Gândire algoritmică: concept și analiză a resurselor educaționale digitale formative

Burlacu Natalia[1]
(1) Universitatea Tehnică din Moldova,
Facultatea Calculatoare, Informatică şi Microelectronică,
str. Studenților 9/7, mun. Chișinău, Rep. Moldova, MD-2020
natburlacu[at]hotmail.com

Abstract

The present material contains an analytical framework regarding the concept of algorithmic thinking (AT): semantics, the area of use, the training mode, the imperatives of developing and implementing of the cognitive skills related to the notion of the AT.
The is also given the description of modern digital educational resources through which AT can be formed and / or developed by interested peoples in different didactical circumstances: in the classroom and / or in the non-formal framework in this article.
The author considers that the concept of AT already acquires emergent features and in a very short time it will become a natural component in the functional literacy of children and adults in the during of the fifth industrial revolution, which is characterized as an era of artificial intelligence and interaction between the human and machine.

1. Introducere

La nivel de documente naționale este perceptibilă atât necesitatea formării competenței de gândire algoritmică (GA), cât și extinderea semanticii noțiunilor de algoritm și algoritmizare, etc. Fenomenul este actual în mai multe țări ale lumii.

Spre exemplu, în "*Cadrul de referință al Curriculumului Național*"[1], editat la Chișinău în 2017, noțiunea dată se întâlnește de mai multe ori cu diverse semnificații, cum ar fi: (1.) algoritmul de formulare a competențelor specifice; (2.) tipul problemei (de la cele algoritmice la cele inventiv-creative); (3.) strategii algoritmice referitoare la strategiile de predare-învățare; (4.) formarea algoritmului acțiunilor de implementare a curriculumului; (5.) algoritmul aplicării instrumentelor de evaluare sumativă; (6.) algoritmul evaluării aplicat asupra produselor activității didactice; (7.) competența de modelare și algoritmizare, etc. [p.10]. În anul curent, 2019, Ministerul Educației, Culturii și Cercetării al Republicii Moldova a lansat noua versiune a curriculumului național la aria curriculară Matematică și Științe[2], în care la disciplina Informatică pentru clasele VII-IX este stipulat că obiectul dat "[…] participă la formarea și dezvoltarea generală a personalității și are drept scop principal dezvoltarea gândirii algoritmice a elevului […]" [p. 4].

Idea de valorizare a gândirii algoritmice este propulsată mai mult sau mai puțin și în alte versiuni de Curriculum Național cam la toate treptele de învățământ, mai ales pentru ariile curriculare ce au tangență directă cu informatica, științele și matematica, atât în versiunile recente, cât și cele anterioare ale documentului în cauză. În contextul abordării integrate de studiere a Informaticii aceiași sursă, pledând pentru "[…] structurarea conținuturilor într-un model integrat, modular […]" subliniază disciplina școlară dată drept una ce are mai multe scopuri printre care și

[1] În: http://bit.do/faY3K, accesat 18 August 2019
[2] În: http://bit.do/faY3X, accesat 18 August 2019

cel de „ [...] cultivare continuă a modului de gândire algoritmică" [ibidem]. Mai mult decât atât, noţiunile derivate din cuvântul algoritm par a fi îndrăgite chiar şi de cei de la disciplinele umanistice. Deseori în manuale, elaborări metodice, documente reglatorii, curriculum întâlnim sintagme gen: "conform algoritmului", "în funcţie de algoritm", "reieşind din algoritmul prezentat" etc.

Conceptul de GA nu este unul nou, deşi, totodată, este unul vehiculat tot mai larg în publicaţii academice, acte oficiale, proiecte de legi, mass media, etc., căpătând astfel unele conotaţii şi/sau fiind utilizat în unele domenii ştiinţifice mai puţin sau chiar deloc afiliate domeniului de tehnologii informaţionale şi comunicaţie (TIC).

Considerăm că **conceptul de GA deja capătă trăsături emergente** şi, într-un timp foarte scurt, **va deveni o componentă firească în cadrul alfabetizării funcţionale** a copiilor şi adulţilor în "[...] epoca celei de-a cincea revoluţii industriale [...]" [3], caracterizate drept eră a inteligenţei artificiale şi a interacţiunii dintre om şi maşină.

Puterea gândirii algoritmice constă în faptul că îi permite purtătorului soluţionarea rapidă, eficientă, iar în cazurile sistemelor de calcul şi automatizată, a unor probleme.

Dacă avem deja elaborat algoritmul, nu trebuie să cercetăm modul de înmulţire de la zero de fiecare dată când ne confruntăm cu o nouă problemă, ca de altfel, nu trebuie să exersăm de la zero modul de conectare a maşinii de spălat, nu trebuie să învăţăm de la zero consecutivitatea acţiunilor de preparare a cafelei, etc. Pot fi aduse foarte multe exemple de acest fel.

Semantica, aria de utilizare, modul de formare, imperativele dezvoltării şi implementării abilităţilor cognitive relaţionate noţiunii de GA, dar şi instrumentele educaţionale digitale prin care acestea pot fi formate şi / sau dezvoltate la etapa iniţială, reprezintă câteva aspecte pe care ni le propunem spre analiză în prezentul articol.

2. Expoziţie retrospectivă a evoluţiei conceptelor derivate din noţiunea de algoritm

În cadrul unei analize retrospective efectuate am stabilit că încă în îndepărtatul 1960, Alan Perlis afirma că conceptul de „algoritmizare" era deja o parte a culturii de pe acea vreme. Savantul susţinea că "[...] calculatoarele vor automatiza şi transforma procesele în toate domeniile şi, astfel, algoritmizarea va apărea în cele din urmă în toate domeniile [...]" [3] de activitate umană. La mijlocul anilor '60, pionierii informaticii Allen Newell, Alan Perlis şi Herb Simon, apărau noul domeniu de criticii externi care afirmau că informatica nu poate fi fundamentată ca ştiinţă aparte "[...] deoarece calculatoarele sunt artefacte artificiale, pe când ştiinţa veritabilă se bazează doar pe studierea unor fenomene naturale" [4]. Allen Newell, Alan Perlis şi Herb Simon promovau idea că "[...] ştiinţele se formează în jurul unor fenomene pe care oamenii doresc să le valorifice, iar calculatoarele, ca şi transformatoare de informaţii, reprezintă un nou fenomen focal care [...]", până atunci nu fusese acoperit „[...] de nici un alt domeniu". În aceiaşi lucrare autorii caracterizează GA, drept un proces care "[…] distinge informatica de alte domenii [...]" prin abilitatea de „[...] proiectare a unei serii de instrucţiuni de maşină pentru gestionarea soluţiei de calcul a unei probleme" [ibidem].

În 1974, Donald Knuth afirmă că "[...] exprimarea unui algoritm este o formă de predare (către o maşină necuvântătoare, numind-o în original „dump machine") care duce la o înţelegere profundă a problemei; învăţarea unei abordări algoritmice ajută la înţelegerea conceptelor de toate tipurile în multe domenii" [1].

Cea mai reprezentativă lucrare în materie de gândire computaţională este "Computational Thinking" după Jeannette M. Wing [5]. Autoarea caracterizează gândirea computaţională drept o capacitate

și "[…] un set de abilități universal aplicabile tuturor [...] care [...] ar fi dornici să învețe și să le folosească" și nu doar celor care fac parte din domeniul profesional al tehnologiilor informaționale.

Este firesc că ulterior noțiunea de **computational thinking** sau **gândire computațională** (GC) a căpătat o amplă răspândire, îndeosebi, printre reprezentanții mediului academic preocupați de pregătirea specialiștilor pentru domeniile de TIC și / sau conexe Informaticii, Didacticii Informaticii, etc. De altfel, găsim reflecții interesante asupra conceptului de GA la autorul Chenglie Hu (2011). Cercetătorul consideră că noțiunea dată nu este încă percepută corect de mediul academic și că pe marginea semanticii noțiunii de GC încă se fac speculații, precum că GC este "[...] o paradigmă de gândire hibridă care trebuie să se adapteze diferitelor moduri de gândire [...]" [2] în ceea ce privește modul în care fiecare individ în parte influențează parcursul gândirii sale în timp ce efectuează anumite calcule.

În urma analizei opiniilor listate mai sus, noi venim cu idea că, reieșind din modul diferit în care sunt definite conceptele de GA și GC, dar și a ariei de circulație ale acestora putem afirma că:

- ✓ **gândirea algoritmică (GA)** este capacitatea de a rezolva anumite probleme / însărcinări / exerciții, etc. prin ***abilitatea de a aplica*** *clar pași / instrucțiunile necesare, care deseori sunt stipulate și / sau se află la suprafață etc.;*
- ✓ **gândirea computațională (GC)** este capacitatea de a rezolva anumite probleme / însărcinări / exerciții, etc. prin ***abilitatea de a elabora*** (înscris într-un proces segmentat în: a vedea, a căuta, a găsi) *un set de instrucțiuni, a determina un ansamblu de acțiuni bine definite într-un mod inteligibil pentru mașina de calcul*.

Astfel, noi le percepem ca noțiuni tangente, înrudite, dar care, totuși, au și careva deosebiri nu atât semantice, cât funcționale. GA, fiind o categorie mult mai largă care poate, dar deja este aplicată în mai multe domenii de activitate umană: științifică, industrială, economică, socială, etc. Dacă este vorba despre educație, la noțiunea de GC se poate recurge doar în anumite arii curriculare și la anumite niveluri se studii, acolo unde este solicitată nu doar gândirea abstractă.

3. Prezentarea resurselor educaționale digitale formative de gândire algoritmică

Actual, când educația digitală este introdusă în învățământ, începând cu ciclul primar, nu mai putem vorbi despre dezvoltarea GA în lipsa calculatorului, mai ales că pe piața ofertelor de resurse educaționale digitale există o mulțime de instrumente libere și / sau gratuite, de care este, pur și simplu, păcat să nu beneficieze cadrul didactic interesat de dezvoltarea acestei abilități la elevii / studenții săi. Respectivele resurse pot fi utile și altor categorii de persoane, copiilor și adulților, care doresc să-și perfecteze nu doar competențele digitale, dar și cele corelate cu GA, adică: logică, gândirea abstractă, gândirea critică, etc.

Reieșind din tematica abordată în prezentul material ne-am propus să descriem câteva instrumente abile să formeze și dezvolte GA, care în opinia noastră, deși sunt în tendințele actuale ale educației digitale, încă sunt puțin cunoscute, de aceia și mai puțin utilizate de publicul larg. Resursele date sunt grupate conform Gradului de dificultate (GD) și Categoriei de vârstă (CV). Gradul de dificultate, fiind specificat, începând cu nivelul *inițial* până la cel *avansat*, fiind recomandate de dezvoltatori utilizatorilor, începând cu vârsta preșcolară mare până la vârsta corespunzătoare nivelului liceal superior / universitar.

A. ***www.spritebox.com*** – GD / CV: inițial – mediu inițial / preșcolar mare – gimnazial. SpriteBox Coding (SBC) este prima resursă proiectată cu codificatoare și a fost dezvoltată de aceeași echipă care a creat LightBot, o aplicație pentru învățarea coding-ului. Statisticile afirmă că în aplicația

dată s-au jucat peste 20 de milioane de copii şi a fost folosită deja de zeci de mii de profesori din întreaga lume. SBC este un joc plin de aventură care oferă utilizatorilor oportunitatea de a se iniția în programare. Etapa inițială a software-lui prevede instruirea de la zero a utilizatorului, propunându-i rezolvarea de puzzle-uri de codare, folosind pictograme, etc. La etapa medie, în timp util, pictogramele sunt înlocuite cu comenzi textuale. La etapa superioară, relativ rapid, utilizatorii pot învăța să soluționeze puzzle-uri într-un limbaj de programare real, aici fiind prezentate conținuturi didactice mult mai complexe celor de la etapele anterioare.

B. ***https://lightbot.com/*** – GD / CV: inițial – mediu inițial / preşcolar mare – gimnazial.

Figura 1. Pasul I: elaborarea vizuală a procedurii deplasării robotului și aprinderii becurilor din butoane

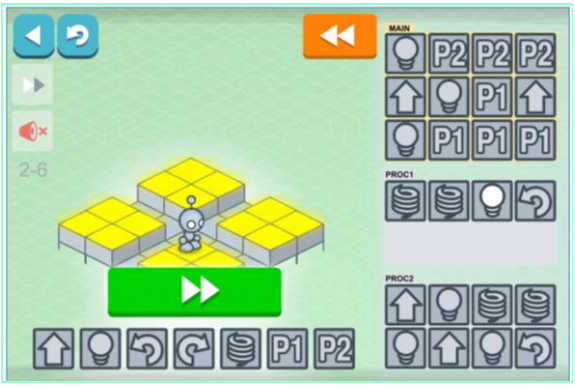

Figura 2. Pasul final: elaborarea vizuală a procedurii deplasării robotului și aprinderii tuturor becurilor incorporate în butoane

Resursa reprezintă un joc video educațional pentru învățarea conceptelor de programare software. Jocul a fost dezvoltat de Danny Yaroslavski. Lightbot a fost jucat de 7 milioane de ori și este foarte apreciat pe iTunes și magazinul Google Play. Lightbot este disponibil ca un joc flash online, dar și aplicație pentru telefoanele mobile Android și iOS. Lightbot a fost creat cu Flash și OpenFL. Resursa are interfață elaborată pentru mai multe limbi europene. Resursa conține o captură de ecran de Lightbot cu un robot care se mișcă, inițial plasat pe suprafața de lucru a aplicației și un panou de comandă (aflat pe dreapta). Scopul aplicației Lightbot este de a gestiona mișcarea unui mic robot pentru a-l face să navigheze într-un labirint, aprinzând luminile pe suprafața unor butoane. Jucătorii aranjează simbolurile pe ecran pentru a comanda robotului să meargă, să se întoarcă, să sară, să aprindă o lumină și așa mai departe (vezi Figurile 1-2). Labirintul și lista de simboluri devin mai complicate pe măsură ce complexitatea sarcinilor progresează. În timp ce folosesc astfel de comenzi, jucătorii învață concepte de programare, precum buclele, procedurile și multe altele, fără a introduce cod în niciun limbaj de programare.

C. ***https://code.org/*** – GD / CV: inițial – avansat / preşcolar mare – liceal superior. Resursa aparține organizației non-profit cu același nume, care este și dezvoltatorul site-ului web omonim condus de Hadi Partovi (care și-a propus să motiveze oamenii, să învețe știința calculatoarelor). În încercarea de a încuraja instituțiile de învățământ preuniversitare să planifice mai multe ore de informatică în curriculum, site-ul web include lecții gratuite de codificare. Hour of Code a fost lansat la 9 decembrie 2013 pentru uz la nivel național, având ca scop promovarea studierii științei calculatoarelor. Evenimentul de lansare a avut loc în cadrul Săptămânii Educației Informatice (în perioada din 9/12-2013 până la 15/12-2013). Platforma https://code.org/ găzduiește resurse

educaționale pentru studierea științei calculatoarelor, acestea fiind clasificate în funcție de utilizatori (elevi, profesori, materiale didactice) și nivele de complexitate. Resursele / activitățile didactice pot fi accesate gratuit, în regim on-line. Rezultatele activităților elevilor și profesorilor pot fi salvate doar fiind create conturi pe prezenta platformă.

D. https://studio.code.org/courses – GD / CV: inițial – avansat / preșcolar mare – liceal superior. Resursa reprezintă un sub compartiment al https://code.org/. Dezvoltatorii îl recomandă drept un *Atelier de Programare* în care sunt prezintate cursuri organizate pe 4 nivele. Tot aici sunt amplasate linkuri spre alte compartimente ale site-ului, cum ar fi: Oră de Programare (Code of hour). Dezvoltatorii susțin că resursa este utilizată de milioane de elevi și profesori din peste 180 de țări. Nivelele incluse de elaboratori în resursa dată sunt: (a.) **Cursul 1** este recomandat elevilor din clasa a 1-ia; a fost proiectat pentru a le permite utilizatorilor începători, interesați să creeze programe de calculator, să învețe să colaboreze cu alții, să-și dezvolte abilitățile de rezolvare a problemelor. (b.) **Cursul 2** este recomandat elevilor din clasele a 2-5-cea și a fost conceput pentru elevii care deja pot lectura, dar nu au nici o experiență anterioară de programare. (c.) **Cursul 3** este recomandat elevilor din clasele a 4-5-cea și a fost conceput pentru elevii care au parcurs cursul 2. Aici elevii vor aprofunda subiectele introduse în cursurile anterioare, pentru a le adapta la soluții pentru probleme mai complexe de programare. (d.) **Cursul 4** este recomandat elevilor din clasele a 4-8-a; cursul a fost conceput pentru cei care au terminat cursurile 2 și 3. Cursanții vor învață cum să abordeze puzzle-urile de complexitate sporită prin combinarea mai multor concepte. Până la terminarea acestui curs, elevii vor crea programe care să pună în evidentă aptitudini multiple, inclusiv lucrul cu bucle și funcții cu parametri. Compartimentul Oră de Programare (Code of hour) vine cu un set de tutoriale cu durata de o oră, concepute pentru toate vârstele. Tutorialele sunt elaborate în baza unor subiecte preluate din basme și / sau axate în fabula unor jocuri video și / sau de calculator (vezi Figurile 3-4; Tabelul 1). În continuare pe site sunt prezentate: un catalog complet de cursuri (resursa este disponibilă doar în limba engleză), dar și un curriculum, recomandat de dezvoltatori pentru studierea științei calculatoarelor.

Figura 3. Animarea personajului Frozen, creată prin blocuri de cod

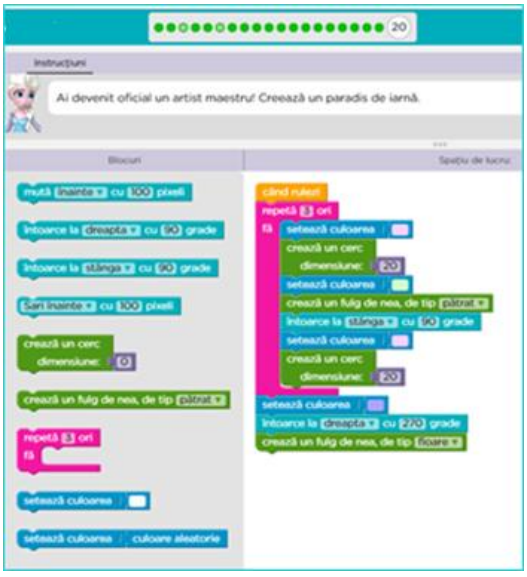

Figura 4. Algoritmul animării lui Frozen, reprezentat în blocuri de cod

Tabelul 1

Listingul codului în JavaScript de Animare a personajului Frozen, generat de resursă la elaborarea algoritmului prin coding

```
for (var count3 = 0; count3 < 3; count3++) {
  penColour('#fdd0fd');
  // create_a_circle
  for (var count = 0; count < 36; count++) {
    moveForward((20));
    turnRight(10);
  }
  penColour('#d0fdd0');
  drawSnowflake('square');turnLeft(90);
  penColour('#fdd0fd');
  // create_a_circle
  for (var count2 = 0; count2 < 36; count2++) {
    moveForward((20));
    turnRight(10);
  }
}
penColour('#aea4ff');
turnRight(270);
drawSnowflake('flower');
```

E. ***https://scratch.mit.edu*** – GD / CV: inițial – avansat / preşcolar mare – liceal superior. Serviciul este dezvoltat de MIT Media Lab; a fost tradus în peste 70 de limbi și își are circulația în majoritatea țărilor lumii. Scratch-ul este învățat și utilizat în programele after-school, școli și colegii, precum și în alte instituții educaționale publice. Din mai 2019, statisticile comunității de pe site-ul oficial al dezvoltatorului indică mai mult de 40 de milioane de proiecte diseminate între peste 40 de milioane de utilizatori, cu cca 40 de milioane de vizite lunare. Scratch este un limbaj de programare vizuală, bazat pe blocuri și dedicat unei comunități online ce vizează, în principal, copiii. Utilizatorii site-ului pot crea proiecte online, folosind o interfață asemănătoare unui bloc. Utilizarea Scratch permite tinerilor să înțeleagă logica programării, prin activități de colaborare creativă bazate pe construirea blocurilor de programe. Scratch a fost popularizat în Marea Britanie prin intermediul Code Clubs. Scratch-ul este folosit ca limbaj introductiv de coding, deoarece permite crearea unor programe interesante relativ ușor, iar abilitățile învățate de utilizatori pot fi aplicate în studierea altor limbaje de programare, cum ar fi Python și Java. Prin vizualizările furnizate, beneficiarii pot crea animații, text, povești, muzică și multe altele. Există deja multe programe pe care elevii le pot utiliza pentru a învăța subiecte de matematică, istorie și, chiar, fotografie. Scratch permite profesorilor să creeze lecții conceptuale cu sarcini de laborator, având tentă științifică, prin utilizarea de animații care ajută perceperea vizuală a conceptelor dificile. Pentru studierea științelor sociale, instructorii pot crea teste, jocuri și tutoriale cu elemente interactive.

F. ***https://www.bootstrapworld.org/*** – GD / CV: Inițial – avansat / Preşcolar mare – liceal superior. Bootstrap este un cadru CSS gratuit și open-source, direcționat spre dezvoltarea receptivă și mobilă de Front-End Web. Conține CSS și (opțional) JavaScript, șabloane de proiectare pentru editarea de formulare, butoane, elemente de navigare și alte componente de interfață. Bootstrap este al treilea proiect inițiat și dezvoltat de cele mai mari vedete de pe GitHub, care a adunat peste 131.000 de stele, după el fiind plasate astfel de proiecte precum: freeCodeCamp (cu un rating de cca 300.000 de stele) și Vue.js. Potrivit Alexa Rank, Bootstrap getbootstrap.com se află în top-2000 în SUA, în timp ce vuejs.org este în top 7000 în SUA. Platforma Bootstrap creează module

curriculare bazate pe cercetare pentru clasele 6-12, furnizând materialele care consolidează conceptele de bază din algebră ce permit profesorilor non-informaticieni să-şi adapteze materialele didactice, oferind, în acelaşi timp, conținuturi de calcul riguros şi antrenant, transpus din cursurile de profil informatic de la astfel de centre universitare, precum: universitățile Brown, WPI şi Northeastern. Resursele dislocate pentru nivelul inițial pot fi integrate independent la orele de informatică sau la cele de matematică de tip mainstream. Acestea, fiind aliniate la standardele naționale ale SUA pentru studierea matematicii. Celelalte module accesibile pe platformă modelează şi simulează fenomene din fizică, din domeniul Data Science, dar şi programe interactive sofisticate care pot fi integrate în cursurile de ştiință, matematică, etc. Profesorii pot mixa şi potrivi conținuturile didactice din diferite module astfel, încât să satisfacă necesitățile educaționale ale contingentului oricărei clase.

G. ***https://www.anaconda.com*** – GD / CV: mediu – avansat / liceal superior / universitar. Anaconda este o distribuție gratuită şi open-source a limbajelor de programare Python şi R, dedicate calculului ştiințific în domenii precum: Data Science, Machine Learning Applications, Large-Scale Data Processing, Predictive Analytics, etc.), care urmăreşte simplificarea gestionării pachetelor şi implementărilor. Distribuția Anaconda este folosită de peste 15 milioane de utilizatori şi include peste 1500 de pachete pentru studiul ştiințelor, compatibile cu Windows, Linux şi MacOS; conține mai multe pachete printre care este şi Anaconda Cloud care se prezintă drept un serviciu de gestionare a pachetelor ce permite găsirea, accesarea, stocarea şi partajarea notebook-uri, mediilor şi pachetelor conda şi PyPI publice şi private. Vom descrie doar câteva pachete incorporate în Anaconda. (1.) **JupyterLab** - proiectul sprijină ştiința interactivă a datelor şi calculul ştiințific în toate limbajele de programare prin dezvoltarea de software open-source; (2.) **Notebook Jupyter** oferă un REPL bazat pe browser, bazat pe o serie de biblioteci open-source populare, cum sunt: IPython; ØMQ; Tornado (server web); jQuery; Bootstrap (cadru front-end); MathJax; QtConsole; Spyder; Glueviz, etc.; (3.) **Cod Studio vizual** este un editor de cod sursă dezvoltat de Microsoft pentru Windows, Linux şi MacOS. Include suport pentru depanare, control Git încorporat şi GitHub, evidențierea sintaxei, completarea inteligentă a codului, fragmente şi re factorizarea codului. Cod Studio vizual este foarte personalizabil şi permite utilizatorilor să modifice tema, comenzile rapide de la tastatură, preferințele şi să instaleze extensii care adaugă funcționalitate.

Concluzii

MOTTO: *"Excelența nu este o abilitate, ci o atitudine."*
(Ralph Marston)

Dezvoltarea GA este una din competentele necesare în secolul XXI oricărui cursant (elev / student), dar şi adult deja integrat total în viața socială şi / sau profesională. În opinia noastră competență de GA poate şi trebuie să fie formată nu doar în cadrul orelor de studii corelate cu ştiința calculatoarelor şi matematica, ci şi în cadrul altor discipline, în funcție, de contextul didactic în care obiectivul de formare a GA este posibil de integrat, mai ales că actual există şi premise, dar şi resurse educaționale digitale care permit atingerea acestui obiectiv atât într-un cadru auditorial, cât şi într-un regim after-school, dar şi de sine stătător de cei care sun dornici să învețe.

Fiind dezvoltată sistemic, GA devine funcțională şi operațională pentru acei care o posedă şi o pot aplica în diverse circumstanțe, după caz: de soluționare a diferitor probleme cu care se confruntă nu doar la lecțiile de Informatică şi TIC, dar şi la alte obiecte de studiu şcolar / academic, fie formulate în scopuri didactice de profesor, inspirate din manual sau alte surse multimedia şi / sau oferite de realitatea socială, economică, ştiințifică cu care oricum interacționăm în viața cotidiană.

Bibliografie

[1] Denning, Peter J. *Remaining Trouble Spots with Computational Thinking*. COMMUNICATIONS OF THE ACM. Volume 60. Issue 6. pp. 33-39. Publication Date 2017-05-24. Publisher: ACM New York, NY, USA. ISSN: 0001-0782
[2] Hu, Chenglie. *Computational thinking: what it might mean and what we might do about it.* ITiCSE '11 Proceedings of the 16th annual joint conference on Innovation and technology in computer science education. pp. 223-227. 398 p. Publication Date 2011-06-27. Publisher ACM New York, NY, USA, 2011. ISBN: 978-1-4503-0697-3
[3] Muir. George. *AI – The Fifth Industrial Revolution.* În: http://bit.do/faY4g, accesat 18 August 2019
[4] Newell, A., Perlis, A.J., Simon, H.. *Computer Science,* [letter] Science 157 (3795): (Sept. 1967), 1373–1374
[5] Wing, Jeannette M.. *Computational Thinking.* pp. 33-35. COMMUNICATIONS OF THE ACM March 2006/Vol. 49, No. 3. În: http://bit.do/faY3S, accesat 18 August 2019