessment method and software

In the modern concept, the assessment is not only a teaching and educational method, but also a regulatory element. Using the information obtained through assessment, the teacher's, but also the learner's activity can be regulated. It is an advantage if the feedback information is often produced, as the teaching-learning process can have a fine-tuning. [44, 45, 46]

The methods applied today in education are various, partly the traditional assessment methods are still present, partly they have broadened with the methods of scientific research and with the methodical requirements connected to them (objectivity, validity, reliability). The computers and the appearance of computer networks focused the attention on testing, as it can avoid several shortcomings of the traditional methods, but unfortunately, it creates new ones.

The ideal teaching-learning process should be practical and in the same time enjoyable. [6] The question we have analyzed is whether we can extend this basic principle of Comenius to the assessment phase as well.

Our endeavour is should not be considered a pioneer work, as several researchers and teachers have achieved good results in this field. The computer and later the appearance of the computer networks have brought new possibilities. Another help might be the media itself. Many TV-games have become popular, which beyond the game itself are meant to educate and in the same time to assess the player's level of knowledge. The success of these games has shown that it is not impossible to make the assessment exciting, as well. Of course, there is a big difference between classrooms and TV studios: the worst thing that can happen to the player is that he might not win, whereas for the student the examination might end with a bad mark or with a failure at the exam. Therefore, the educational software inspired by TV-games is more often used for the revision of the learnt part than for examination.

Such applications are for example the classroom implementations of the popular TV-game Jeopardy. [49, 50] An other successful TV-game is “Who wants to be a millionaire?”. Although many teachers have been inspired by this TV-game and have created software for testing the students, we think that the testing network software “Who wants to be eminent” created by us is from many points of view a pioneer in this field and can be considered one of the most successful implementations of the above mentioned TV-game. Many applications, though they exemplify very well the way different classroom implementations can increase the efficiency of the teaching-learning process, are far bellow the possibilities offered by computers and computer sciences. [51, 52, 57]

The “Who wants to be eminent” software is a user-friendly application, which you can access through the Internet as well. The software can be used as three different types of users: we can log in as teacher, responsive student or “spectator student”. Contrary to the TV-game, there can be several responsive students.

The software can be exploited in different operational modes. The classical testing mode has been implemented, when the students get the questions one after the other one and the software sums up the good answers at the end. In the implicit “eminent” mode the students have to answer 9 questions and if they answer all of them correctly, with the 1 point received officially means a 10 mark (in accordance with the marking system 1-10 used in Romania). There are two margin values as the 5 and the 8. The questions are grouped in three categories of difficulty. The software chooses the first four questions in a random order from the first (easiest category), the next three from the second category (up to mark 8), finally the last two questions are chosen from the third category (up to mark 10). The responsive students can any time stop the process of their assessment, if they are contented with the result they have obtained and are not sure in the answers to the following questions. In this case they are transferred among the spectators. Should they give a wrong answer, their mark falls back to the closest marginal value. Every responsive student has the possibility of the three helps: 50-50, can ask for the statistics of the spectators’ answers, can chat with one of the spectators.

The “Who wants to be eminent” software can be well used for examination, for subjects which allow the assessment based on testing. We suggest the following course of the examination:

- *Firstly we run the software – twice consecutively – in the “eminent” operational mode. At first the students from the first half of the class are responsive students and the others spectators, and secondly we invert the roles.*
- *Then the program runs in classical testing operational mode*
- *Eventually we write the higher mark out of the two the student received, into the examination register.*

By ending the examination with a traditional testing (which is generally accepted and which would have been performed anyhow) and by writing the higher mark, we ensure the fact that the student can only win from the “eminent” operational mode.

In order to be able to use this software in the classroom as well, we have developed an alloyed operational mode, too. In this case the responsive students get the first four questions in a traditional testing system. After answering the fourth question, the software displays the points reached by the student, which will be considered the first margin for the “eminent” mode starting at this moment (the next margin will automatically be set for three points higher). The students who have “fallen out” will take part in the following testing as spectators: “post eminent” testing.

By developing the “Who wants to be eminent” educational software our aim was to ensure such a method and tool for the assessment process, which can bring excitement into testing, activates the whole classroom, beyond the evaluation it teaches and educates as well, has all the advantages of testing compared to the traditional way, exploits maximally the advantages given by the computers and computer sciences and leaves behind a data base which makes extensive educational research possible. We have also paid attention on building into the software the advantages of the already existing applications, next to the original ideas. The following summing up of the method’s advantages ensure us that we have reached our goal. We hope that further experiences and research will enforce this conclusion.

- *It creates spirit of contest and offers excitement, which are further resources concerning the preparation for the test.*
- *The classroom variant of the software “models” the teacher, who pays attention to the stress at the start, makes help possible when the student has problems, is ready to take into account the knowledge revealed after the end of the test.*
- *Several students can answer in the same time. If we allow one minute for each of the first four questions, two minutes for each of the following three and two minutes and a half for each of the last two questions, we can finish the testing in 15 minutes.*
- *The software makes it possible to use the strong points of the “eminent” testing at examinations.*
- *The whole class is working, as the questions appear on the screen of each student and they have to answer. Furthermore, with the help options the whole class is part in the assessment of the students. The “eminent” mode gives a “legal” frame to the way the whole class can help the responsive students. All these mean an extra resource in the preparation for the next class.*
- *The server monitors the activity of every student for the teacher. This eliminates the possibility, that the lack of knowledge or the very high level of knowledge of the student not being tested remains unnoticed.*
- *After the end of the testing the teacher already has some printable statistical figures.*
- *As the module editing the questions can be used independently, the students can be involved in editing the questions, which has several advantages.*
- *There is an electronic minute made about every second of the testing. This database makes possible a later analysis of the teaching-learning process.*
- *Regarding the low time needed for this testing, it can be used quite often and thus a fine-tuning of the teaching-learning process becomes possible.*
- *Beyond assessment, it makes education possible, not only teaching and learning.*
 - *It activates such features in the students as selflessness, helpfulness, sympathy. The spectator students express these features by living thoroughly the testing as though they should answer, in order to do their best with their help.*
 - *Self-confidence becomes evident, respectively the confidence in the classmates and the class as a whole.*
 - *Can contribute to the strengthening of the class spirit.*
- *The strict punishment of the wrong answers, the possibility to stop, the common presence of the margins and the help resources often put the students in front of choices which test them but in the same time it develops their wisdom.*

References

- [1] R. Kotulak, *Inside the Brain*, Andrews McMeel, Kansas City, 1996
- [2] R. L. Atkinson, R. C. Atkinson, *Pszichológia*, Osiris, Budapest, 1997
- [3] P. Drucker, *Knowledge-worker Productivity: The Biggest Challenge*, California Management Review 41, S. 79–94, 1999
- [4] P. Drucker, *Beyond the Information Revolution*, The Atlantic, 1999. 12
www.theatlantic.com/issues/99oct
- [5] F. Schaffhauseri, *Társadalom-iskolaelmélet – iskolaszervezet*, Bábosik I., R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [6] Comenius, *Orbis sensualium pictus*, 1653
- [7] Z. Krygowska, *A matematikadidaktika jelenkori kutatásainak főbb irányzatai és problémái*, Dydaktyka Matematyki, Varsó, 1/1982
- [8] H. Braun, *Reflections on the future of assessment*, Konferenciaelőadás, EARLI (European Association for Research on Learning and Instruction), Assesment and evaluation, 2000. szeptember 13–15., Maastricht
- [9] A. Stephen, *Hogy kell a tanulás eredményeit használni? „Csoportkapcsolatok”*, Bolognai Konferencia, Budapest, 2006. január 25.
- [10] K. Garnitschnig, G. Khan-Svik, *Aktív tanulás. A sikeres oktatásszervezés aspektusai*, Bábosik I., R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [11] Káta Z., *Algoritmustervezés felülnézetből*, Sapientia Kiadó, Kolozsvár, 2006
- [12] B. S. Bloom, *Taxonomie von Lernzielen im kognitiven Bereich*, Weinheim, Base, 1972
- [13] B. S. Bloom, D. R. Krathwohl, *Taxonomie von Lernzielen im affektiven Bereich*, Weinheim, Base, 1975
- [14] E. Persy, *Motiváció a tanítás-tanulás folyamatában*, I. Bábosik, R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [15] E. Ch. Wittmann, *Grundfragen des Mathematikunterrichts*, Vieweg, 1981
- [16] L. Eliot, *What's Going On In There? How the Brain and Mind Develop in the First Five Years of Life*, Bantam Doubleday Dell Pub, 2000
- [17] T. H. Cormen, C. E. Leirserson, R. L. Rives, *Introduction to Algorithms*, by The Massachusetts Institute of Technology, 1990, 266-270, 287-289.
- [18] I. Odagescu, C. Copos, D. Luca, F. Furtuna, I. Smeureanu, *Metode și tehnici de programare*, Intact, Bucuresti, 1994, 95-108.
- [19] R. Andone, I Garbacea, *Algoritmi fundamentali. O perspectivă C++*, Libris, Cluj-Napoca, 1995, 185-187, 219-221.
- [20] Revákné Markóczi I., Máthé J., *A természettudományos problémamegoldó gondolkodás fejlesztése a középiskolában*, Új Pedagógiai Szemle, 2002/10
- [21] J. Dewey, *How we think*, New York, Holt, Rinehart and Winston, 1910
- [22] J. Rossman, *The psychology of the inventor*, Washington, Inventor's Publishing Co, 1931
- [23] A. Osborne, *Applied imagination*, New York, Scribner, 1963
- [24] Pólya Gy., *A problémamegoldás iskolája*, Budapest, Tankönyvkiadó, 1979
- [25] Lénárd F., *A problémamegoldó gondolkodás*, Budapest, Akadémiai Kiadó, 1963
- [26] J.Holt, *How Children Fail*, New York, Pitman, 1994.
- [27] *Research on the Brain*, Classroom Compass, 1997, Volume 3, Number 2

- [28] D. Dickinson, Washington State Art Commission, winter 2003 issue
- [29] H. Gardner, *Frames of Mind*, 1985
- [30] *Rhythm of Mathematics*, Classroom Compass, 1998, Volume 4, Number 2
- [31] M. D. Levine, *A Mind at a Time*, Simon & Schuster, 2002
- [32] H. C. Hughes, *Sensory exotica – a world beyond human experience*, Bradford Book, 2001
- [33] Bábosik I., R. Olechowski (Szerk.), Tanítás-Tanulás-Értékelés, Peter Lang Tudományok Európai Kiadója, 2003
- [34] D. Laird, *Approaches to training and development*, Addison-Wesley, Reading, Mass., 1985
- [35] *Navy Instructor Manuals*, www.tpub.com/content/administration
- [36] *Out of Memory*, <http://library.thinkquest.org/C0110291>
- [37] <http://www.cis.upenn.edu/~matuszek/cit594-2004/Lectures/44-dynamic-programming.ppt>
- [38] P. Russel, *The Brain Book*, Dutton, New York, 1979
- [39] Komenczi B., *Orbis sensualium pictus. Multimédia az iskolában*, Iskolakultúra, 1997/1
- [40] Paivio, *Mental representations. A dual coding approach*, New York, Oxford University Press, 1986
- [41] D. L. Nelson, V. S. Reed - J. R. Walling, *Pictorial superiority effect*, Journal of Experimental Psychology: Human Learning and Memory, 1976/2, 523-528
- [42] M. Paechter, *Sprechende Computer in CBT: eine didaktische Konzeption*, Arbeiten aus der Seminar für Pädagogik, Bericht 1/93. TU, Braunschweig, 1993
- [43] M. Pyter, *Textrepresentation in Hypertext. Empirische Analyse von visuellen versus audiovisuellen Sprachdarbietungen in Hypertext*, Papier zum Kongress der DGPs, Hamburg, 1994
- [44] Báthory Z., *Értékelés a pedagógiában*, Pedagógiai Szemle, 1972/3, 212-220
- [45] Golnhofer E., *A pedagógiai értékelés*, Falus Iván (szerk.), Didaktika. Nemzeti Tankönyvkiadó, Budapest. 392-414.
- [46] H. Brückner, *Számítógépek az oktatásban, Számítógépes oktatás*, KSH Nemzetközi Számítástechnikai Oktató és Tájékoztató Központ, Budapest 1978
- [47] Bálya D., *Az informatika kihívása a teszt-technológiában*, TDK dolgozat, BME TIO 1997
- [48] A. Binet, *A propos de la mesure de l' intelligence*, Paris, 1905
- [49] J. Benek-Rivera, V.E. Mathews, *Active learning with Jeopardy: Students ask the questions*, Journal of Management Education, 28 (1), 104-118, 2004
- [50] J. J. Grabowski, Michelle L. Price, *Simple HTML Templates for Creating Science Oriented Jeopardy! Games for Active Learning*, Department of Chemistry, University of Pittsburgh, 2002
- [51] D. Marsh, *Who wants to be a maths millionaire?*, <http://ferl.becta.org.uk>, 26 April 2002
- [52] J. J. Cochran, „*Who Wants To Be A Millionaire: The Classroom Edition*”, INFORMS Conference, San Antonio, TX., 2000
- [53] G. Brown, M. Atkins, *Effective Teaching In Higher Education*, Methuen, London, 1988
- [54] J. Hartley, I. K. Davies, *Note Taking: A Critical Review*, Programmed Learning and Educational Technology, Vol. 15, 207-224, 1978
- [55] W. J. McKeachie, *McKeachie's Teaching Tips: Strategies, Research and Theory for College and University Teachers*, Houghton-Mifflin, Boston, MA, 1999

- [56] T. Sutherland, C. Bonwell, *Using Active Learning In College Classes: A Range Of Options For Faculty*, New Directions for Teaching and Learning, Vol. 67, 1996
- [57] J. M. Collins, *Who wants to be a millionaire? An educational game for learning enhancement*, Innovative Teaching Practices at Bloomsburg University, Teaching and Learning Enhancement Center, 2004
- [59] Kátai Z., *Dynamic programming and d-graphs*, Studia - Informatics, Cluj, 2006
- [60] T. Angelo, P. Cross, *Classroom assessment techniques: A handbook for college teachers* (2nd edition), San Francisco, 1993
- [61] Kátai Z., *Dynamic programming strategies on the decision tree hidden behind the optimizing problems*, Informatics in Education, Institute of Mathematics and Informatics, Lithuania, 2006 (elfogadva)
- [62] Kátai Z., *Involvement of the sight, hearing, touching and moving in teaching basic algorithms*, Computers & Education, 2006 (*benyújtva*) (Társ szerzők: Nyakóné Juhász Katalin, Adorjáni Alpár Károly)
- [63] Kátai Z., "Upperview" algorithm design in teaching computer science in high schools, Teaching Mathematics and Computer Science, 3 (2005) 2
- [64] Kátai Z., *Programozás C nyelven*, Sapientia-EMTE, Scientia Kiadó, Kolozsvár, 2004

Publications

Papers published in international journals:

- [1] "Upperview" algorithm design in teaching computer science in high schools, Teaching Mathematics and Computer Science, 3 (2005) 2, (221-241)
- [2] Dynamic programming and d-graphs, Studia Universitatis Babes-Bolyai -- Series Informatica, 2006 (accepted)
- [3] Dynamic programming strategies on the decision tree hidden behind the optimizing problems, Informatics in Education, Institute of Mathematics and Informatics, Lithuania, 2006 (accepted)
- [4] Involvement of the sight, hearing, touching and moving in teaching basic algorithms, Computers & Education, 2006 (submitted) (Co-authors: Nyakóné Juhász Katalin, Adorjáni Alpár Károly)

Lecture notes:

- [1] Programozás C nyelven, Sapientia-EMTE, Scientia, Kolozsvár, 2004
- [2] Algoritmustervezés felülnézetből, Sapientia-EMTE, Scientia, Kolozsvár, 2006

Conference presentations:

- [1] Hogyan tanítsuk a programozási technikákat, XIVth Computer Science in Education International Conference, EMT, Kolozsvár, March 25-28 2004
- [2] Programozási technikák felülnézetből, XVth Computer Science in Education International Conference, EMT, Kolozsvár, March 17-20 2005
- [3] Algoritmus tervezés (Didaktikai szempontok), Informatics in Education International Conference, Debrecen, August 24-26 2005

Other publications:

Conference publications:

- [1] Hogyan tanítsuk a programozási technikákat?, XIVth Computer Science in Education International Conference, EMT, Kolozsvár, 2004, (50-56)
- [2] Programozási technikák felülnézetből, XVth Computer Science in Education International Conference, EMT, Kolozsvár, 2005, (139-146)

Other articles published in scientific journals:

- [1] Programozási technikák felülnézetből (I rész), Firka, 2003/2004-4, EMT, Kolozsvár (145-148)
- [2] Programozási technikák felülnézetből (II rész), Firka, 2003/2004-5, EMT, Kolozsvár, (190-192)

Software products:

- [1] "Legyél te is eminens", Assessment software, (Co-author: Máthé Szabolcs)
- [2] "Elemi algoritmusok anatómiája", Educational software, (Co-author: Adorjáni Alpár Károly)