Informatikai kompetenciák: Algoritmikus gondolkodás

Zsakó László, Szlávi Péter

{zsako, szlavi}@ludens.elte.hu
ELTE IK

Absztrakt. Nagyon sokan és sokat beszélnek arról, hogy mit tanítsunk informatikából a közoktatásban. Erről komoly viták is folynak. Sokkal kevesebb szó esik arról, hogy miért is kell tanulni informatikát. [1] Az ember által elvégezhető tevékenységekhez, megoldható feladatokhoz kötjük a kompetenciákat (valaki kompetens valamilyen tevékenységgel összefüggésben, ha képes megoldani az ahhoz a tevékenységhez tartozó szokásos feladatokat). Az informatika tantervekben e kompetenciák, továbbá fejlesztési feladatok, tevékenységek, képességek és készségek kapnak szerepet, mindig feltételezve mögöttük egy tudásrendszert is. Az egyik nagyon fontos és a dinamikusan fejlődő országok közoktatásában egyre nagyobb teret nyerő kompetencia az *algoritmikus gondolkodás*. Előadásunkban ennek mélységeivel foglalkozunk.

Bevezető gondolatok

A Nemzeti alaptantervben megjelenő kulcskompetenciák alapját a Recommendation of the European Parlament and of the Council of 18 December 2006 on Key Competences for Lifelong Learning (2006/962/EC) című dokumentum képezi. [2]

Az Európai Unió kialakította azt a nyolc elemből álló kulcskompetencia keretrendszert, amely a gazdaság világában és a modern társadalomban történő boldoguláshoz, a tudás megszerzéséhez és megújításához, az élethosszig tartó tanulás paradigmájához, a műveltség igényének kialakulásához, valamint a személyes önmegvalósításhoz szükséges kompetenciákat tartalmazza (kommunikáció anyanyelven, kommunikáció idegen nyelven, matematikai és természettudományos, digitális, tanulni tanulási, szociális, vállalkozói kompetencia, kulturális kifejezőkészség).

A magyar Nemzeti alaptanterv ezt kissé átfogalmazva a következő 9 kulcskompetenciát definiálja: [3]

1. Anyanyelvi kommunikáció
2. Idegen nyelvi kommunikáció
3. Matematikai kompetencia
4. Természettudományos kompetencia
5. Digitális kompetencia
6. A hatékony, önálló tanulás
7. Szociális és állampolgári kompetencia
8. Kezdeményezőképesség és vállalkozói kompetencia
9. Esztétikai-művészeti tudatosság és kifejezőképesség

A kulcskompetenciákra és a fejlesztési területekre (tantárgyakra) alapozva megfogalmazhatók egyes tantárgyi, műveltségterületi kompetenciák.

Az informatikai kompetenciák birtokában az egyén rendelkezik azzal a képességgel, hogy alkalmazni tudja az alapvető informatikai eszközöket és módszereket az ismeretszerzésben és a problémák megoldásában, a mindennapokban, otthon és a munkahelyen. Gyakorlatias módon tudja a tudását alkalmazni új technológiák, módszerek megismerésében és működtetésében, a problémamegoldásban, egyéni és közösségi célok elérésében, valamint az információs társadalom lehetőségeinek ismeretét igénylő döntések meghozatalában. [3]

Az informatikai kompetencia emiatt magában foglalja a főbb számítógépes alkalmazásokat – szövegszerkesztés, adattáblázatok, adatbázisok, információtárolás-kezelés, az internet által kínált lehetőségek és az elektronikus média útján történő kommunikáció – a szabadidő, az információmegosztás, az együttműködő hálózatépítés, a tanulás és a kutatás terén. Az egyénnek ismernie kell az elérhető információ hitelessége és megbízhatósága körüli problémákat, valamint az informatikai eszközök interaktív használatához kapcsolódó etikai elveket.

A szükséges képességek felölelik az információ megkeresését, összegyűjtését és feldolgozását, a kritikus alkalmazást, a valós és a virtuális kapcsolatok megkülönbözetését. Idetartozik a komplex információ előállítása, bemutatása és megértését elősegítő eszközök használata, valamint az internet alapú szolgáltatások elérése, a velük való kutatás, az informatikai módszerek alkalmazása a kritikai gondolkodás, a kreativitás és az innováció területén.

Sok kompetencia részben fedi egymást, és egymásba fonódik: az egyikhez szükséges elemek támogatják a másik terület kompetenciáit.

Az informatikai kompetenciák komponensei

Mivel a Nemzeti Alaptantervből indultunk ki, az informatikai kompetenciákat is a közoktatás résztvevői számára fogalmazzuk meg. Természetesen az alábbi kompetenciákat tovább lehet bővíteni az informatikai szakemberképzés szempontjai alapján – ez azonban már egy másik cikk témája lehetne.

Algoritmikus gondolkodás

A mindennapi életben, tanulásban, munkában lépten-nyomon algoritmusokat hajtunk végre, algoritmusokat készítünk mások számára, tevékenységsorozatokat, információ-áramlási folyamatokat tervezünk, s ezt a világot csak az értheti igazán, aki tisztában van ezen tevékenységek alapjaival. [4]

Adatmodellezés

A hétköznapokban rendszeresen töltünk ki nyomtatványokat, űrlapokat, készítünk mások számára ilyeneket. Ezek készítésekor mindig valamilyen információszerzés vagy átadás a célunk, melyet adathalmazokkal, adatstruktúrákkal kell alátámasztanunk. Emiatt is alapvető fontosságú a világ objektumainak adatokkal való leírása: az adatmodellezés.

A valós világ modellezése

A valós jelenségeket sokszor a modelljeiken keresztül ismerjük meg. Ehhez tisztában kell lenni a modellezés alapfogalmaival, tevékenységeivel, a modellek felhasználásának módszereivel. A megismerésen túl tudatosan használni kell a modelleket valós jelenségek előrejelzésére is! A modellezés informatikán belüli különlegessége, hogy a modellek működtetése is komplex alkotó folyamat.

Problémamegoldás

Az alkotó emberi tevékenység nagyon sok esetben problémamegoldás, a probléma minél pontosabb megfogalmazásától, a megoldás megtervezésén és elkészítésén át, a megoldás kritikus értékeléséig. Emiatt szükség van a probléma analizálására; annak eldöntésére, hogy szükség van-e a megoldáshoz informatikai eszközre; mely informatikai eszköz vagy eszközök használhatók; hogyan használhatók; ha nincs ilyen eszköz, akkor pedig hogyan készíthetünk ilyet. [5]

Kommunikációs képesség

Napjainkra az ember-ember, illetve az ember-csoport kommunikáció alapvetően megváltozott, a kommunikáló felek közé intelligens eszközök épültek be; illetve ezen intelligens eszközök újfajta kommunikációs lehetőségeket teremtettek vagy a régieket egyszerűsítik. Ezt az újfajta kommunikációt alkotó és a jogokat körültekintően figyelembe vevő módon kell használni a mindennapokban: a tanulásban, a munkavégzésben, a kapcsolatteremtésben, a kikapcsolódásban, az önképzésben, a pihenésben, a fejlődésben. [6]

Alkalmazói képesség

A mindennapokban felmerülő problémák gyakran könnyebben, gyorsabban és pontosabban megoldhatók informatikai eszközökkel, mint a hagyományosakkal. Ehhez tisztában kell lenni az alapvető általános alkalmazásokkal: funkcióikkal, az azokhoz tartozó informatikai eszközökkel és módszerekkel.

Csoportmunka, együttműködő-képesség

Az informatika lehetőséget teremt olyan feladatok megoldására is, melyeket nem egyetlen személy old meg, a feladatmegoldásban mások eredményeinek felhasználási képességétől kezdve a projektmunkákban való részvételen át, egészen a projektek tervezéséig és megvalósításáig. Mindehhez szükség van a csoportmunkát támogató informatikai eszközök használatának képességére, valamint a csoportmunka metodikájának ismeretére.

Alkotó képesség

Az informatikai feladatmegoldás nagyon sokszor alkotómunka, ahol az alkotásban informatikai eszközöket használunk. Mint minden alkotómunka, ez is elemi alkotások megismétlésével kezdődik, ezen alkotásokból újabbak készítésével folytatódik, majd adott igények alapján önálló

alkotások készítésével zárul. Az alkotókészség fejlődése tehát az egyszerű „utánzástól” / mintakövetéstől a mintavariáláson át a valódi kreativitásig tart.

Információs tájékozódási és tájékoztatási képesség

Az információs társadalom egyik lényege az információk hozzáférési jogának biztosítása. A hatalmas információhalmazban azonban nehéz a számunkra szükséges információ megtalálása, illetve a mások számára hasznos információ olyan elrendezése, elhelyezése, hogy azt könnyen találhassák meg és hatékonyan használhassák fel. Külön kiemelendő ezen képességen belül az információ hitelességéről való meggyőződés képessége; és természetesen az elhelyezett információ hitelességének „bizonyítása” például a releváns forrásokra hivatkozás által.

Rendszerszintű gondolkodás

Számtalan rendszerrel találkozunk: ilyen az egész informatikai rendszer, ami körülvesz bennünket: a különböző hálózati rendszerek, kommunikációs rendszerek. Azt szoktuk meg, hogy logikusan gondolkodunk, és dolgokat elemzéssel értsük meg, ez sokszor sikerre is vezet, de nem mindig. A rendszerrel szemben másfajta gondolkodásmódot is igénybe kell venni. Az egyszerűbb klasszikus logika sokszor tehetetlen a rendszerrel szemben. A rendszer „önmagától” működik, a részek elemezhetetlenül összetett kölcsönhatása révén. A jelenségek kimenetelét sokszor a rendszer struktúrája, nem pedig az egyedi összetevők állapotai határozzák meg. Az eseményeket irányító összefüggések felismerésével képesek leszünk befolyásolni életünk, munkánk alakulását. Ezek vizsgálata, kezelése, fejlesztése és előállítása, másfajta tudást, vizsgálatot igényel.

Az algoritmikus gondolkodás kompetencia részletezése és szintjei

Az algoritmizálás először nem számítógépes megvalósításról szól. Csak egy klasszikus, több ezer éves algoritmusra, a két egész szám legnagyobb közös osztóját meghatározó Euklideszi algoritmusra kell gondolnunk! Az algoritmus végrehajtója – a processzor – sok esetben lehet maga az algoritmust megalkotó, azt értelmező ember. (Sőt egy új probléma megoldásánál a rutinos programozó is magát „képzeli” a számítógép helyébe, s így próbálja ki a megoldás működőképességét.) Algoritmusokat mindenki hajt végre nap, mint nap, sőt az emberek többsége alkot is algoritmusokat saját maga és mások számára.

Csak ezután következhet az, hogy a precízen megfogalmazott algoritmus végrehajtását egy automatára, a számítógépre bízzuk. Ez alapvetően azért igényel más gondolkodást, mert egy intelligens algoritmus végrehajtó az algoritmusainkat is ésszerűen hajtja végre, kontrollálja, időnként kijavítja az általunk elkövetett hibákat. Egy automata azonban nem gondolkodik, ami a programjában szerepel, azt végre is hajtja (ha tudja).

Megállapíthatjuk, hogy az emberek többsége alkot algoritmusokat mások (és saját maguk) számára, algoritmust végrehajtani azonban kivétel nélkül mindenkinek gyakori feladat. [4]

1. Algoritmus (tevékenységsorozat) felismerése, megértése

Az algoritmikus gondolkodás legelemibb szintje az, amikor felismerünk algoritmusokat, algoritmussal megoldható problémákat.

Megfogalmazhatjuk, mit nevezünk algoritmusnak: lépésekre bontható; végrehajtható (tartozik hozzá végrehajtó), rögzített végrehajtási sorrenddel rendelkezik; a lépések során valamivel valami történik; a lépések vagy elemi tevékenységek (amit a végrehajtó már megért) vagy maguk is algoritmusok (további pontosítást feltételez).

A megértést két szintre bonthatjuk:

- megértjük, hogy mit kell tenni,
- megértjük, hogy miért azt (és nem mást) kell tenni.

A „miért" kérdése már egy magasabb gondolkodási szinthez, az algoritmus elemzéséhez tartozik. Az elemzés az alternatívák felismerését, elvetését, variálását és szelektálását jelenti.

Ugyancsak különböző szintű tudás egy adott, ismert algoritmus megértése (mert pl. elmagyarázták), illetve új, eddig ismeretlen algoritmusok megértésének képessége. Ez utóbbi az algoritmikus gondolkodás harmadik szintjét alkotja.

A hétköznapokban számos rutinszerű tevékenységet végzünk. Ezek némelyikét környezetünk „programozza" belénk (felkelés, lefekvés), másokat magunk hozzuk létre hasonlók módosításával többnyire valamely megváltozott –gyakran– „szükséghelyzetben" (utazás iskolába).

2. Algoritmus (tevékenységsorozat) végrehajtása

Ha egy tőlünk független személytől (különösen függőségi viszonyban) lépésenként kapnánk az utasításokat, akkor minimális mérlegeléssel hajthatnánk végre az algoritmusokat, ha azonban magunknak kell ezt megtenni, akkor ennél többről van szó.

Az algoritmusok végrehajtási képessége egy fokozattal magasabb szintű, mint a megértési képesség. Itt nemcsak arra van szükség, hogy egy folyamatot megértsünk, hanem arra is, hogy a végrehajtása közbeni állapotokat folyamatosan kövessük (észben tartsuk, nyilvántartsuk, …). Azaz nem arra kell figyelnünk, hogy mi történik, hanem arra, hogy mitől függően mit csinálunk.

E kompetencia birtokában képesek vagyunk érzékelni a végrehajtás szempontjából fontos állapothatározókat, állapot változásokat, és igazítani tudjuk a tevékenység egyes lépéseit az állapothoz.

3. Algoritmus (tevékenységsorozat) elemzése

Az algoritmusok elemzése részben arról szól, hogy felismerjük az algoritmusok alapvető felépítési szabályait:

- az elemi lépések mindegyikét végre kell hajtani (az adott sorrendben),
- az elemi lépések közül választani kell egyet, s azt végrehajtani,
- az elemi lépést többször, ismétlődően kell végrehajtani.

Mivel az algoritmus egyes lépései maguk is lehetnek újabb algoritmusok, s ezek elnevezhetők (sőt a hivatkozás miatt elnevezendők), kialakulhat az eljárás absztrakt fogalma.

Az algoritmusok elemzése más szempontból azt jelenti, hogy megértjük, hogy az egyes részei miért vannak, mi volt a célunk vele, a teljes feladatmegoldást hogyan bontottuk részfeladatok megoldására, … [7,8]

A fentieken túl algoritmusolvasási képességet is idesorolandó. Azaz valamely algoritmusleíró nyelven, más által megfogalmazott összetett tevékenységet képesek legyünk megérteni: az egyes részek célját, többiekkel való kapcsolatát lássuk, képesek legyünk elmagyarázni (a verbalizálás nem biztos, hogy azonos a megértéssel; sőt alighanem egy újabb, magasabb fokozat).

4. Algoritmus (tevékenységsorozat) alkotása

Aki képes algoritmusok megértésére, végrehajtására, az még nem biztos, hogy alkotni is tud újabb algoritmusokat. A többlet: itt először el kell képzelni, hogy

- mit ismerünk;
- mire vagyunk kíváncsiak;
- mi fog történni;
- milyen adatokkal kell dolgozni;
- a tennivalóinkat hogyan tudjuk résztevékenységekre bontani;

Ez egyrészt az olvasási tevékenység írási tevékenységre cserélését jelenti, de ennél sokkal többről van szó. Az alacsonyabb kompetencia-szinteknél voltak támpontjaink, amelyek a gondolkodásunkat segítették, itt azonban mindent magunknak kell kitalálnunk. Ez olajozottan csak úgy mehet, ha elsajátítunk valamilyen szisztematikus módszert algoritmusok (és hozzájuk kapcsolódó adatmodellek) alkotására. [9] Arra kell módszert találni, hogy az algoritmusok végtelen halmazát leszűkítsük kezelhetően kévés számúra. Természetesen ez nem megy, csak a feladatok nagyfokú korlátozásával, illetve csak akkor, ha megengedjük, hogy a kevés számú algoritmus helyett kevés számú algoritmus sémát adjunk meg. Az alkotás folyamata ez esetben a megfelelő algoritmus sémák kiválasztását, kombinálását és a konkrét tevékenységhez adaptálását jelenti. A sémarendszer felállítása analogikus és absztrakt gondolkodást igényel. A lényegi és lényegtelen jellemzők megkülönböztetése: absztrahálás; a hasonlóságok felismerése: analógia felismerés.

5. Algoritmus (tevékenységsorozat) megvalósítása

Ez már kifejezetten informatikai feladatról szól: az algoritmust le kell írni egy olyan eszközzel, amit egy automata (pl. számítógép) végre tud hajtani. Ez nyilván azzal is jár, hogy meg kell ismerni azt az eszközt (praktikusan programozási nyelvet), amellyel az algoritmust leírjuk.

Az egyes eszközökhöz is tartozik egy sajátos, rá jellemző gondolatvilág, amelyek megismerésére és átlátására szükségünk van:

- hogyan képzeljük el benne a programok végrehajtását
- hogyan épülnek fel benne a programok.

Másrészről – mivel ritka lehet az az eset, hogy elsőre tökéleteset alkotunk – szükségünk lehet a megvalósított algoritmusok helyességének belátásra, a hibák felismerésére és kijavítására. Ez utóbbi látszólag az algoritmus egyszerű elemzését jelenthetné, az informatikában azonban ennél több lehetőségünk is van. Léteznek olyan komplex rendszerek (programfejlesztői környezetek), amelyek használatával ez a tevékenység a puszta gondolkodásnál hatékonyabbá tehető (bár természetesen nem pótolja a gondolkodást).

A kódolás jóval kevésbé fárasztó tevékenységgé tehető, ha megfogalmazzuk azokat a szabályokat, amelyekkel a megoldás lényegi leírását tartalmazó algoritmusból az adott nyelvű kódmegfelelőt hozzuk létre. Ilyen szabályok definiálhatók bármely nyelvhez, és az átültetés 90%-ig mechanikussá tehető. [9] Ára: a szabályok felismerése, megjegyzése.

A tesztelés „nem szeretem” kötelessége jóval több gondolkodást kíván, mint várnánk. A tesztelés célja: a kód helyesség-belátásának partitúrát, menetrendet adni. Ahhoz, hogy a célt elérhessük, módszert kell találni arra, hogy milyen adatokkal lehet a kódot minden pontján „megmozgatni”. Ehhez kézenfekvő a 3.-ként említett elemző képességre építeni, habár nem az egyedüli szisztéma. A feladat (tehát nem a megoldás!) értő ismerete segíthet a releváns esetek megtalálásában. A hibajelenség detektálása után a hibakeresésnél megint csak az elemző képesség jut szóhoz.

Meg kell jegyezzük, hogy a kódolás – az informatika oktatásban sem– csupán az algoritmus helyességének végső kimondásának az eszköze, hanem sokszor a problémamegoldás betetőzése (pl. a szimulációs modellezésnél). [10]

6. Algoritmus (tevékenységsorozat) módosítása, átalakítása

Mások által készített algoritmus megértése is viszonylag könnyű feladat, egy saját algoritmus elkészítése is viszonylag könnyű feladat ahhoz képest, hogy egy más által elkészített algoritmust módosítanunk, továbbfejlesztenünk kell.

Itt ugyanis nemcsak a végrehajtást kell elképzelnünk, hanem meg kell értenünk, hogy az eredeti algoritmus készítője hogyan gondolkodott, miért úgy készítette az algoritmusát, …

Azt is meg kell értenünk, hogy ebbe a más gondolatvilágba hol léphetünk be, mit módosíthatunk, milyen beavatkozás lesz hatásos, … Ez sokkal nehezebb lehet, mint a semmiből egy saját algoritmust készíteni.

Sokszor kapunk másoktól pontatlanul megtervezett algoritmusokat (az informatika világában programokat), amelyek nem felelnek meg a céljainknak. Ezért képesnek kell lennünk ezek átalakítására úgy, hogy számunkra hasznosak legyenek!

7. Komplex algoritmus (tevékenységsorozat) tervezése

Az algoritmusok méretének növekedésével a befektetett munka nem lineárisan növekszik. Előbb-utóbb elérünk arra a szintre, amikor egy lépésben nem látjuk át a megoldandó feladatot. Ekkor lehet határozottan erős szerepe az absztrakciónak, az algoritmusok szisztematikus tervezésének.

Itt jöhet szóba a részcélok kitűzése, az egyes részcélokhoz tartozó tevékenységsorozat megtervezése (a programozásban nem feltétlenül – bár nagy programrendszerek fejlesztésekor mindig, de a hétköznapi életben általában több ember együttes munkájaként). Természetesen az is komoly feladat, hogy belássuk: a részcélok teljesülésével, jó összeépítéssel a feladat megoldható.

Az algoritmikus gondolkodás és a NAT 2007 – áttekintő táblázat

	1-4	5-6	7-8	9-12
Az informatikai eszközök használata	Algoritmus megértése	Algoritmus megértése	Algoritmus megértése	Algoritmus megértése
Informatika alkalmazói ismeretek	—	Algoritmus végrehajtása	Algoritmus elemzése	Algoritmus alkotása
Infotechnológia	Algoritmus végrehajtása	Algoritmus megvalósítása	Algoritmus módosítása, átalakí-	Komplex algoritmus tervezése
Infokommunikáció	—	Algoritmus végrehajtása	Algoritmus elemzése	Algoritmus alkotása
Médiainformatika	—	—	—	—
Az információs társadalom	—	—	—	—
Könyvtári informatika	—	Algoritmus megértése	Algoritmus végrehajtása	Algoritmus végrehajtása

1. táblázat

A fenti táblázatban az *algoritmikus gondolkodás* kompetencia az adott korcsoportra és témakörre meghatározott legmagasabb szinten szerepel. Az infotechnológiában való megjelenése természetes (hiszen az szól az algoritmizálásról, adatmodellezésről, problémamegoldásról), a többi terület azonban némi magyarázatot igényel.

Az informatikai eszközök (hardver, szoftver) használata mindig valamilyen algoritmus végrehajtását igényli, egy szoftver installálása sokszor addig ismeretlen algoritmusok megértése, majd végrehajtása elé állít minket.

Az alkalmazói ismertek egy része is algoritmus alkalmazás, magasabb szinten azonban megjelenhet benne akár új algoritmus alkotása is. Ez kézenfekvő lehet pl. a táblázatkezelési feladatoknál, ahol az algoritmus bizonyos képletek egymástól függő kiszámítását igényelheti. Kevésbé kézenfekvő esetek is vannak. Az egyik idei országos alkalmazói verseny feladatsorában például volt olyan feladat, amiben a versenyzők egy táblázat nyersanyagát egy egyszerű szöveges állományban kapták meg. Itt a szöveg még nem táblázatban szerepelt, még csak nem is tabulátorokkal volt tagolva, hanem a táblázat elemek között annyi szóközzel kiegészítve, hogy a táblázat nyomtatva jól nézzen ki. Ebből a táblázatból valódi táblázatot előállítani vagy hosszú gépelési-másolási munka 30-40 percben, vagy pedig egy ügyes algoritmus megalkotásával 3 perces feladat.

Logikusan az infokommunikáció is algoritmus végrehajtásból áll, de itt is többről van szó. Kommunikációs hálózatok megalkotása, kommunikációs hálózatokon üzenetek eljuttatás a megfelelő körnek komoly algoritmus alkotási (és nem mellesleg adatmodellezési) képességet igényel.

Más elképzelések

Az algoritmikus gondolkodás szintjeit a következőkben lehet meghatározni, illetve kapcsolni más gondolkodási módokhoz (Buda Mariann javaslata alapján). [11 – a tanulmányban szereplő megállapításokat dőlten szedve]

1. szint

Alkalmazás esetén: amikor az emlékezetből csupán elő kell hívni azt a meghatározott (megadott, megnevezett) eljárást, amelynek segítségével a tanuló úrrá lesz az adott problémán. (Pl.: Határozd meg a közös nevezőt! Ekkor a tanuló kiválasztja, előkeresi a megnevezett eljárást, algoritmust.)

Végrehajtás: deduktív gondolkodás (az általános formában tárolt – vagy megadott – algoritmus alkalmazása a konkrét esetre).

Mi ezt a szintet két szintnek fogalmaztuk meg. Határozott állításunk, hogy mást jelent egy algoritmus megértésének, illetve végrehajtásának képessége. A végrehajtás ugyanis nem egyszerűen a megértést jelenti. Végrehajtáshoz pontosan követnünk kell a megoldás állapotait, tudnunk kell, hogy éppen milyen állapotban milyen tevékenység kerül sorra, …

2. szint

Az algoritmus megalkotása, vagyis az a folyamat, amelyben a lépéseket igyekszünk felismertetni, a szabályokat, általánosításokat kezdjük kialakítani; alapjaiban induktív gondolkodást igényel.

Lényege: a tanuló a problémamegoldás során megfigyel egy szabályszerűséget (szabályt), azt többször is észreveszi, majd ezt valamilyen értelmes módon kódolja (rögzíti).

Nálunk ez is két szint, meg kell különböztetni egy elemző képességet és egy alkotó képességet. Abból, hogy megértjük egy algoritmus elemeit, tudjuk azt, hogy minek mi a szerepe benne, még nem következik, hogy képesek vagyunk alkotni is algoritmusokat. Az egyik ugyanis egyértelműen analizáló tevékenység, a másik pedig szintetizáló.

3. szint

Az algoritmusok felkutatására, kiválasztására irányuló tudatos törekvés. Újabb hasonló esetben egészében vagy kissé módosított formában – adott probléma megoldásakor – felidézi, követi, használja.

Analogikus gondolkodás.

Úgy gondoljuk, hogy egy feladathoz algoritmus keresése, választása elemibb gondolkodást igényel, mint egy új algoritmus megalkotása. Egy korábban megismert algoritmus alkalmazásához nincs szükség feltétlenül új algoritmus tervezésére. Ez még akkor is igaz, ha az algoritmust adaptálni kell egy adott feladatra. Ezt egy hétköznapi példával illusztrálhatjuk: ha tudunk lágytojást főzni, akkor a főzési algoritmus módosításával képesek lehetünk keménytojás főzésre is. Ez azonban nem azt jelenti, hogy képesek vagyunk tetszőleges étel receptjét (azaz előállítási algoritmusát) megalkotni.

4. szint

Az algoritmus módosítása, „illesztése": az algoritmusok hatékony, rugalmas átalakítása, egybefűzése. Az alapalgoritmusból új eljárásmód kialakítása.

Kreativitás.

Ez a mi 6. és 7. szintünknek felel meg, a komplexitás kezeléséről azt gondoljuk, hogy az algoritmizálási képesség egy sokkal magasabb szintjét jelenti.

Algoritmikus gondolkodás és programfejlesztés

Az algoritmikus gondolkodási kompetenciával sokan, mint a programfejlesztéshez szükséges képességgel foglalkoznak. Ebben a cikkben nekünk nem ez a célunk, ezért csak egy (egyébként nagyon érdekes tanulmányból származó) rövid idézettel hivatkozunk erre az érdekes továbbgondolási lehetőségre:

The general concepts of algorithmic thinking, including functional decomposition, repetition (iteration and / or recursion), basic data organizations (record, array, list), generalization and parameterization, algorithm vs. program, top-down design, and refinement. Note also that some types of algorithmic thinking do not necessarily require the use or understanding of sophisticated mathematics.

Algorithmic thinking is key to understanding many aspects of information technology. Specifically, it is essential to comprehending how and why information technology systems work as they do. To troubleshoot or debug a problem in an information technology system, application, or operation, it is essential to have some expectation of what the proper behavior should be, and how it might fail to be realized. Further, algorithmic thinking is key to applying information technology to other personally relevant situations. [12]

Algoritmikus gondolkodást fejlesztő eszközök

Az algoritmikus gondolkodás elmélyítéséhez sokan programozási nyelvektől független fejlesztő eszközöket dolgoznak ki. Egyik ilyen legkorábbi eszköz volt Karesz, a robot [13, 14, 15].

Karesz egy „utcagyerek", aki egy négyzetrácsos úthálózat utcasarkain lépdelve mozog. Egyes utcasarkokon falak vannak, amelyeken Karesz nem tud átmászni. Más sarkokon kavicsok találhatók (egy helyen csak egy), amiket Karesz tetszőleges számban felvehet, majd újra letehet. Karesz, mint algoritmus végrehajtó az alábbi nyelvet érti:

Karesz tevékenységei	**Amit Kareszről kérdezhetünk**
➢ Indulj Karesz	➢ északra néz
➢ Fordulj jobbra	➢ délre néz
➢ Fordulj balra	➢ keletre néz
➢ Lépj	➢ nyugatra néz
➢ Vedd fel a kavicsot	➢ fal előtt áll
➢ Dobj el egy kavicsot	➢ van itt kavics

- Állj

Példák:

```
Fordulj északra:
  AMÍG NEM északra néz
  ISMÉTELD
    Fordulj jobba
  ISMÉTLÉS VÉGE.

Menj a falhoz:
  HA NEM fal van előtte
  AKKOR Lépj; Menj a falhoz
  HA VÉGE.
```

```
Menj előre 10-et:
  ISMÉTELD 10-szer
    Fordulj jobba
  ISMÉTLÉS VÉGE.

Cserélget:
  ISMÉTELD 10-szer
    HA van itt kavics
    AKKOR Vedd fel a kavicsot
    KÜLÖNBEN Dobj el egy kavi-
csot
    HA VÉGE
  ISMÉTLÉS VÉGE.
```

Karesz, a robot lényege:

- felülről lefelé tervezés támogatása, elsődleges az eljárásfogalom;
- adatfogalom elrejtése (állapotlekérdező műveletekkel helyettesítve), algoritmusok változók nélkül;
- eljátszható algoritmusok.

Újabban definiálták Karesz „3 dimenziós testvérét", Alice-t.

Alice is primarily a scripting and prototyping environment that allows the user to build virtual worlds and write simple programs to animate objects (e.g., animals and vehicles) in those worlds. Objects in Alice can move, spin, change color, make sounds, react to the mouse and keyboard, and more. By writing simple scripts, Alice users can control object appearance and behavior. During script execution, objects may respond to user input via mouse and keyboard. Each action is animated smoothly over a specified duration. In the rest of this paper, we will describe the Alice programming environment, and then propose how Alice may also be used to assist in supporting the development of algorithmic thinking for novice programmers. [16]

Conclusion

Áttekintve a témában fellelhető irodalmat, s végiggondolva saját tapasztalatainkat, egyértelműen megállapítható, hogy az algoritmikus gondolkodás képességét mindenki nagyon fontosnak tekinti. Sokan összefüggésbe hozzák a matematikai kompetenciával is. [12]

A mérvadó szerzők abban is egyetértenek, hogy ez a kompetencia az algoritmusok megértési, végrehajtási, elemzési, tervezési képességét jelenti. Szakmai nézeteltérések egyrészt abban vannak, hogy ezeket a képességeket hogyan építhetjük egymásra, másrészt pedig nagyon kevesen foglalkoznak a programozási tevékenységtől független, mindenki számára szükséges algoritmikus gondolkodási képességekkel. Mi éppen ezért ezt emeltük ki elsősorban a cikkünkben.

Bibliography

1. Győző Horváth, Péter Szlávi, László Zsakó: *Informatics (ICT) competencies.* 8th International Conference on Applied Informatics, Eger, Hungary, 2010.
2. *Recommendation of the European Parliament and of the Council of 18 December 2006 on Key Competences for Lifelong Learning* (2006/962/EC)
3. *Nemzeti Alaptanterv.* http://www.okm.gov.hu/
4. 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
5. Vass Vilmos: *A kompetencia fogalmának értelmezése*. Oktatáskutató és Fejlesztő Intézet, 2009.
6. Benczúr András: *A kommunikáció fejlődése és az információs forradalom*. (*Evolution of Communication and the IT Revolution*), Természet világa (The World of Nature, special edition), pp74-79, 2003.
7. J. Hvorecky, J. Kelemen: *Algoritmizácia, elementárny úvod*. ALFA, Bratislava, 1983.
8. Juraj Hromkovic: *Algorithmic Adventures – From Knowledge to Magic.* Springer, 2009.
9 Szlávi Péter, Zsakó László: *Módszeres programozás*. Műszaki Könyvkiadó, 1986.
10. Horváth László, Szlávi Péter, Zsakó László: *Modellezés és szimuláció*. ELTE IK, 2005.
11. Szántó Sándor: *Az algoritmikus gondolkodás fejlesztése*. Új Pedagógiai Szemle 2002 május, http://www.oki.hu/oldal.php?tipus=cikk&kod=2002-05-mu-Szanto-Algoritmikus
12. *Fluent With Information Technology by the National Research Council.* National Academy Press. June 1999. http://www.nap.edu/html/beingfluent/
13. C.H.A. Koster: *Systematisch leren programmeren*. Educaboek, 1984.
14. Hanák Péter: *Programozás ELAN-nal*. Műszaki Könyvkiadó, 1988.
15. R. Pattis: *Karel the Robot*. New York: John Wiley & Sons, 1981.
16. S. Cooper, W. Dann, R. Pausch: *Developing Algorithmic Thinking with Alice.* http://www.sju.edu/~scooper/alice/isecon00.PDF.