

Életkor független programozási készség tesztelő rendszer

Age independent programming skills testing system

Szilágyi E. D.^a, Szikra Á.^a, Kiss G.^b, Stóka Gy.^c, Godó Z. A.^a,

^a Debreceni Egyetem, Informatikai Kar
godo.zoltan@inf.unideb.hu

^b Óbudai Egyetem

^c Eszterházy Károly Egyetem, Sárospataki Comenius Campus

Absztrakt: Az informatika felsőoktatásában tapasztalható a hallgatók heterogén készsége a programozás elsajátításában. A jelenség szélsőséges differenciához vezet, jóllehet a felsőoktatásba a terület iránt elkötelezett és feltehetően a középfokú informatika oktatásban már jó eredményeket felmutató diákok felvételére számítunk. A jelenség nem feltétlenül a tanulási affinitásban keresendő. A készséget a képesség előzi meg, amelyben genetikai, majd korai szocializációs folyamatok is szerepet játszanak. A korábban általunk kifejlesztett fényprogramozási eljárás, mikrokontroller alapú hardver és szoftverrendszer, alkalmasnak bizonyult a teljes életkor intervallumban az algoritmus alkotó-képesség tesztelésére. Vizsgálataink abba az irányba mutatnak, hogy a hosszabb iskolázottság során elsajátított programozási tudás nincs olyan szoros összefüggésben az algoritmus-alkotó képességgel, mint ahogyan azt feltételeztük. Habár a tanulási folyamat szignifikánsan növeli az algoritmizálási képesség fejlődését, e-mellett markánsan fellépnek olyan tényezők, amelyeket már kisgyermek korban is differenciáltan tudunk mérni a rendszerünkkel. Ezek a képességek jelentősen befolyásolják az oktatás sikerességét, így mélyebb vizsgálatuk mindenképp indokolt. Az algoritmus-alkotó képesség kialakulásának jobb megértése közelebb vihet minket a programozás oktatás hatékonyságának javításához az alapfokútól a felsőoktatásig.

Kulcsszavak: algoritmusalkotó-képesség, fényprogramozás, oktatás

Abstract: The skill-levels of Information Technology students at a higher education level needed for learning programming tend to be heterogeneous. Although, the students that are expected to gain admission are those committed to their field of study and supposedly with good results at the middle-level education of Information Technology, the phenomenon leads to extreme differences. The phenomenon is not necessarily rooted in their affinity for learning. Skill is preceded by ability, which is affected by genetics and early socialization. Developed by us earlier, light programming, a micro controller based hardware and software system, has proven suitable for testing algorithm designing ability at any age. Research results point to the conclusion that the programming skills learned during a longer education and the ability to design algorithms are not so closely related as was previously assumed. Although education significantly helps develop the algorithm designing ability, such factors that we can measure differentially with our system in early childhood already also have a major role. These abilities affect the success of education considerably, calling for a deeper examination. An understanding of the development of the algorithm designing ability might bring us closer to improving the efficiency of teaching programming from the basic level to higher education.

Keywords: algorithm designing ability, light programming, education

1. Bevezetés

Az informatika felsőoktatásában tapasztalható a hallgatók heterogén készsége a programozás elsajátításához. A jelenség szélsőséges differenciához vezet, jóllehet a felsőoktatásba a terület iránt elkötelezett és feltehetően a középfokú informatika oktatásban már jó eredményeket felmutató diákok felvételére számítunk [6].

A közoktatásban egyre nagyobb hangsúlyt kap a tanulók kognitív képességeinek fejlesztése. A mindennapi életben szükség van a tudatos gondolkodásra, mellyel a napi problémákra tudunk gyors és hatékony megoldásokat találni. [2,5] Így az algoritmusok (tudatos gondolatsorok) alkotásának tanítása, illetve gyakoroltatása fontos didaktikai cél. [1] Indokolt már kisgyermek korban elkezdni a tudatos gondolkodásra való nevelést, hiszen a későbbi életkorokban egyre jobban hasznosítható ez a képesség és elősegíti a hatékony tanulási technikák elsajátítását. [3] Az algoritmus-alkotó készség méréséhez hiánypótló volt az általunk fejlesztett speciális eszköz, melynek segítségével nagy pontossággal tudjuk eltárolni és értékelni a differens tesztalanyok eredményeit. Fontos az eszköz univerzialitása, egyszerűsége is, így egészen óvodás kortól kezdve tudjuk nyomon követni a tesztalanyok fejlődését. A fejlesztésünk ezért a vizualizációs programozás irányult, ahol a kisgyermek kortól kezdve, azonos feltételekkel tudjuk vizsgálni a tesztalanyokat a felnőttkorig [8].

Munkánk célja, a különböző életkorokban, tanulási fázisokban jelen levő algoritmizáló képességek mérése, illetve a korosztályok közötti különbségek feltárása, a tanulási- szociális- és genetikai összetevők differenciálása.

2. Képesség a készség előtt

A készséget a képesség előzi meg, amelyben genetikai, majd korai szocializációs folyamatok is szerepet játszanak [7]. A korábban általunk kifejlesztett fényprogramozási eljárás, mikrokontroller alapú hardver és szoftverrendszer, alkalmasnak bizonyult a teljes életkor-intervallumban az algoritmus alkotó-képesség tesztelésére.

Vizsgálataink abba az irányba mutatnak, hogy a hosszabb iskolázottság során elsajátított programozási tudás nincs olyan szoros összefüggésben az algoritmus-alkotó képességgel, mint ahogyan azt feltételeztük. Habár a tanulási folyamat szignifikánsan növeli az algoritmizálási képesség fejlődését, e-mellett markánsan fellépnek olyan tényezők, amelyeket már kisgyermek korban is differenciáltan tudunk mérni a rendszerünkkel. Az iskolában elsajátított tudás esetén nagy különbség van a megértés és a tudatos használat között. Míg az adott tananyag megértéséhez elég „bemagolni”, memorizálni a konkrét leírásokat, fogalmakat, addig az ismeretek alkalmazása igényli a fogalmi megértést. A programozás tanítása során is minél inkább arra kell törekedni, hogy a hallgatók a különböző feladatokat a saját maguk módszere szerint oldják meg. [4] Így folyamatosan fejleszthetik a kreatív attitűdjeiket.

Ezek a képességek jelentősen befolyásolják az oktatás sikerességét, így mélyebb vizsgálatuk mindenképp indokolt. Az algoritmus-alkotó képesség kialakulásának jobb megértése közelebb

vihet minket a programozás oktatás hatékonyságának javításához az alapkötől a felsőoktatásig.

3. Hardver & szoftver

A rendszer egyetlen felületből áll, melyet a tesztelt és a mérést végző személy is kezel. A műszer alsó felén 12 LED és alatta 12 nyomógomb található. Az egyes gombokkal ki- és be tudjuk kapcsolni a hozzájuk tartozó LED-eket, melyet korlátlan számban tehetünk meg.



1.ábra: A kész mérőeszköz

A LED sor felett jobb szélén két nyomógomb található, a felsővel a tesztelő el tudja indítani a mérést, az alsóval pedig a vizsgált személy a memóriába tudja helyezni az általa bevitt adatokat. A két nyomógomb mellett egy LCD kijelző helyezkedik el, mely a beadott feladatot kiértékeli és összehasonlítja az eredeti mintával, ezután kiírja a hibák számát és a feladat megoldásának idejét, századmásodperc pontossággal.



2.ábra: A teszter belülről

A készülék a következő hardware elemekből tevődik össze:

- mikrokontroller,
- LCD kijelző (2x16),
- 12db LED
- 14db nyomógomb.
- Illetve egy potméter, mellyel az LCD kijelző kontrasztját lehet beállítani.

A mikrokontroller egy Arduino mega 2560-as mikrovezérlő, melyek főbb paraméterei:

- 54 db I/O (ki és bemenet)
- 16 Analóg bemenet
- 16 MHz-es processzor sebesség
- Működési tápfeszültség USB portról 5V, külső tápegységről: 7V-12V

A mikrovezérlő választásának oka, hogy relatív sok digitális ki és bemenettel rendelkezik, valamint más egyéb eszköz (például demultiplexer) alkalmazása/bevonása nélkül rá tudtuk csatlakoztatni perifériákat. Az Arduino mellett szól az elterjedtsége, és a "könnyű" programozhatósága, mely a C nyelvhez hasonlít leginkább, kiegészítve a mikrokontrollerekre jellemző speciális utasításokkal.

A nyomógombok "pozitív logika" szerint működnek, azaz megnyomásukkor logikai 1 kerül a mikrokontroller bemenetére. A nyomógombok pergesmentesítését szoftveresen korrigáltuk. A teszter mellett, hogy az LCD kijelzőn megjeleníti az információkat, a soros portján keresztül is elküldi az adatokat. Így lehetőség van egy másik eszközzel pl. PC, mikrokontroller vagy más,

hozzá fejlesztet intelligens eszközzel fogadni, és feldolgozni az információkat. Egyúttal az USB portot felhasználva külső tápforrás is biztosítható számára.

4. Működés

1. Statikus minta másolása

A rendszer bemutat egy statikus mintát, majd kis idő múlva a LED-ek kialszanak. Ezután megkezdődik a mérés. A tesztalany a gombok megnyomásával tetszőlegesen ki/be kapcsolja a fényforrásokat, feladatul kapva, hogy a látott mintával megegyezőt hozzon létre. Ha elkészült beviszi az állapotot a rendszerbe. A jobb felső gombbal lefuttatja az elkészült programot. A minta a későbbi lépésekben egyre jobban nehezíthető, bonyolultabb ábrák is kirajzolhatók, vagy csökkenthető a bemutatásra szánt idő.

A mikrokontroller által bemutatott egyszerű minta:



Ha a tesztalany helyesen követi a rendszer által bemutatott állapotot, akkor az sikeresnek értékelhető. Mérhető a megoldásban levő hibák száma, a megoldás „beadásáig” eltelt idő, a megoldás során történt hibás próbálkozások száma, illetve a megoldás eléréséig lenyomott billentyűk száma. Minden adat objektíven, grafikonon ábrázolható és összehasonlítható.

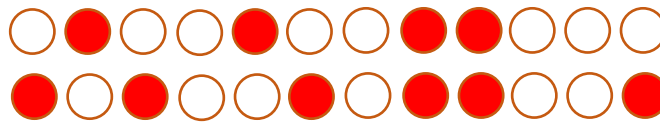
Példa a helytelen megoldásra:



Egy, a rendszer által bemutatható bonyolultabb minta:



Jellemző hibás válaszok:



Példa a PC-n feldolgozott első feladat megoldására, melyet felhasználók szerinti lebontásban láthatunk:

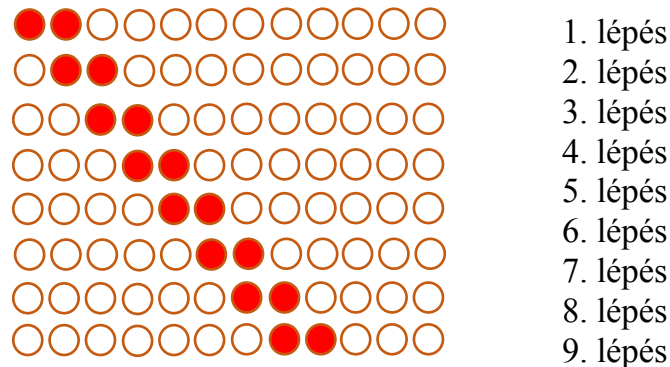
Programozni tudó tesztalany	Hibák száma	Lenyomott gombok száma	A helyes megoldáshoz szükséges gombok száma	Idő (mp)
1.	2	5	4	3,40
2.	1	3		4,10

3.	4	6	2,40
4.	2	4	2,40
5.	0	4	4,77
6.	1	3	5,22
7.	5	7	4,23
8.	0	4	3,65
9.	1	5	1,56
10.	0	4	6,45

2. Dinamikus minta követés

A mérési eljárás során egy-vagy több LED fény felvillan, ami a következő lépésben megismétlődik, valamilyen helyváltozással. A lépések egymás utáni futtatása adja a fényjáték programját. A tesztalany feladata a fények mozgásának lépésekre bontása, vagyis az algoritmus megalkotása. A tesztalany megadja a számára helyesnek vélt minta első lépését, majd azt az LCD kijelző mellett található alsó nyomógombbal eltárolja a mikrokontroller memóriájában. Ezután sorban elvégzi ugyanezt a minta további részével. A feladat befejezése esetén megnyomja a felső gombot, mellyel vissza tudja játszani a saját megoldását.

Példa a dinamikus mintára:



Kiértékelése hasonlóan történik, mint a statikus minta másolásánál. Itt azonban jóval több adattal rendelkezünk, a feladat jelentősen nagyobb összetettsége miatt. A 12 LED lehetőséget biztosít az egészen egyszerű minták létrehozásától kezdve a szinte követhetetlenül bonyolult és egyben igen látványos feladatokig. A legegyszerűbbtől a legnehezebb feladatokig nagyon kis felbontással bonyolíthatóak a megoldások. A feladatok nehézségét a felhasznált fényforrások és a lépések száma alapján pontosan tudjuk számszerűsíteni is. Mindezek igen jól feldolgozható, differenciált adatsorokat nyújtanak a kiértékelés során.

5. Eredmények kiértékelése

A tesztalany által bevitt adatok a mikrokontroller memóriájában tárolódnak. Az adatok könnyen transzportálhatóak PC-re, ahol tetszőleges programok segítségével kiértékelhetőek. A teszter jelentős számú mérést képes eltárolni, a hibák és az összesen lenyomott gombok számán kívül, századmásodperc pontossággal elmenti a megoldásra felhasznált időt is. Így a rendszer könnyen mozgatható, PC és egyéb kiegészítők felhasználása nélkül azonnal használható például óvodai vagy iskolai környezetben egyaránt.

A teszter kísérleti alkalmazása során alkalmasnak bizonyult csaknem a teljes életkor intervallumban az algoritmus alkotó-képesség vizsgálatára, mintegy 3 éves kortól. Eredményeink abba az irányba mutatnak, hogy a hosszabb iskolázottság során elsajátított programozási tudás nincs olyan szoros összefüggésben az algoritmus-alkotó képességgel, mint ahogyan azt feltételeztük. Jelentős az „adottság”, feltételezhetően genetikai háttér befolyása. Habár a tanulási folyamat szignifikánsan növeli az algoritmizálási képesség fejlődését, e-mellett markánsan fellépnek olyan tényezők, amelyeket már kisgyermek korban is differenciáltan tudunk mérni a rendszerünkkel. Óvodás vagy kisiskolás gyermekek gyakran meglepően, néha egyes programozó szakos felsőoktatási hallgatóknál is jobb eredményeket értek el. Ezen gyermekek fejlődésének hosszútávon történő nyomon követése, választ adhatna, hogy milyen mértékben segítik a gyermekkori képességek az esetleges sikeres programozóvá válásukat.

Az adatok összehasonlításra kerülnek ugyan azon alanyokon elvégzett pszichológiai tesztekkel. Kisgyermekeknél az iskolaérettségi, motilitás, agresszivitás, kreativitás tesztek létjogosultságát vizsgáljuk, idősebb korosztálynál a különböző intelligencia tesztekét. Ezek eredményeinek korrelációját kívánjuk összevetni a saját tesztrendszerünk méréseivel.

Ezek a képességek jelentősen befolyásolják az oktatás sikerességét, így mélyebb vizsgálatuk mindenképp indokolt. Az algoritmus-alkotó képesség kialakulásának jobb megértése közelebb vihet minket a programozás oktatás hatékonyságának javításához az alapfokútól a felsőoktatásig.

Irodalomjegyzék

- [1] Cláudio Amorim: Beyond Algorithmic Thinking: An Old New Challenge for Science Education. Eighth International History, Philosophy, Sociology & Science Teaching Conference, July 15 to July 18, 2005, University of Leeds, England.

- [2] Csapó Benő: A kognitív képességek szerepe a tudás szervezésében
- [3] Eilers, Merry L.: Older adults and computer education: "Not to have the world a closed door.", *International Journal of Technology & Aging*, Vol 2(1), (1989), 56-76.
- [4] Geda Gábor, Hernyák Zoltán: *Algoritmizálás és adatmodellek*, Educatio Kht., (2011)
- [5] Molnár Gyöngyvér és Csapó Benő: A képességek fejlődésének logisztikus modellje. *Iskolakultúra*, No. 2. (2003)
- [6] Pap-Szigeti Róbert: A programozási készségek vizsgálata középiskolában és a felsőoktatásban. *GRADUS*, 4:(1) 216-222. (2017)
- [7] Toni Clarke: *Brain fitness seen as hot industry of the future* (2008)
- [8] Törley Gábor: *Algoritmus vizualizáció a tanítási gyakorlatban*. (2013)