

DEZVOLTAREA GÂNDIRII COMPUTAȚIONALE ÎN ÎNVĂȚĂMÂNTUL PREȘCOLAR CU AJUTORUL ROBOȚILOR EDUCAȚIONALI

1. INTRODUCERE

Lucrând ca profesor pentru învățământ timpuriu, mi s-a părut firesc să îmi desfășor cercetarea cu copii de vârstă preșcolară. Acest lucru este important nu numai pentru că sunt zilnic alături de ei și observ modul în care gândesc și lucrează, ci și pentru că această vârstă (3-6 ani) este o perioadă deosebit de incitantă, marcată de schimbări majore.

Conform lui Brown și colab.(2012), perioada preșcolară este o perioadă de dezvoltare mentală intensă, în care apar pentru prima dată o serie de abilități psihologice, care continuă să se dezvolte și să se perfecționeze până la vârsta de adult tânăr . De asemenea, dezvoltarea creierului la această vârstă poate fi caracterizată ca fiind similară cu "înflorirea", deoarece atunci se manifestă cele mai dinamice modificări anatomice și fiziologice. Pornind de la aceste considerente, am ales ca temă de cercetare evaluarea și dezvoltarea abilităților de gândire la copiii preșcolari. Dintre diversele abilități de gândire, m-am concentrat în special pe cele care sunt abordate în prezent ca subiect de cercetare. Astfel am ajuns să studiez caracteristicile gândirii algoritmice/computaționale. Ne planificăm viața de zi cu zi în funcție de algoritmi și este important să îi învățăm pe copii de la o vârstă fragedă să urmeze o cale logică, un proces de gândire conceput în rezolvarea problemelor (Szántó, 2002) Conceptul de gândire computațională a fost introdus de doi cercetători, Jeannette Wing și Seymour Papert. Încă din 1980, Papert subliniază , în abordarea sa construcționistă (Papert,1980), rolul esențial al implicării sociale și emoționale în utilizarea programării ca instrument interdisciplinar. Wing a subliniat în 2006 rolul central al gândirii computaționale în dezvoltarea competențelor tehnice și a formulat, de asemenea, ideea că gândirea computațională va deveni a patra abilitate fundamentală în secolul XXI, alături de citit, scris și aritmetică. Gândirea computațională este o competență complexă care nu se concentrează doar pe utilizarea unui computer. Gândirea computațională presupune o înțelegere a rezolvării problemelor, a proiectării sistemelor și a comportamentului uman (Wing, 2008). În cazul copiilor preșcolari, Bers (2018) a definit gândirea computațională ca fiind capacitatea de a abstractiza comportamentele computaționale și de a identifica erorile. Criticile referitoare la gândirea

compuțională (de exemplu, Mannila și colaboratorii săi, 2014) îi acuză pe oamenii de știință din domeniul informaticii că doresc să transforme fiecare copil într-un dezvoltator de software, însă aceasta este o caricatură grosolană a ceea ce susține Wing. Scopul nu este de a învăța dezvoltarea de software, ci de a preda informatica pentru a oferi oamenilor instrumentele necesare pentru a înțelege lumea noastră digitalizată și a acționa în ea. Deși mulți cercetători s-au ocupat de acest subiect, până în prezent nu există încă o definiție universal acceptată în ceea ce privește gândirea compuțională. Conform definiției lui Selby și Woollard (2014), gândirea algoritmică este, în esență, un subansamblu al gândirii compuționale. În timp ce gândirea algoritmică se concentrează pe dezvoltarea de proceduri pas cu pas, care facilitează rezolvarea problemelor concrete, gândirea compuțională include un set mai cuprinzător de abilități care depășesc algoritmi specifici.

Structura lucrării mele urmează cronologic cercetările efectuate, pe baza cărora am explorat și analizat tema aleasă. Partea de după introducere conține o revizuire a literaturii de specialitate, adică o explicare a conceptelor-cheie, cum ar fi activitățile STEM, gândirea algoritmică/compuțională. Aceasta este urmată, în cadrul aceleiași secțiuni, de prezentarea instrumentelor de evaluare a gândirii algoritmice/compuționale, precum și a celor mai frecvent utilizate metode și instrumente de dezvoltare.

A treia parte constă în prezentarea a trei cercetări. Prima cercetare a vizat activitățile STEM, în care am acoperit integrarea cunoștințelor tehnice și dezvoltarea abilităților de gândire algoritmică și compuțională. La întrebările despre activitățile STEM au răspuns 85 de studenți ai Departamentului de Pedagogie și Didactică Aplicată -secția maghiară din cadrul Universității Babeș-Bolyai (Bálint-Svella și Zsoldos-Marchis, 2022). Pe lângă cei 85 de studenți, la întrebările privind includerea cunoștințelor digitale și a gândirii algoritmice și compuționale au mai răspuns și 115 educatoare active (Bálint-Svella și Zsoldos-Marchis, 2022, Bálint-Svella și Zsoldos-Marchis, 2024). Scopul acestei cercetări a fost de a evalua gradul de conștientizare a educatorilor preșcolari activi precum și a candidaților pentru funcția de educator cu privire la potențialul utilizării tehnologiei în activitățile STEM în învățământul preșcolar și la competențele care pot fi dezvoltate prin intermediul acestor activități. Această evaluare a situației a fost urmată de dezvoltarea, pretestarea și revizuirea testului AlgoPaint de gândire compuțională (Zsoldos-Marchis și Bálint-Svella, 2023). Pentru a treia cercetare, am elaborat un plan de dezvoltare care folosește roboți educaționali pentru a ajuta la dezvoltarea gândirii compuționale la copiii preșcolari (Bálint-Svella, 2023). Programul de intervenție a fost pre-testat, apoi

revizuit și aplicat în trei grupe de grădiniță.

Nu doar elaborarea instrumentului de evaluare este considerată o noutate, ci și planul de intervenție, care se bazează pe roboți educaționali și vizează grupa de vârstă a preșcolarilor. În sistemul educațional din România, dezvoltarea bazată pe utilizarea roboților educaționali nu este integrată în practica cotidiană. Diverse cercuri de specialitate și activități extrașcolare oferă oportunități în acest sens, vizând în majoritatea cazurilor copiii de vârstă școlară. Deoarece în perioada preșcolară se dezvoltă numeroase abilități și competențe, cercetările confirmă că în această perioadă se poate începe și dezvoltarea abilităților de gândire computațională. Cercetarea mea răspuns acestei lacune și nevoi, deoarece în literatura de specialitate din România nu există exemple în acest sens.

Ceea ce am constatat în cadrul cercetării mele este că roboții de podea pot fi un instrument excelent de joacă pentru copii, deoarece sunt atractivi pentru ei. Un alt mare avantaj al acestor sarcini prin joacă este că pot fi folosite pentru a verifica corectitudinea executării sarcinii și pentru a oferi copilului feedback imediat cu privire la performanța sa. În acest fel, ele nu numai că oferă distracție, dar contribuie și la dezvoltarea copilului. În timpul îndeplinirii sarcinilor, copiii au constatat că este permis să greșească, iar această greșeală poate fi corectată, pot încerca din nou și pot începe procesul de la început. Pentru copil există posibilitatea de a încerca de mai multe ori, putând exersa până când soluția devine cu adevărat corectă, iar în acest timp își poate identifica cu exactitate greșelile. Aceasta este o experiență foarte importantă, deoarece, la această vârstă, copilul poate deja să privească sarcinile din perspectiva rezolvării problemelor, mai degrabă decât să-și măsoare performanța în funcție de punctele finale corect-greșit, succes-eșec. Majoritatea sarcinilor planificate s-au desfășurat în perechi sau în grupuri, astfel încât rezolvarea în comun a problemelor, sprijinul reciproc, alegerea comunicării potrivite, colaborarea și cooperarea au făcut parte din îndeplinirea sarcinilor.

În cercetările mele am avut în vedere obiectivitatea și științificitatea, lucrarea mea conținând munca din ultimii patru ani și prezentând o mică parte a educației preșcolare din România.

2. FUNDAMENTE TEORETICE

2.1. Activități STEM în grădiniță

2.1.1. Prezentarea activităților STEM

STEM este un acronim format din primele litere ale cuvintelor știință, tehnologie, inginerie și matematică, care se referă la interconectarea acestor domenii (English, 2016). Activitățile STEM includ sectoare ale științei, tehnologiei, ingineriei, matematicii și integrează aceste informații și cunoștințe în baza de cunoștințe existentă la nivelul copiilor preșcolari, prin joacă. (Pantoya și colab., 2015, DeJarnette, 2018). Cunoștințele științifice, matematice, tehnologice și de inginerie se interconectează în mod continuu în contextul vieții de zi cu zi. Acestea furnizează cunoștințe, abilități și instrumente pentru a îmbunătăți calitatea vieții umane. În ultimul deceniu, în Statele Unite ale Americii a existat o îngrijorare crescândă cu privire la starea educației în domeniul științei, tehnologiei, ingineriei și matematicii (English, 2016; National Research Council [NRC], 2014). Conform datelor, doar 16% dintre elevii americani de liceu sunt competenți în matematică, iar Statele Unite ocupă locul 17 în domeniul științei printre națiunile industrializate. Este extrem de important pentru toată lumea, dar în special pentru tineri, să aibă un nivel adecvat de cunoștințe științifice și tehnologice pentru a putea deveni membri creativi și activi ai societății (English 2016, NGSS lead states, 2013).

2.1.2. Aplicarea activităților STEM în mediul preșcolar

În zilele noastre, educația STEM (Știință, Tehnologie, Inginerie și Matematică) joacă un rol foarte important în soluționarea provocărilor lumii moderne. Cunoștințele STEM sunt esențiale nu doar în numeroase locuri de muncă, ci și în viața de zi cu zi. Cercetările indică faptul că copiii trebuie să fie implicați în activități STEM de la o vârstă fragedă, deoarece perioada preșcolară este o perioadă sensibilă pentru dezvoltarea abilităților fundamentale de gândire (Driscoll și Nagel, 2008). Potrivit unui studiu pe termen lung realizat de Wai și colab. (2010), cei care au avut contact cu activități STEM în primii ani de viață, au avut abilități matematice ridicate în perioada adultă. Experiențele STEM pozitive ale copiilor, în primii ani de viață, sunt importante pentru dezvoltarea abilităților care îi vor ajuta să facă față provocărilor vieții (Lippard și colab., 2019), și în același timp, aceste experiențe timpurii influențează semnificativ performanța școlară (Watts și colab., 2014). În primii

ani, educația STEM trebuie să fie focalizată pe copii și bazată pe rezolvarea problemelor (Fridberg și colab., 2022). Aceasta trebuie să fie realizată în principal prin activități practice, care influențează pozitiv atitudinea copiilor față de STEM (Ortiz-Revilla și colab., 2021). Introducerea și practicarea activităților STEM în învățământul preșcolar este esențială pentru dezvoltarea copiilor. Mai multe studii științifice indică faptul că aceste activități oferă beneficii semnificative în dezvoltarea cognitivă, socială și emoțională. Activitățile STEM stimulează abilitățile de rezolvare a problemelor și gândirea critică a copiilor. Un studiu sugerează că educația timpurie în domeniul STEM contribuie la înțelegerea de către copii a conceptelor matematice și științifice de bază, ceea ce oferă ulterior un avantaj și în performanța școlară (Clements și Sarama, 2011). Preșcolarii care participă la astfel de activități obțin performanțe mai bune la matematică și științe de-a lungul anilor de școală (Duncan și colab., 2007). De asemenea, activitățile STEM îi stimulează pe copii să lucreze în echipă și să coopereze, deoarece implică adesea munca în comun, ceea ce dezvoltă și abilitățile sociale și inteligența emoțională (Brenneman, 2011). De exemplu, atunci când copiii construiesc împreună un dispozitiv simplu sau efectuează experimente, ei învață cum să comunice eficient, cum să își împărtășească ideile și să se sprijine reciproc. Activitățile STEM stimulează creativitatea și inovația. Copiii se confruntă cu probleme care necesită gândire creativă pentru a fi rezolvate, ceea ce îi ajută să-și dezvolte viitoarele abilități de inovare (Resnick, 2017). Cercetările au arătat că acei copii care au beneficiat de educație STEM timpurie au o probabilitate mai mare de a alege ulterior o carieră în STEM, ceea ce este extrem de important pe piața modernă a muncii (Tai și colab., 2006). Acest lucru este deosebit de important din perspectiva egalității de gen și sociale, deoarece întâlnirea timpurie cu aceste domenii poate ajuta la reducerea diferențelor de gen în domeniile STEM (Fundatia Națională pentru Știință, 2019).

În cadrul cercetării lor, Pantoya și colaboratorii săi (2015) au dezvoltat un program pentru copiii cu vârsta cuprinsă între 3-7 ani, pentru cultivarea abilităților ingineresti. Cercetătorii au dezvoltat și testat o carte de povești ingineresti care sporește cunoștințele, creativitatea și interesul copiilor pentru inginerie. Rezultatele au arătat că această lectură și activitățile conexe pot ajuta copiii să înțeleagă conceptele STEM și să-și dezvolte identitatea inginerescă. Autorii sugerează că ar trebui pus un accent mai mare pe inginerie și conținutul STEM în învățământul preșcolar și primar, sprijinind astfel succesul viitor al copiilor în aceste domenii.

În România, *Curriculum-ul de Educație Timpurie* (Ministerul Educației Naționale, 2019) include mai multe domenii de experiență, dintre care cunoștințele de mediu și matematica

sunt domenii care pot fi considerate parte a STEM. Diferite domenii de experiență sunt conectate în activități. Lipesc din curriculumul pentru educația timpurie elementele de dezvoltare a cunoștințelor și competențelor de inginerie și tehnologie, care fac parte din educația STEM.

2.2. Gândirea algoritmică/computațională la grădiniță

Algoritmii sunt parte integrantă a vieții noastre de zi cu zi, utilizăm în mod conștient sau inconștient algoritmi în activitățile noastre cotidiene. Ne desfășurăm activitățile zilnice conform unor algoritmi specifici, care aduc stabilitate, siguranță și ordine în viața noastră. Un algoritm este o procedură deterministă care poate fi aplicată oricărui element din clasa de intrări simbolice și care produce pentru fiecare astfel de intrare o ieșire simbolică corespunzătoare (Rogers, 1972). „Gândirea algoritmică este o modalitate de a ajunge la o soluție prin definirea clară a pașilor” (Curzon și colab., 2014, p. 2). Generalizând, „gândirea algoritmică este un sistem de metode de gândire necesar pentru a construi o serie de rezultate intermediare, pentru a planifica structura acțiunilor și a le executa, ceea ce duce la atingerea obiectivului” (Sadykova și Il'bahtin, 2019, p. 421).

În ceea ce privește conceptele de gândire algoritmică și gândire computațională, Hromkovich și colab. (2017) consideră că sunt două formulări echivalente și interschimbabile ale aceleiași concepții, în ciuda faptului că au fost introduse în epoci diferite, deoarece gândirea algoritmică este utilizată de câteva decenii, în timp ce conceptul de gândire computațională este un termen mult mai recent. Autorii consideră că gândirea algoritmică este înrădăcinată în nucleul științific al informaticii, așa cum sugerează și definiția lui Aho (2012): gândirea computațională poate fi văzută ca un proces de gândire implicat în formularea problemelor, astfel încât soluțiile acestora să poată fi prezentate ca pași computaționali și algoritmi.

Conceptul de gândire computațională nu este complet nou, a trecut printr-o lungă evoluție. A fost menționat pentru prima dată de Seymour Paper (1980), care a fost dezvoltatorul limbajului de programare LOGO, dar expresia a fost popularizată de Wing (2006), care a introdus-o ca o prescurtare a expresiei "a gândi ca un informatician". Informal, gândirea computațională descrie activitatea mentală implicată în formularea unei probleme pentru care se caută o soluție computațională. Gândirea computațională nu se referă doar la rezolvarea problemelor, ci și la formularea lor. Gândirea computațională este, de fapt, un proces prin care o persoană dezvoltă numeroase strategii de gândire în abordarea unei probleme date. În gândirea computațională, cel mai important și cel mai înalt nivel al

procesului de gândire este abstractizarea. Folosim abstractizarea pentru a defini modele, precum și pentru a evidenția trăsăturile esențiale comune, în timp ce ascundem caracteristicile neesențiale dintre ele. Potrivit lui Wing, există două aspecte fundamentale ale gândirii computaționale: crearea de abstracții (forme de abstracție în informatică: formulări algoritmice și modularitate) și implementarea abstracțiilor. Articolul lui Wing (2006), în care susține că gândirea computațională este un set de atitudini și abilități universal aplicabile, la fel de importante ca cititul, scrisul și aritmetica, a stârnit o mare dezbatere. Deși nu există o definiție unanim acceptată, definiția oferită de Brennen și Resnick (2012) este una dintre cele mai acceptate. Autorii împart gândirea computațională în următoarele componente: concepte computaționale (concepte legate în principal de informatică și programare, adică „secvențe, bucle, paralelism, evenimente etc.”), practici computaționale (strategiile și practicile necesare pentru aplicarea conceptelor menționate) și perspective computaționale („punctele de vedere ale proiectanților asupra lumii și asupra lor înșiși”).

Selby și Woollard (2014) au urmărit și analizat evoluția și schimbările definiției gândirii computaționale propuse de Wing în literatura de specialitate. Conform recomandărilor lor, gândirea computațională este un proces cognitiv sau de gândire care reflectă capacitatea de a gândi abstract, capacitatea de a gândi fragmentat, capacitatea de a gândi în algoritmi, capacitatea de a gândi în evaluări și generalizări. Această definiție include doar acei termeni asupra cărora există un consens în literatura de specialitate.

Conform studiilor lui Gadzikowski (2019), gândirea computațională este în general o combinație a patru categorii de competențe: recunoașterea tiparelor, crearea și aplicarea algoritmilor, descompunerea în elemente și înțelegerea abstractizării. Recunoașterea modelelor este procesul de identificare, definire, extindere și creare a modelelor, algoritmi reprezintă o serie de pași pentru a rezolva o problemă, descompunerea înseamnă procesul de divizare a ceva (?) în elemente de bază, iar abstractizarea implică generalizări, tragerea de concluzii și alte procese de gândire pentru a imagina ceva ce nu poate fi văzut sau atins. Deși importanța gândirii computaționale este din ce în ce mai recunoscută, limitele sale conceptuale nu sunt încă clare. Au fost create multe definiții. Gândirea computațională a suferit o extindere semnificativă de conținut în comparație cu concepția din anii 1950, deoarece, pornind de la rădăcinile sale în informatică, acum reprezintă un concept mai cuprinzător, concentrându-se pe procesele cognitive și metacognitive care au loc în timpul prelucrării informațiilor de către individ. Așadar, gândirea computațională este acea capacitate care utilizează concepte informatice pentru formularea și rezolvarea

problemelor.

Pe baza unei revizuirii a literaturii de specialitate, Chaparro (2020) a comparat conceptele de gândire computațională și gândire algoritmică. Potrivit acestuia, diferența dintre cele două concepte constă în faptul că gândirea algoritmică se concentrează pe algoritmi, în timp ce gândirea computațională se concentrează pe metodele computaționale. El rezumă rezultatele cercetărilor anterioare și include printre conceptele cheie ale gândirii computaționale proiectarea algoritmică, care reprezintă o serie ordonată de operațiuni elementare create pentru a rezolva probleme similare.

2.2.1. Evaluarea gândirii computaționale la grădiniță

Deși interesul pentru studiul gândirii computaționale a crescut, încă nu există suficiente cercetări cu privire la modul de predare și evaluare a acestui domeniu de gândire, în special în copilăria timpurie (Rich și colab., 2018). Studiul gândirii computaționale ar oferi un feedback util educatorilor, cursanților și cercetătorilor care evaluează eficacitatea programelor educaționale, a planurilor de învățământ sau a intervențiilor. În ultimele două decenii, au fost dezvoltate o serie de instrumente de evaluare pentru a măsura gândirea computațională, dar puține s-au concentrat asupra copiilor mici (4-9 ani). Majoritatea lucrărilor anterioare au folosit interviul în sondaj sau au folosit evaluarea codării pe baza unui proiect. Deși interviurile și evaluările bazate pe proiecte oferă o perspectivă asupra gândirii copiilor, formatul și timpul necesar pentru aceste evaluări le fac inadecvate pentru utilizarea în afara mediului de cercetare.

Conform lui Lee și colab. (2011), „deoarece gândirea computațională nu este măsurată cu teste standardizate, în mediul educațional actual este dificil pentru profesori să predea conceptele de gândire computațională... domeniul necesită proceduri sistematice de evaluare/măsurare”. (p. 36). Acest lucru reprezintă însă o provocare majoră pentru cercetători, mai ales că abilitățile cognitive ale copiilor preșcolari sunt încă limitate. Desigur, au fost dezvoltate numeroase instrumente de măsurare în ultimele două decenii, dar doar câteva dintre acestea s-au concentrat pe copiii mici, adică pe grupa de vârstă 3-7 ani (El-Hamamsy, 2022, Marinus și colab., 2018, Relkin, 2018, Relkin și colab., 2020, Relkin și Bers, 2021, Zapata-Cáceres, 2020, Zhang și Wong, 2023).

În cele ce urmează, vor fi prezentate câteva instrumente de evaluare concepute pentru măsurarea gândirii computaționale la copiii mici. Unele dintre instrumentele de evaluare elaborate includ sarcini care implică roboți educaționali (Marinus, 2018, Relkin, 2018).

Marinus și colab. (2018) și-au propus să dezvolte un instrument de evaluare standardizat pentru a evalua abilitățile de codare ale copiilor cu vârste cuprinse între 3 și 6 ani. Testul a fost dezvoltat pentru robotul Cubetto. Cubetto constituie o variantă simplificată a activității de programare LOGO, elaborată de Seymour Papert. Testarea a fost efectuată pe 18 copii. Au fost introduse mai multe modificări și simplificări la nivelul sarcinilor, comparativ cu ceea ce poate face inițial robotul Cubetto.

Relkin și colab. (2018) au dezvoltat un alt instrument de măsurare pentru copiii cu vârsta cuprinsă între 5 și 7 ani, care a utilizat și un robot educațional pe podea. Testul a fost numit TACTIC-KIBO (Tufts Assessment of Computational Thinking in Children). Platforma KIBO este utilizată la nivel mondial pentru predarea abilităților de programare la copiii mici. În elaborarea testului, s-au bazat pe cele 7 idei esențiale/puternice/definitorii formulate de Bers (2018) privind gândirea computațională (7 powerful ideas), care sunt următoarele: algoritmi, modularitate (structuri repetitive), structuri de control, reprezentări, hardware/software, proces de proiectare, identificarea erorilor/(depanare). Celălalt punct de plecare este modelul de dezvoltare a programării creat de Vizner (2017), în care au fost distinse patru niveluri în dezvoltarea programării la copii: nivelul proto-programării, nivelul de programare incipient, nivelul de programare și programarea avansată/fluentă. Pornind de la aceste două teorii, au fost create sarcinile/întrebările TACTIC-KIBO, în care pentru fiecare dintre cele patru niveluri au fost elaborate întrebări/sarcini pentru fiecare concept. Pe baza rezultatelor, TACTIC-KIBO s-a dovedit a fi un instrument de măsurare promițător pentru evaluarea gândirii computaționale la copiii mici, prin care pot fi determinate cele patru niveluri de competență în programare.

Fiecare dintre cele trei instrumente de măsurare menționate presupune cunoștințe prealabile de codare, adică rezultatele sunt influențate de cunoștințele anterioare ale copiilor. Pe lângă acestea, alți cercetători au încercat să elaboreze instrumente care nu necesită cunoștințe prealabile de programare și care nu necesită utilizarea niciunui instrument tehnologic/digital. Instrumentele prezentate mai jos sunt de tip alegere multiplă, adică copilul trebuie să aleagă răspunsul corect dintre cele oferite. Un astfel de instrument de evaluare este cel al Cărților de joc pentru gândirea computațională Bebras (Bebras Unplugged Computational Thinking Cards), dezvoltat de comunitatea internațională a educatorilor Bebras Challenge (www.bebbras.org), cu scopul de a populariza informatica și predarea gândirii computaționale în școli. Testul a fost conceput pentru copii cu vârste cuprinse între 3 și 10 ani, conținând sarcini diferențiate pe grupe de vârstă, cu niveluri de complexitate variate. Fiecare carte de joc evaluează un concept fundamental al gândirii

compuționale: modele, algoritmi, logică și abstractizare. Cele 48 de cărți de joc sunt clasificate în trei niveluri de dificultate: ușor (16 cărți), mediu (17 cărți) și dificil (15 cărți). Sung și colaboratorii săi (2022) au validat aceste cărți de joc printr-un studiu realizat pe un eșantion de copii coreeni. Rezultatele obținute indică faptul că aceste cărți de joc Bebras posedă proprietăți psihometrice satisfăcătoare și sunt adecvate pentru evaluarea diferitelor abilități de gândire computațională la copiii mici.

Testul de gândire computațională TechCheck (TechCheck Computational Thinking (CT) assessment), dezvoltat de Relkin și colaboratorii săi în 2020, evaluează competențe similare cu cele măsurate de testul Bebras. TechCheck se axează pe conceptele de gândire computațională descrise de Bers (2018): algoritmi, modularitate (structuri repetitive), structuri de control, reprezentări, hardware/software, procesul de proiectare și identificarea erorilor. Dintre aceste șapte domenii, doar procesul de proiectare nu este evaluat de test. Instrumentul de evaluare online conține 15 întrebări cu răspuns multiplu, fiecare având patru variante de răspuns. Fiecare răspuns corect aduce un punct (punctajul maxim fiind de 15 puncte). Testul a fost făcut pe eșantion de 768 de copii cu vârste cuprinse între 5 și 9 ani, aplicând instrumentul de măsurare în grupuri. Pentru validare, rezultatele au fost comparate cu cele ale testului TACTIC-KIBO. În ansamblu, TechCheck a demonstrat proprietăți psihometrice satisfăcătoare, dovedindu-se a fi un instrument valid și fiabil pentru evaluarea gândirii computaționale la copiii cu vârste cuprinse între 5 și 9 ani. Cercetătorii au dezvoltat o versiune adaptată pentru copii mai mici, TechCheck-K (Relkin și Bers, 2021), destinată copiilor de vârstă preșcolară. Principala diferență constă în reducerea numărului de opțiuni de răspuns de la patru la trei pentru fiecare întrebare, dintre care copiii trebuie să aleagă una. Această modificare a fost justificată de studii anterioare care au arătat că memoria de lucru a copiilor preșcolari (5 ani) este limitată (pot reține simultan doar 3 elemente) (Simmering, 2012), ceea ce poate afecta performanța lor în sarcinile cu alegere multiplă (cu mai mult de 3 opțiuni de răspuns). Testul a fost online, copiii trebuind să selecteze răspunsul corect dintre cele trei opțiuni. Conducătorul experimentului a citit întrebarea de două ori la rând pentru copii. La studiu au participat 89 de preșcolari cu vârste între 5 și 6 ani. Conform cercetătorilor, un beneficiu al testului este că nu necesită cunoștințe anterioare de programare pentru efectuarea sarcinilor, dar aceștia au menționat și anumite limitări: datorită caracterului de tip alegere multiplă, se exclude exprimarea creativă și rezolvarea problemelor deschise, care sunt componente importante și semnificative ale gândirii computaționale.

Zapata-Cáceres și colaboratorii săi (2020) au dezvoltat Testul de Gândire Computațională

pentru Începători (Beginners Computational Thinking Test - BCTt), destinat copiilor cu vârste cuprinse între 5 și 12 ani. BCTt evaluează următoarele concepte ale gândirii computaționale: secvențe, structuri repetitive (simple și multiple), structuri condiționale (dacă-atunci, dacă-atunci-altfel, până când). Testul a fost conceput pentru copii preșcolari și școlari mici care încă nu știu să scrie sau să citească, motiv pentru care utilizează simboluri și imagini intuitive. Au fost concepute două tipuri de sarcini: sarcini de urmărire a liniilor (în care copilul trebuie să reproducă un model) și sarcini de tip labirint sau matrice. Testul, compus din 25 de întrebări, solicita alegerea răspunsului corect dintr-un set de patru opțiuni, fiecare opțiune reprezentând o secvență de pași ilustrată prin săgeți, simboluri și numere. Prima versiune a testului a fost trimisă spre evaluare unui grup de 45 de experți, care au evaluat itemii pe baza a 66 de întrebări. Pe baza sugestiilor lor, a fost creată a doua variantă și a fost testată pe 299 de copii. Rezultatele obținute indică faptul că nivelul de dificultate al testului este cel mai potrivit pentru elevii din învățământul primar (cu vârste cuprinse între 5 și 10 ani).

Zang și Wong (2023) au dezvoltat un test similar cu cel anterior, numit Testul de Gândire Computațională pentru Învățământul Primar Inferior (Computational Thinking Test for Lower Primary - CTtLP). Conceptele de gândire computațională abordate în test au fost: secvențe, direcții, structuri repetitive și structuri condiționale. Similar cu testul anterior, au creat două tipuri de sarcini: sarcini de desen și ghidarea unui personaj pe o rețea pătrată. Exercițiul de ghidare pe rețeaua din pătrate a fost adaptat din testul lui Román-Gonzáles (2015), iar rețeaua formată din pătrate a fost preluată din testul realizat de Zapata-Cáceres și colab. (2020). Au fost dezvoltați în total 30 de itemi, conform a trei scenarii: desen, Pac-Man și Șarpele Flămând. Pentru validare, testul a fost evaluat de 20 de experți prin intermediul unui chestionar online, iar itemii au fost analizați în urma unor interviuri cu 6 elevi, în același timp realizându-se un test pilot cu 72 de elevi din clasele primare. Pe baza feedback-ului oferit de experți și elevi, au fost efectuate mai multe modificări, inclusiv adăugarea unei mici săgeți pentru a indica direcția de pornire și înlocuirea termenului „90 de grade” cu o reprezentare vizuală, deoarece acest concept este necunoscut pentru elevii de clasa a doua. Rezultatele obținute indică faptul că testul este adecvat pentru a măsura abilitățile vizate și, în general, este potrivit pentru grupa de vârstă țintă.

2.2.2. Dezvoltarea gândirii computaționale în grădiniță

Wing (2017) subliniază faptul că conceptul introdus de ea a influențat semnificativ numeroase domenii științifice în cei peste 10 ani de la apariție. Dezvoltarea acestei

competențe câștigă tot mai mult teren și în domeniul educației. Predarea gândirii computaționale s-a răspândit la nivel internațional, fiind inclusă în curricula multor țări încă din învățământul primar. Ministerul Educației din Regatul Unit a modificat curriculumul național, integrând învățarea gândirii computaționale pentru toți elevii începând cu anul 2014 (UK Department for Education, 2013). Comitetul Danez pentru Dezvoltare a prezentat într-un raport de 120 de pagini importanța integrării învățării gândirii computaționale la toate nivelurile de educație și a recomandat această abordare guvernului danez (The Danish Growth Council, 2016).

Kazakoff, Sullivan și Bers (2012) au investigat evoluția capacității de ordonare și secvențializare a preșcolarilor după ce aceștia au participat la un program intensiv de programare a roboților. Ipoteza lor principală a fost că, dintre procesele implicate în programarea roboților, ordonarea este cea componentă care este prezentă și atunci când copiii pun evenimentele unei povești în ordine logică. Deosebirea dintre rezultatele pre- și post-test au arătat diferențe semnificative în sarcinile de ordonare a imaginilor pentru grupul experimental. Pe baza acestor rezultate, au concluzionat că la copiii implicați în programarea roboților s-a îmbunătățit capacitatea de ordonare.

În cercetarea lor, Bers și colab. (2014) au aplicat Programul de Robotică „TangibleK” în trei grupe de grădiniță, program a cărui dezvoltare a avut la bază constructivismul și cadrul Dezvoltării Tehnologice Pozitive. Curriculumul „TangibleK” promovează dezvoltarea gândirii computaționale. Programul vizează în special următoarele competențe ale gândirii computaționale: reprezentarea problemelor, consecvența în generarea și aplicarea soluțiilor, explorarea mai multor soluții posibile, rezolvarea de probleme la diferite niveluri, o atitudine creativă în fața eșecurilor și identificarea erorilor/concepțiilor greșite pe parcursul unui proiect de succes, precum și aplicarea de strategii pentru abordarea problemelor complexe. Rezultatele studiului indică faptul că educatoarele pot implementa cu succes acest curriculum și că preșcolarii sunt interesați și capabili să învețe și să aplice diverse aspecte ale roboticii, programării și gândirii computaționale.

Lindenberg și colab. (2019) au evidențiat o tendință globală în creștere în domeniul educației, pentru predarea gândirii computaționale, ceea ce subliniază importanța introducerii educației informatice încă din perioada preșcolară. Având în vedere capacitățile cognitive limitate ale copiilor preșcolari, s-a impus necesitatea de a elabora o logică de programare adaptată și un curriculum adecvat. Având în vedere acest aspect, Ching și colab. (2018) și-au propus ca obiectiv în cercetarea lor elaborarea unui cadru conceptual care să sprijine dezvoltarea abilităților informatice ale copiilor preșcolari, să

sporească interesul pentru învățare și să îmbunătățească rezultatele școlare. S-a integrat o metodă de învățare bazată pe joc cu o interfață tangibilă - o interfață care permite interacțiunea cu informații digitale prin intermediul unor obiecte fizice (de exemplu, un mouse) - (TUI - tangible user interface) pentru a dezvolta gândirea computațională la preșcolari. Performanțele de învățare au fost evaluate prin intermediul unor întrebări de diferite niveluri de dificultate, iar analiza rezultatelor a condus la concluzia că metoda de învățare bazată pe joc, în combinație cu o interfață tangibilă, îmbunătățește semnificativ performanțele de învățare ale preșcolarilor și le crește abilitățile de gândire computațională.

2.3. Roboți educaționali în grădiniță

2.3.1. Roboți educaționali

De când Seymour Papert a dezvoltat limbajul de programare LOGO în 1967, roboții educaționali au fost integrați ca instrumente didactice în procesul educațional și reprezintă în continuare subiectul a numeroase cercetări. În ultimele două decenii, frecvența utilizării roboților a crescut considerabil. Astăzi, există numeroși roboți concepuți special pentru copii mici. Unii dintre acești roboți, precum Bee-bot/Blue-bot sau șoricelul robot Colby, pot fi programați prin intermediul butoanelor situate pe spatele lor. Aceste seturi de roboți includ și carduri de comandă (pentru a merge înainte, înapoi, a se întoarce la stânga, a se întoarce la dreapta), permițând copilului să creeze o secvență de comenzi (algoritm - cod) înainte de a programa robotul. Acești roboți includ, de asemenea, tablouri cu rețea pătrată: robotul avansează sau se retrage cu un pătrat la fiecare comandă .

Un alt tip de robot, KIBO, este programabil prin asamblarea/(conectarea) unor cuburi/blocuri din lemn (cu ajutorul găurilor și mânerelor) (Bers, 2018). Fiecare bloc reprezintă o comandă (a merge înainte, a merge înapoi, a se întoarce la stânga, a se întoarce la dreapta, a se roti, a aprinde luminile, a emite un semnal sonor, a aștepta aplauze etc.). Acest robot poate executa mai multe comenzi decât roboții cu butoane prezentați anterior. În plus, el dispune de structuri precum repetiția și condiția "dacă". Cercetările au arătat că preșcolarii sunt capabili să înțeleagă și să utilizeze atât repetiția, cât și parametrii numerici (Bers și colab., 2014), care se pot programa cu robotul KIBO.

Un al treilea tip de robot fără ecran este Cubetto, care vine cu o tablă și blocuri de comandă ce trebuie plasate pe aceasta. Robotul nu are butoane pe spate și poate fi programat doar cu ajutorul tablei. O caracteristică importantă a robotului este bara de funcții, care permite

utilizarea subprogramelor (Gadzikowski, 2018): pe tablă există o zonă specială cu patru spații unde pot fi plasate cel mult patru blocuri, formând astfel un subprogram. Astfel, atunci când se programează Cubetto, în bara de comenzi trebuie plasat doar blocul corespunzător subprogramului. Bara de funcții îi ajută pe copii să exerseze abstractizarea și modularizarea (Yu și Roque, 2019).

2.3.2. Implementarea roboților educaționali în grădiniță

Cea mai recentă generație de kituri robotice destinate copiilor mici permite învățarea manipulativă, și în același timp, dat fiind faptul că activitățile de robotică nu sunt de obicei legate de ecrane, promovează munca în echipă și cooperarea (Sullivan și Bers, 2016). Cercetările au arătat că copiii cu vârste cuprinse între 4 și 6 ani sunt capabili să proiecteze și să construiască roboți simpli (Cejka și colab., 2006), dobândind în acest proces cunoștințe de inginerie, tehnologie și programare, în timp ce își dezvoltă abilitățile de gândire computațională. Activitățile de robotică îmbunătățesc motricitatea fină a copiilor, coordonarea mână-ochi, promovând în același timp cooperarea și munca în echipă. În plus, copiii experimentează concepte ingineresti și își dezvoltă abilitățile narative prin crearea de povești și scenarii pentru proiectele lor (Bers, 2008).

În cercetarea sa din Suedia, Palmér (2017) a investigat utilizarea conceptelor computaționale în activitățile de rezolvare a problemelor, concentrându-se în mod special pe relația dintre gândirea spațială, una dintre abilitățile matematice, și programare la preșcolari. Copiilor cu vârste cuprinse între 3 și 5 ani li s-au introdus activități de programare în rutina zilnică a grădiniței, pe o perioadă de 4 luni, cu scopul de a observa evoluția gândirii lor spațiale. Gândirea spațială implică compararea mentală, rotația și memorarea relațiilor, formarea conceptelor și transformarea obiectelor. În cadrul cercetării, copiii au lucrat cu robotul Bee-Bot. Rezultatele au arătat că copiii au participat cu entuziasm la activități și toți au reușit să rezolve sarcinile. De asemenea, copiii au înțeles semnificația simbolurilor (săgeți în diferite direcții), utilizându-le corect și demonstrând prin mișcări de mână și corp cum se va mișca robotul conform săgeților.

Otterborn și colab. (2019) au investigat, tot în Suedia, opiniile educatorilor de grădiniță cu privire la conținutul, obiectivele și metodele activităților de programare, pe baza propriei practici didactice. Rezultatele au evidențiat două abordări principale în ceea ce privește integrarea programării în grădiniță: programarea deconectată - „unplugged programming”, adică fără dispozitive digitale, și programarea digitală. Un mod posibil de programare fără

utilizarea dispozitivelor digitale este ca un copil să joace rolul de robot, iar alt copil să dea diferite comenzi pentru a-l mișca și a-l ghida pe „robot”. Programarea digitală a fost clasificată în două tipuri: programarea directă prin intermediul diferitelor aplicații sau programe (pe computer sau tabletă) și programarea cu ajutorul unor instrumente tangibile și manipulative, cum ar fi robotul Blue-Bot și alți roboți. Rezultatele au arătat că educatorii de grădiniță au integrat programarea în diverse proiecte și teme, promovând colaborarea, rezolvarea problemelor și încrederea copiilor în propriile abilități.

Bers și colab. (2019) au realizat un studiu cu copii de grădiniță (3-5 ani), în care au investigat gândirea computațională folosind setul de roboți KIBO. Rezultatele au arătat că, prin utilizarea roboticii, copiii au dobândit un nivel înalt de cunoștințe de codare și abilități de gândire computațională.

Çakır și colab. (2021) au studiat impactul activităților de programare și robotică asupra abilităților de rezolvare a problemelor și a creativității la preșcolari. Pentru evaluarea abilităților de rezolvare a problemelor, s-a folosit sarcinile din Scala de Abilități de Rezolvare a Problemelor PSSS (Problem-Solving Skill Scale), iar pentru evaluarea creativității, testul de Creativitate Integrativă (Integrative Creativity test). În timpul intervenției, grupul experimental a participat la activități de programare și robotică, în timp ce grupul de control a primit sarcini pe bază de hârtie și creion. Rezultatele au arătat că grupul experimental a obținut performanțe semnificativ mai bune în post-test atât în ceea ce privește rezolvarea problemelor, cât și creativitatea.

Bakala și colaboratorii (2021) au efectuat o revizuire sistematică a studiilor publicate care au investigat impactul activităților de robotică și programare asupra gândirii computaționale la preșcolari. Cercetătorii au căutat răspunsuri la patru întrebări principale: ce tipuri de roboți au fost utilizați și cum pot fi aceștia categorisiți, care sunt caracteristicile activităților destinate dezvoltării gândirii computaționale, cum și cu ce instrumente a fost măsurată gândirea computațională, în ce țări și de către care cercetători s-au efectuat cele mai multe studii în acest domeniu și care sunt motivațiile acestora. În urma analizei, au rămas 15 articole relevante pentru acest subiect. Revizuirea a evidențiat următoarele aspecte: toate studiile au utilizat roboți comerciali, nu există un consens privind modul de implementare a acestor activități – structura, durata și organizarea fiind foarte diferite – și s-au folosit metode de evaluare extrem de eterogene. Rezultatele sugerează necesitatea unor raportări de cercetare mai riguroase în acest domeniu.

3. OPINII ȘI EXPERIENȚE ALE EDUCATORILOR DE GRĂDINIȚĂ ÎN FORMARE ȘI ACTIVITATE, CU PRIVIRE LA IMPLEMENTAREA ACTIVITĂȚILOR STEM ȘI DEZVOLTAREA GÂNDIRII ALGORITMICE/COMPUTAȚIONALE

3.1. Opiniile și experiențele studenților cu specializarea în Pedagogie și Didactică Aplicată cu privire la aplicarea activităților STEM și dezvoltarea gândirii algoritmice/computaționale în grădiniță

Metodologie

Acest studiu a fost realizat în primul semestru al anului universitar 2021-2022, în rândul studenților de la secția de Pedagogie și Didactică Aplicată. Studiul a avut mai multe obiective: în primul rând, a vizat explorarea opiniilor și experiențelor studenților cu privire la implementarea activităților STEM în grădiniță; în al doilea rând, s-a urmărit investigarea percepțiilor studenților asupra utilizării componentei tehnologice în activitățile STEM. În plus, ne-am propus să explorăm opiniile studenților despre algoritmi și importanța dezvoltării gândirii algoritmice și computaționale la preșcolari.

Întrebările cercetării

1. Ce domenii de experiență și activități relevante pot fi integrate în planificarea activităților educaționale pentru grădiniță?
2. Sunt familiarizați cu termenul STEM și semnificațiile acestuia?
3. Care sunt aspectele esențiale ale activităților STEM, ce oportunități oferă și ce abilități/capacități cognitive, sociale, fizice și emoționale dezvoltă?
4. Ce experiențe personale au acumulat în aplicarea activităților STEM în practica lor pedagogică?
5. Cum poate fi stimulată aplicarea mai frecventă a activităților STEM în grădinițe, conform opiniei lor?
6. Care este opinia lor și ce experiențe au cu privire la activitățile tehnologice ca parte a STEM?
7. Ce cunoștințe/experiențe anterioare au despre algoritmi?
8. Ce cunoștințe și experiențe au cu dezvoltarea gândirii computaționale în grădiniță?

Participanți

La studiu au participat 85 de studenți ai Facultății de Psihologie și Științe ale Educației, specializarea Pedagogie și Didactică Aplicată a Universității Babeș-Bolyai: 51 de studenți în anul II și 34 de studenți în anul III. În ceea ce privește distribuția pe sexe, dintre respondenți doar 1 (1,2%) a fost de sex masculin, ceea ce poate fi explicat prin faptul că, în general, femeile sunt mai interesate de această specializare și se înscriu în număr mai mare.

Instrumentar

Pentru realizarea acestui studiu, am utilizat un chestionar online, creat în Google Forms, care conținea 36 de întrebări. Dintre acestea, 5 întrebări se refereau la date demografice, 3 întrebări vizau modalitățile de integrare a activităților în grădiniță, 10 întrebări au explorat cunoștințele și experiențele respondenților legate de activitățile STEM, 7 întrebări s-au axat pe integrarea cunoștințelor tehnologice – ca parte a STEM – în grădiniță, iar 11 întrebări au abordat importanța dezvoltării gândirii computaționale la preșcolari. Chestionarul a inclus atât întrebări deschise, cât și întrebări închise (întrebări cu răspunsuri multiple și afirmații măsurate pe o scală Likert de 5 puncte). Întrebările referitoare la activitățile STEM au fost preluate dintr-un alt chestionar folosit într-un proiect ERASMUS+, coordonat de Universitatea din Catalonia, „Kitchen Lab for Kids” (K4K, 2020).

Rezultate

1. Întrebare: Ce domenii de experiență și activitățile aferente pot fi integrate cu succes în planificarea activităților din grădiniță?

95,3% dintre respondenți integrează activitățile aferente domeniilor de experiență. Acest rezultat confirmă faptul că în planificare se realizează principiul integrării, așa cum recomandă și Curriculumul pentru educația timpurie (Ministerul Educației Naționale, 2019). A doua întrebare se referea la modalitatea de integrare: participanții trebuiau să bifeze ce tipuri de activități pot fi integrate în mod optim. Rezultatele arată că atât studenții în anul II, cât și cei în anul III au asociat cel mai frecvent activitățile de cunoașterea mediului cu matematica (40 de răspunsuri) și limba maternă (17 răspunsuri), iar matematica a fost, la rândul ei, asociată cel mai frecvent cu activitățile de cunoașterea mediului (38 de răspunsuri). În cazul matematicii, activitățile alese pentru a fi integrate alături de activitățile de cunoașterea mediului au diferit între cei doi ani de studiu: studenții din anul II au preferat integrarea cu educația artistică (8 persoane) și lucrul manual (6

persoane), în timp ce studenții din anul III au optat pentru limba maternă (4 persoane) și educația fizică (4 persoane). Activitățile de limbă maternă sunt asociate de ambele grupe cu educația muzicală (25 persoane), cunoașterea mediului (14 persoane) și educația artistică (16 persoane), în proporții diferite. Limba română (care este limba oficială a statului, dar este predată ca limbă străină) este cel mai frecvent asociată cu educația muzicală (28 persoane) de studenții din ambele grupe, precum și cu educația artistică (17 persoane). Pentru a integra activitățile muzicale, studenții din anul II au preferat mai degrabă educația fizică (16 persoane), în timp ce cei din anul III au optat pentru educația artistică (12 persoane), deși educația fizică a fost a doua cea mai frecventă alegere (9 persoane). Educația artistică a fost asociată de ambele grupuri cu aceleași activități, doar în proporții diferite: în timp ce studenții din anul II au preferat să o conecteze cu cunoașterea mediului (16 persoane), studenții din anul III au ales mai des limba maternă (15 persoane). Activitățile de morală/gospodărie au fost cel mai frecvent integrate cu limba maternă (39 persoane) de ambele grupe. Cea mai mare diferență a fost observată în integrarea activităților de lucru manual: cele două grupuri alegând activități complet diferite: studenții din anul II au preferat să le conecteze cu cunoașterea mediului (12 persoane) și matematica (9 persoane), în timp ce studenții din anul III au ales limba maternă (11 persoane) și muzica (5 persoane). Activitățile de educație fizică au fost asociate cel mai frecvent cu muzica (37 persoane) de ambele grupuri.

2. Întrebare: Cunosc respondenții termenul STEM și semnificația acestuia?

67,4% dintre respondenți nu sunt conștienți de semnificația termenului STEM: 64,7% dintre studenții din anul II și 73,52% dintre studenții din anul III nu cunosc acronimul STEM. Este surprinzător faptul că rezultatul este mai slab în cazul studenților din anul III, acest lucru sugerând că disciplinele studiate pe parcursul studiilor universitare nu i-au ajutat să se familiarizeze mai aprofundat cu educația STEM. Comparând acest rezultat cu rezultatele din alte țări, se pare că nivelul de cunoștințe STEM al studenților la pedagogia învățământului preșcolar și al educatorilor preșcolari practicieni variază de la o țară la alta și, mai mult, depinde și de instituția de învățământ superior în care studiază. De exemplu, Karademir și Yıldırım (2021) în studiul lor au arătat că studenții în ultimul an la pedagogia învățământului preșcolar au putut defini ce înseamnă educația STEM, în contrast cu rezultatele cercetării lui Baltsavias și Kyridis (2020), conform cărora educatorii de grădiniță în activitate nu sunt la fel de familiarizați cu educația STEM, mai mult, în cercetarea experimentală a Aleksievei și a colaboratorilor săi (2021), educatorii de

grădiniță participanți nu cunoșteau deloc termenul STEM și semnificația acestuia înainte de intervenție. Aceste rezultate sugerează că familiarizarea pedagogilor cu educația STEM depinde de curricula instituțiilor de formare.

După ce am definit și am oferit o scurtă descriere a ceea ce înseamnă educația STEM, le-am cerut studenților să decidă dacă anumite afirmații sunt adevărate sau false. Prima afirmație falsă se referea la faptul că activitățile STEM sunt incluse în Curriculumul românesc în vigoare. 7,84% (4 persoane) dintre respondenții din anul II au considerat această afirmație ca fiind adevărată, niciunul dintre studenții din anul III nu a marcat-o ca fiind adevărată. Celălalt răspuns fals s-a referit la faptul că activitățile STEM pot fi aplicate cu succes doar în școală. În legătură cu aceasta, 11,76% (6 persoane) dintre studenții din anul II au considerat afirmația ca fiind adevărată, 5,88% (2 persoane) dintre studenții din anul III au considerat-o adevărată. A treia afirmație falsă s-a referit la faptul că activitățile STEM necesită în toate cazurile instrumente digitale. 18,82% (16 persoane) dintre respondenți au considerat această afirmație falsă ca fiind adevărată: dintre studenții din anul II, 21,56% (11 persoane) au considerat-o adevărată, iar dintre studenții din anul III, 14,7% (5 persoane) au considerat afirmația ca fiind adevărată. Un rezultat interesant este că, în ciuda faptului că studenții din anul III cunoșteau mai puțin educația STEM, după o scurtă explicație au înțeles mai bine ce este STEM. Considerăm posibil ca aceștia să fi avut deja anumite cunoștințe despre educația STEM, doar că nu s-au întâlnit încă cu acronimul STEM în sine.

3. Întrebare: Care este esența activităților STEM, ce oportunități oferă și ce abilități/competențe cognitive, sociale, fizice și emoționale dezvoltă?

Pentru a cartografia opiniile studenților despre esența educației STEM în învățământul preșcolar și oportunitățile pe care le oferă aceasta, am formulat două serii de afirmații, iar rezultatele au fost măsurate cu o scală Likert cu 5 puncte. Pentru compararea răspunsurilor, am aplicat un test T pentru două eșantioane pentru fiecare afirmație (Tabelul 1). Deși în cazul anumitor afirmații au existat mici diferențe între medii, nu am găsit o diferență statistic semnificativă în cazul niciunei afirmații. În ceea ce privește esența educației STEM, majoritatea studenților au fost de acord cu afirmația că aceasta stimulează gândirea creativă a copiilor în domeniile științelor naturii. Și alte cercetări au evidențiat gândirea creativă ca fiind cel mai important beneficiu al educației STEM (Karademir și Yıldırım, 2021). În ceea ce privește cele mai importante oportunități pe care le oferă educația STEM, majoritatea studenților au fost de acord cu trei afirmații: activitățile STEM construiesc

cunoștințele copiilor despre lumea socială, naturală și tehnică; aceștia se îmbogățesc cu experiențe practice; precum și oferă emoții pe învățarea științelor.

Tabelul 1. Compararea răspunsurilor studenților din anul II și anul III pe baza afirmațiilor referitoare la esența și oportunitățile activităților STEM în învățământul preșcolar

Afirmație	Anul II		Anul III		p	t
	Átlag	Szórás	Átlag	Szórás		
Esența educației STEM						
1. Stimularea învățării copiilor prin experiențe directe și personale	4,24	0,66	4,26	1,11	,890	0,138
2. Stimularea gândirii creative a copiilor în domeniile științifice	4,55	0,49	4,32	0,77	,215	-1,254
3. Crearea unui proces activ de predare-învățare	4,16	0,69	4,24	0,91	,698	0,390
4. Identificarea și rezolvarea problemelor în situații naturale, cotidiene	4,12	0,83	3,94	1,33	,456	-0,750
5. Formarea unei perspective integrate, holistice asupra lumii în mintea copilului	3,94	0,74	3,91	1,05	,891	-0,138
6. Sprijinirea dezvoltării holistice a copilului	4,02	0,62	3,85	0,86	,392	-0,862
7. Dezvoltarea procesului de predare-învățare prin includerea a cel puțin două domenii STEM	4,16	0,77	4,00	1,09	,473	-0,721
Oportunități în educația STEM						
1. Formarea unei imagini de sine pozitive	3,61	0,92	3,59	1,16	,932	-0,086
2. Dezvoltarea emoțiilor pozitive și a motivației pentru învățarea științelor	4,53	0,41	4,38	0,55	,348	-0,946
3. Construirea cunoștințelor copiilor despre lumea socială, naturală și tehnologică	4,63	0,32	4,38	0,55	,106	-1,641

4. Învățare auto-dirijată și independentă	4,04	0,80	4,18	0,82	,493	0,690
5. Învățare colaborativă	3,90	0,93	4,03	0,70	,519	0,648
6. Dobândirea de experiență practică	4,59	0,33	4,47	0,68	,472	-0,723
7. Stimularea învățării prin joc	4,47	0,61	4,29	0,88	,368	-0,906
8. Formularea de întrebări și căutarea de răspunsuri prin experimentare	4,41	0,73	4,35	0,72	,756	-0,312

Următoarele patru întrebări au vizat explorarea, din perspectiva studenților, a abilităților cognitive, emoționale, sociale și fizice pe care le dezvoltă activitățile STEM. În ceea ce privește abilitățile cognitive, majoritatea, adică 75,6% (56), consideră că gândirea exploratorie și creativă poate fi cel mai bine dezvoltată prin aplicarea activităților STEM în perioada timpurie a copilăriei. Acest rezultat este în concordanță cu rezultatele țărilor participante la proiectul Erasmus+ K4K (K4K, 2020). De asemenea, viitorii educatori de grădiniță din Turcia consideră că activitățile STEM contribuie la dezvoltarea creativității copiilor (Karademir și Yıldırım, 2021; Ültey și Ültey, 2020). 81,4% (70 persoane) dintre respondenți au considerat că, dintre abilitățile/competențele sociale, munca în echipă este cel mai bine dezvoltată de activitățile STEM. Acest rezultat este, de asemenea, în concordanță cu constatările cercetărilor anterioare (K4K, 2020; Karademir și Yıldırım, 2021). Dintre abilitățile/competențele emoționale, 68,6% dintre respondenți consideră că activitățile STEM dezvoltă cel mai mult autonomia. Acest răspuns s-a situat pe locul al doilea în rezultatele cercetării pe baza căreia am elaborat și adaptat chestionarul nostru (K4K, 2020). Conform studenților, cele două abilități fizice care pot fi cel mai mult dezvoltate prin activitățile STEM sunt: experimentarea lumii prin simțuri (60,5%, 52 de răspunsuri) și abilitățile motorii fine și grosiere (59,3%, 51 de răspunsuri). Aceste rezultate sunt, de asemenea, în concordanță cu rezultatele obținute de echipa K4K (2020).

4. Întrebare: Ce experiențe personale au avut în aplicarea activităților STEM în cadrul practicii lor pedagogice?

60,5% (52 persoane) dintre respondenți au declarat că nu au nicio experiență în acest domeniu (Tabelul 2). Acest rezultat este îngrijorător, deoarece în proiectul K4K, majoritatea educatorilor și a studenților la pedagogie aveau deja experiență personală în

aplicarea activităților STEM. Cel mai frecvent domeniu în care studenții aveau experiență (menționat de 18 studenți), a fost conducerea observațiilor științifice și efectuarea de experimente.

Tabelul 2. Experiențele personale ale respondenților în activitățile STEM

Tipul de experiență	Anul II (frecvență)	Anul III (frecvență)	Total (frecvență)
1. Conducerea/ghidarea de observații științifice și experimente	11	7	18
2. Studiul caracteristicilor fizice ale lumii	8	3	11
3. Ateliere practice legate de informatică (de ex., funcționarea miniroboților)	5	0	5
4. Proiecte interdisciplinare, care au fost realizate prin conectarea a cel puțin două domenii STEM	1	3	4
5. Grădiniță în natură/practică pe teren	7	3	10
6. Excursii la centre științifice, laboratoare universitare	8	3	11
7. Nu am nicio experiență în aplicarea activităților STEM în educația timpurie	29	23	52

5. Întrebare: Conform opiniei lor, prin ce mijloace poate fi stimulată aplicarea mai frecventă a activităților STEM în grădiniță?

Respondenții au considerat că, pentru a crește frecvența activităților STEM în grădiniță, este necesară îmbogățirea cunoștințelor educatorilor despre conținuturile STEM, precum și extinderea cunoștințelor metodologice legate de aceste activități (64%, 56 de răspunsuri). Cercetările au demonstrat că programele de formare profesională privind educația STEM, precum și sprijinirea studenților în proiectarea și desfășurarea de activități STEM în grupele de grădiniță, sunt modalități eficiente de a crește încrederea studenților și a educatorilor în cunoștințele lor STEM și de a le schimba atitudinea față de activitățile STEM (Aleksieva și colab., 2021, Fridberg și colab., 2022). În plus, este necesară includerea disciplinelor științifice conexe educației STEM în formarea viitorilor educatori preșcolari. Chiar și introducerea predării unei singure discipline științifice legate de educația STEM poate avea un impact semnificativ asupra viitorilor educatori în tendința lor de a integra/aplica activități STEM în munca lor (Uğraş și Genç, 2018). Totodată, în

implementarea activităților STEM este nevoie și de sprijin profesional pe termen lung. Cele mai eficiente programe de formare profesională sunt cele care oferă practică sub îndrumarea unui mentor și oferă experiențe de predare STEM (Chen și colab., 2021).

11,6% dintre respondenți au considerat că modificarea motivației educatorilor față de activitățile STEM ar contribui, de asemenea, la îmbunătățirea situației. Cercetările arată că motivația pentru predarea STEM crește pe măsură ce crește încrederea în conținutul STEM și cunoștințele metodologice (Aleksieva și colab., 2021), deci asigurarea formărilor adecvate poate reprezenta soluția pentru aceasta. 9,3% dintre respondenți au considerat că dotarea grădinițelor cu instrumente STEM ar contribui, de asemenea, la creșterea frecvenței activităților STEM, deoarece, în multe cazuri, unul dintre principalele obstacole în introducerea activităților STEM în grădinițe este lipsa instrumentelor necesare (Ûltey și Ûltey, 2020).

6. Întrebare: Ce părere și ce experiență au, legat de activitățile tehnologice – ca parte a STEM?

Pentru a mapa opiniile respondenților cu privire la aplicarea cunoștințelor tehnologice în activitățile preșcolare, am formulat patru enunțuri/afirmații, acestea trebuind evaluate pe o scală Likert cu cinci puncte (unde 1 = nu sunt deloc de acord și 5 = sunt pe deplin de acord).

Studentii sunt de acord cu faptul că o atitudine pozitivă față de sarcinile tehnologice poate fi fundamentată în grădiniță. Cu toate acestea, nu sunt convinși de necesitatea predării cunoștințelor tehnologice în grădiniță. Și cu atât mai puțin sunt de acord cu afirmația că sarcinile tehnologice realizate în grădiniță servesc drept bază pentru programare (planificare, implementare). Acest lucru poate fi explicat prin faptul că nu au experiență în integrarea cunoștințelor tehnologice în activitățile preșcolare, deoarece 89,5% dintre ei (77 persoane) nu au încercat încă niciodată să dezvolte competențele tehnologice ale copiilor. În continuare, am solicitat respondenților, ca cei care au aplicat cunoștințe tehnologice să descrie (într-o întrebare deschisă) ce cunoștințe și cum le-au aplicat în timpul activităților. Analiza răspunsurilor a arătat că mulți au confundat educația tehnologică cu utilizarea tehnologiei, deoarece majoritatea au înțeles prin educație tehnologică utilizarea instrumentelor tehnologice. Această confuzie/eroare conceptuală poate explica de ce studenții nu sunt atât de deschiși în ceea ce privește integrarea cunoștințelor tehnologice în grădiniță. Din răspunsurile la următoarea întrebare, am dorit să identificăm ce i-a împiedicat să planifice activități bazate pe cunoștințe tehnologice. 40,3% dintre

respondenți (31 persoane) au afirmat că nu au pregătirea metodologică necesară pentru planificarea unor astfel de activități, 32,5% (25 persoane) au considerat că lipsa instrumentelor didactice adecvate a fost cel mai mare obstacol, iar 28,6% dintre ei (22 persoane) au identificat lipsa de încredere în sine ca fiind un impediment. 95,3% dintre respondenți (82 persoane) ar dori să afle mai multe despre activitățile care vizează introducerea cunoștințelor tehnologice în grădiniță. Acest rezultat este în concordanță cu constatările lui Zsoldos-Marchis și Ciascai (2019) și subliniază importanța integrării activităților tehnologice – ca parte a activităților STEM – în formarea viitorilor educatori preșcolari și învățători. Cunoștințele metodologice necesare pentru implementarea STEM pot fi asimilate eficient prin cursuri de formare și programe de mentorat (Uğraş și Genç, 2018, Chen și colab. 2021).

7. Întrebare: Ce cunoștințe/experiență anterioară au despre algoritmi?

În prima întrebare din această parte, am solicitat studenților să ofere exemple de activități care pot dezvolta gândirea algoritmică în grădiniță. 73,3% dintre respondenți (63 persoane) au formulat exemple care pot contribui cu adevărat la dezvoltarea gândirii algoritmice. Unele exemple au fost foarte generale, deoarece au enumerat doar domeniile experiențiale care utilizează algoritmi, cum ar fi matematica, științele naturii și artele. Alții au oferit exemple legate de rutinele zilnice din grădiniță, cum ar fi rutina de dimineață, pregătirea pentru masă etc. Pe lângă acestea, au existat și exemple corecte, concrete, cum ar fi utilizarea roboților, construirea cu blocuri (cuburi) de construcție (-educative) sau realizarea unor activități manuale prin urmarea pașilor indicați de profesor etc.

Pentru a evalua capacitatea studenților de a scrie algoritmi, le-am cerut să descrie pașii construirii unui om de zăpadă. Am considerat și am acceptat următoarea soluție ca fiind corectă: 1. pregătește trei bulgări de zăpadă de dimensiuni diferite, 2. așază bulgărele de zăpadă mijlociu deasupra celui mai mare, apoi pe cel mai mic deasupra celui mijlociu, 3. pune doi ochi în mijlocul celui mai mic, un nas și o pălărie deasupra. Desigur, acești trei pași principali au mulți sub-pași, pe care îi putem considera și subprograme. Există și pași care pot fi interschimbabili, deci ordinea lor nu este importantă, de exemplu, cineva poate pune mai întâi nasul, apoi ochii omului de zăpadă. Doar 32,94% dintre respondenți (28 persoane) au formulat corect pașii algoritmului de construire a omului de zăpadă. Majoritatea respondenților au greșit prin faptul că nu au descris suficient de detaliat ordinea pașilor.

În ceea ce privește experiența anterioară a studenților legată de algoritmi, 70,9% dintre ei

(61 persoane) au auzit de termenul „algoritm” doar la ora de matematică, 15,1% dintre ei (13 persoane) au studiat programare, iar 9,3% dintre ei (8 persoane) au afirmat că nu au studiat niciodată algoritmi. Așadar, studenții participanți aveau o experiență foarte limitată în domeniile algoritmilor și codării. Totuși, 93% dintre studenți (80 persoane) au considerat că este necesară dezvoltarea timpurie a gândirii algoritmice.

8. Întrebare: Ce cunoștințe și experiență au despre dezvoltarea gândirii computaționale în grădiniță?

Am solicitat respondenților să evalueze trei afirmații referitoare la tipuri de activități legate de dezvoltarea gândirii computaționale pe o scală Likert de 5 puncte, în funcție de cât de mult sunt de acord sau nu cu afirmația respectivă (1 – nu sunt deloc de acord, 5 – sunt pe deplin de acord). Mediile și abaterile standard ale afirmațiilor sunt rezumate în tabelul 3.

Tabelul 3. Opinii privind dezvoltarea gândirii computaționale în funcție de experiență

Afirmație	Medie	Abatere standard
Dezvoltarea gândirii computaționale necesită întotdeauna un calculator.	2,84	1,204
Gândirea computațională poate fi dezvoltată/învățată și fără a utiliza calculatorul.	3,57	1,189
Gândirea computațională poate fi dezvoltată și prin sarcini bazate pe hârtie și creion.	3,39	1,256

Pe baza răspunsurilor, majoritatea au fost de acord cu afirmația că „gândirea computațională poate fi dezvoltată și fără utilizarea unui calculator”. Acesta este un rezultat promițător, deoarece unul dintre principalele motive pentru care educatorii din grădinițe nu consideră adecvată dezvoltarea gândirii computaționale la această vârstă este acela că cel puțin grădinița ar trebui să contribuie la limitarea timpului petrecut în fața ecranelor. Yavadav și colaboratorii săi (2014) au descoperit în cercetarea lor că studenții de la specializarea pedagogia învățământului preșcolar care nu au beneficiat de formare în domeniul gândirii computaționale erau convinși că gândirea computațională necesită utilizarea tehnologiei computerelor. 67,4% (58 persoane) dintre respondenți au considerat importantă dezvoltarea gândirii computaționale în grădiniță. Ulterior, studenții au fost rugați să ofere exemple de activități care pot dezvolta gândirea computațională în grădiniță.

47,67% dintre ei (41 persoane) au oferit exemple bune pentru dezvoltarea gândirii computaționale. În opinia lor, gândirea computațională poate fi dezvoltată prin sarcini bazate pe algoritmi, sarcini bazate pe rezolvarea de probleme, jocuri bazate pe succesiunea diferiților pași. Doi studenți au menționat și roboții educaționali ca posibile instrumente de dezvoltare. Cu toate acestea, au existat și studenți care au oferit răspunsuri incorecte, de exemplu, au spus că gândirea computațională poate fi dezvoltată prin prezentarea calculatorului și a componentelor acestuia copiilor. Într-o anumită măsură, cunoștințele despre hardware și software fac parte din gândirea computațională, dar nu aceasta este competența care trebuie dezvoltată în educația timpurie. O altă concepție greșită a respondenților despre dezvoltarea gândirii computaționale este că aceasta poate fi realizată prin matematică, ceea ce nu este în concordanță cu conceptul de gândire computațională. Punctul de vedere al gândirii computaționale ca rezolvare a problemelor matematice a fost menționat și de Sands și colaboratorii săi (2022), precum și de Avci și Deniz (2022) în cercetările lor. În realitate, gândirea computațională este mai degrabă legată de algoritmi și codare, iar aproape jumătate dintre respondenți au înțeles acest lucru, fapt care a reieșit din exemplele oferite de ei. Cercetarea realizată de Ari și colab. (2022) a relevat, de asemenea, că viitorii educatori nu au integrat codarea în activitățile lor din grădiniță. Deoarece roboții sunt adesea folosiți în activitățile din grădiniță pentru predarea algoritmilor și a codării, studenții au fost întrebați dacă au auzit vreodată de roboți educaționali. Doar 22,1% dintre ei (19 persoane) au oferit un răspuns afirmativ la această întrebare. Printre roboții cunoscuți de ei s-au numărat roboții Blue-bot/Bee-bot (20,9%, 18 studenți), Ozobot, Vex și Cubetto – fiecare fiind menționat de 5 studenți (5,8%). Restul respondenților nu cunoșteau niciun robot educațional. Niciunul dintre respondenți nu a folosit încă roboți educaționali în predare. Acest rezultat este în concordanță cu faptul că în România învățarea cu roboți educaționali nu este încă răspândită și, în general, oferă această posibilitate doar grupei de vârstă școlară, în cadrul câtorva activități extrașcolare. Luând în considerare rezultatele cercetărilor din alte țări, ar fi oportună includerea gândirii computaționale, ca o posibilă arie de dezvoltare, în cerințele curriculare ale grădiniței. Totodată, ar trebui asigurate și instrumente adecvate pentru dezvoltarea acesteia, precum roboți educaționali.

3.2. Opinia educatorilor din învățământul preșcolar despre dezvoltarea abilităților legate de domeniul tehnologic în grădiniță

Scopul cercetării

Scopul cercetării a fost de a obține o imagine asupra opiniilor și experiențelor educatorilor din învățământul public din România privind utilizarea cunoștințelor și instrumentelor tehnologice în educația timpurie.

Întrebările cercetării

În cadrul cercetării, ne-am propus să găsim răspunsuri la următoarele întrebări:

1. Care este atitudinea educatorilor față de aplicarea cunoștințelor tehnologice în grădiniță?
2. Au încercat educatorii să integreze cunoștințele tehnologice în activitățile instructiv-educative din grădiniță?
3. Ce cunoștințe și experiență au despre gândirea algoritmică/computațională și despre posibilitățile de dezvoltare a acestora în grădiniță?
4. Care este atitudinea educatorilor față de dezvoltarea gândirii computaționale în grădiniță?
5. Cunosc aceștia roboții educaționali?
6. Ce experiență au cu privire la utilizarea roboților educaționali?

Metodologie

Prezenta cercetare s-a desfășurat în anii școlari 2021-2022 și 2022-2023, în România.

Participanți

La sondaj au participat 115 educatori de grădiniță care activează în învățământul public din România, în județele Harghita, Covasna, Mureș, Cluj, Sibiu, Bihor, Sălaj, Satu Mare, Timiș și Bistrița-Năsăud. Toți participanții sunt femei. Din punct de vedere al studiilor, 81% dintre educatoare (93 persoane) au studii universitare (licență), restul având și studii de masterat (MSc). În ceea ce privește gradul didactic obținut până în prezent, 49% dintre participanți (56 persoane) dețin gradul I (cel mai înalt), 20% (23 persoane) dețin gradul II, 14% (20 persoane) au finalizat examenul de definitivat, iar 17% (16 persoane) sunt educatori debutanți. În ceea ce privește mediul de lucru, 60% dintre respondenți (69 persoane) lucrează în oraș, iar 40% (46 persoane) în mediul rural.

Instrumentar

În cadrul cercetării, am utilizat un chestionar elaborat online în Google Forms, care conținea 23 de întrebări: 6 întrebări referitoare la date demografice diferite, 7 întrebări

vizau aplicarea cunoștințelor tehnologice în grădiniță, iar 10 întrebări se refereau la importanța și posibilitățile de dezvoltare a gândirii algoritmice/computaționale, abordând și tema roboților educaționali de podea. Chestionarul conținea 5 întrebări deschise, iar restul erau întrebări închise (întrebări cu răspunsuri multiple, respectiv afirmații măsurate pe o scală Likert de 5 puncte).. În chestionar, întrebările legate de gândirea algoritmică și computațională au apărut în secțiuni separate. Pentru a ne asigura că toți cei care completează chestionarul înțeleg același lucru prin gândire algoritmică și computațională, am plasat înaintea întrebărilor definițiile acceptate de noi pentru ambele concepte.

Rezultate

Rezultatele au fost grupate în trei teme, combinând câte două întrebări de cercetare: (1) cunoștințele tehnologice în grădiniță; (2) dezvoltarea gândirii algoritmice/computaționale; (3) utilizarea roboților de podea în grădiniță.

1. Cunoștințe tehnologice în activitățile instructiv-educative din grădiniță

Pentru a evalua opiniile educatorilor cu privire la introducerea cunoștințelor tehnologice în grădiniță, aceștia au trebuit să evalueze patru afirmații pe o scală Likert de cinci puncte (unde 1=nu sunt deloc de acord și 5=sunt complet de acord). Valorile medii și abaterile standard ale răspunsurilor lor sunt prezentate în Tabelul 4.

Tabelul 4. Valorile medii și abaterile standard pentru fiecare afirmație referitoare la aplicarea cunoștințelor tehnologice în grădiniță

Afirmație	Medie	Abatere standard
În perioada preșcolară este necesară introducerea cunoștințelor tehnologice.	3,05	1,234
În perioada preșcolară nu este locul cunoștințelor tehnologice.	2,18	1,288
În perioada preșcolară putem pune bazele unei atitudini pozitive față de activitățile tehnologice.	3,41	1,199
Sarcinile tehnologice din grădiniță includ bazele programării (planificare, execuție)	2,91	1,274

Conform răspunsurilor, educatorii preșcolari sunt cel mai mult de acord cu ideea că în

perioada preșcolară putem pune bazele unei atitudini pozitive față de activitățile tehnologice. Totodată, majoritatea sunt de acord și cu faptul că în perioada preșcolară este necesară introducerea cunoștințelor tehnologice. Sunt mai puțin de acord cu afirmația că sarcinile tehnologice din grădiniță includ punerea bazelor programării (planificare, execuție).

O întrebare de analizat a fost dacă există vreo diferență între răspunsurile educatorilor care își desfășoară activitatea în mediul rural, respectiv în mediul urban. Deoarece, conform testului Shapiro-Wilk, eșantionul nu are o distribuție normală, pentru comparație am aplicat testul Mann-Whitney, pe baza căruia nu există o diferență statistic semnificativă între opiniile educatorilor preșcolari care lucrează în mediul rural, respectiv în mediul urban, cu privire la cunoștințele tehnologice, deși cei care predau în mediul urban au obținut o medie mai mare pentru fiecare afirmație, comparativ cu cei care predau în mediul rural (tabelul 5).

Tabelul 5. Comparația răspunsurilor educatorilor care predau în mediul rural și cel urban cu privire la aplicarea cunoștințelor tehnologice în grădiniță folosind testul Mann-Whitney

<i>Afirmație</i>	<i>Mediu rural</i> <i>N=41</i>		<i>Mediu urban</i> <i>N=66</i>		<i>W</i>	<i>p</i>
	<i>Medie</i>	<i>Abatere standard</i>	<i>Medie</i>	<i>Abatere standard</i>		
În perioada preșcolară este necesară introducerea cunoștințelor tehnologice.	2,98	1,18	3,10	1,27	1510,00	,652
În perioada preșcolară nu este locul cunoștințelor tehnologice.	2,26	1,34	2,13	1,26	1666,50	,636
În perioada preșcolară putem pune bazele unei atitudini pozitive față de activitățile tehnologice.	3,26	1,10	3,52	1,25	1362,50	,184
Sarcinile tehnologice din grădiniță includ bazele programării (planificare, execuție)	2,85	1,10	2,96	1,39	1526,00	,722

S-a ridicat întrebarea dacă nivelul de studii al educatorilor influențează opiniile lor cu privire la introducerea activităților tehnologice în grădiniță. Deoarece, conform testului Shapiro-Wilk, eșantionul nu are o distribuție normală, pentru comparație am aplicat din nou testul Mann-Whitney. Rezultatele sunt cuprinse în tabelul 6. Pe baza primelor două afirmații, educatoarele cu studii de masterat consideră statistic semnificativ mai necesară introducerea cunoștințelor tehnologice în activitățile din grădiniță.

Tabelul 6. Comparația răspunsurilor educatorilor cu studii universitare de licență și masterat cu privire la aplicarea cunoștințelor tehnologice în grădiniță folosind testul Mann-Whitney

<i>Afirmație</i>	<i>Licență</i> <i>N=93</i>		<i>Masterat</i> <i>N=14</i>		<i>W</i>	<i>p</i>
	<i>Medie</i>	<i>Abatere standard</i>	<i>Medie</i>	<i>Abatere standard</i>		
În perioada preșcolară este necesară introducerea cunoștințelor tehnologice.	3,01	1,24	3,71	1,12	440,00	,045
În perioada preșcolară nu este locul cunoștințelor tehnologice.	2,20	1,28	1,36	0,63	894,50	,017
În perioada preșcolară putem pune bazele unei atitudini pozitive față de activitățile tehnologice.	3,37	1,22	3,86	1,17	495,00	,137
Sarcinile tehnologice din grădiniță includ bazele programării (planificare, execuție)	2,86	1,30	3,36	1,15	501,50	,156

Întrebarea următoare a fost dacă educatorii au încercat să integreze cunoștințele tehnologice în activitățile din grădiniță. 55% dintre respondenți (63 persoane) nu au încercat să integreze cunoștințele tehnologice în activitățile din grădiniță, în timp ce 45% (52 persoane) au încercat. Totuși, întrebarea deschisă ulterioară arată că apar probleme conceptuale. Am rugat educatorii să dea exemple concrete despre modul în care au aplicat cunoștințele tehnologice în grădiniță, iar răspunsurile lor au arătat că mulți confundă

educația tehnologică cu utilizarea instrumentelor tehnologice și a mijloacelor digitale. Respondenții au înțeles mai degrabă utilizarea instrumentelor tehnologice drept educație tehnologică (de exemplu platforme online, aplicații, tablă interactivă, utilizarea tabletei, a telefonului și a mouse-ului). Alții au considerat cunoștințele legate de experimentare ca fiind educație tehnologică: observarea funcționării diferitelor dispozitive sau descoperirea relațiilor cauză-efect. Prin urmare, răspunsul primit la întrebarea anterioară a fost distorsionat, mai puțin de 45% dintre respondenți aplicând cunoștințe tehnologice integrate.

Majoritatea educatorilor care au încercat să integreze cunoștințele tehnologice au răspuns că activitățile încercate au fost apreciate de copii, aceștia participând activ la rezolvarea sarcinilor. Următoarea întrebare s-a referit la dificultățile întâmpinate în planificarea activităților. 26,9% dintre educatoare (31 persoane) au considerat lipsa instrumentelor didactice necesare pentru implementare ca fiind responsabilă pentru dificultățile apărute. 16,5% dintre respondenți (19 persoane) au considerat că planificarea unor astfel de activități este îngreunată de pregătirea metodologică insuficientă, 14,7% dintre aceștia (17 persoane) consideră că mediul din grădiniță nu este adecvat pentru astfel de activități, iar 12,1% dintre cei chestionați (14 persoane) au menționat pregătirea teoretică insuficientă ca fiind cauza.

Următoarea întrebare a vizat identificarea factorilor care le-au împiedicat pe educatoare să integreze cunoștințele tehnologice în activitățile din grădiniță. 37,3% dintre respondenți (43 persoane) au indicat pregătirea metodologică insuficientă ca fiind cel mai mare obstacol, același procent, adică 37,3% (43 persoane) au văzut cauza în lipsa materialelor didactice adecvate, în timp ce 30,4% dintre aceștia (35 persoane) consideră pregătirea teoretică insuficientă ca fiind cel mai mare obstacol. Totuși, la întrebarea dacă ar dori să se familiarizeze cu tipuri de activități care se bazează pe introducerea cunoștințelor tehnologice și să planifice ulterior astfel de activități care introduc cunoștințe tehnologice, 91% dintre respondenți (105 persoane) au răspuns afirmativ.

2. Dezvoltarea gândirii algoritmice/computaționale în grădiniță

După citirea definiției gândirii algoritmice, am solicitat opinia educatorilor preșcolari cu privire la activitățile prin care poate fi dezvoltată gândirea algoritmică a preșcolarilor. Răspunsurile au fost foarte variate, la 76,5% dintre aceștia (88 persoane) am găsit exemple de activități care contribuie la dezvoltarea gândirii algoritmice. Dintre acestea, 37,5% (33 persoane) au menționat generic activitățile matematice ca fiind potrivite în acest scop,

subliniind gruparea, clasificarea, sarcinile logice.

În plus, 19,3% dintre aceștia (17 persoane) au menționat serialitatea/crearea de serii (de ex., urmărirea pașilor unei activități sau ordonarea unor evenimente). Strâns legată de aceasta este urmărirea etapelor activităților manuale/artistice, pe care au ales-o 10,2% dintre aceștia (9 persoane), precum și urmărirea rutinelor zilnice, oferită ca răspuns de 13,6% dintre aceștia (12 persoane). Pe lângă acestea, construcțiile au fost considerate de 10,2% (9 persoane) ca fiind o activitate care dezvoltă gândirea algoritmică, jocurile cu reguli de 9,09% (8 persoane), iar activitățile din domeniul științelor/cunoașterii mediului de 10,2% dintre aceștia (9 persoane). Roboții de podea și programarea au apărut ca răspuns doar la 5,6% dintre aceștia (5 persoane). Motivul evident pentru acest lucru este că roboții educaționali de podea și programarea sunt necunoscute educatorilor .

În ceea ce privește experiența lor anterioară cu algoritmi, 77,3% (88 persoane) dintre educatoare au afirmat că au auzit de acest termen doar la orele de matematică, 12,1% (14 persoane) nu au studiat niciodată algoritmi, 4,3% (5 persoane) au studiat programare, iar 2,6% dintre acestea (3 persoane) au participat la o formare care s-a ocupat de gândirea algoritmică. Rezultatele reflectă bine cât de puțin ne ocupăm în mod conștient de dezvoltarea acestui domeniu, deoarece nici cadrele didactice nu au primit o pregătire în acest sens, ba mai mult, majoritatea nu au nici măcar o experiență anterioară adecvată cu acest concept. Cu toate acestea, 91% dintre educatoarele chestionate (105 persoane) consideră că este importantă dezvoltarea gândirii algoritmice în grădiniță.

Întrebările următoare s-au concentrat pe gândirea computațională. La prima întrebare, educatoarele au trebuit să evalueze pe o scală Likert trei afirmații referitoare la dezvoltarea gândirii computaționale, în funcție de gradul lor de acord cu acestea (unde 1=nu sunt deloc de acord și 5=sunt pe deplin de acord).. Valorile medii ale răspunsurilor lor și rezultatele abaterilor standard sunt cuprinse în tabelul 7.

Tabelul 7. Valorile medii și abaterile standard pentru fiecare afirmație privind dezvoltarea gândirii computaționale

<i>Afirmație</i>	<i>Medie</i>	<i>Abatere standard</i>
Dezvoltarea gândirii computaționale necesită întotdeauna un calculator	2,49	1,327
Gândirea computațională poate fi dezvoltată/învățată și fără a utiliza calculatorul.	3,217	1,412
Gândirea computațională poate fi dezvoltată și prin sarcini bazate pe hârtie și creion.	3,48	1,272

Pe baza răspunsurilor, putem observa că majoritatea educatorilor sunt de acord mai ales cu afirmația că dezvoltarea gândirii computaționale se poate realiza și prin sarcini bazate pe hârtie și creion, adică nu necesită utilizarea calculatorului. Acest rezultat este încurajator, deoarece mult timp s-a făcut o echivalență între dezvoltarea gândirii computaționale și utilizarea calculatorului.

S-a ridicat întrebarea dacă există vreo diferență între răspunsurile educatorilor din mediul rural și cei din mediul urban. Pe baza testului Mann-Whitney, nu există o diferență statistic semnificativă între opiniile celor două grupuri cu privire la dezvoltarea gândirii computaționale. Ambele grupuri preferă activitățile pe bază de hârtie și creion pentru dezvoltarea gândirii computaționale. În cazul educatorilor care predau în mediul rural, valoarea medie pentru această afirmație este puțin mai mare.

Totodată, a fost interesant de examinat efectul experienței didactice asupra opiniei referitoare la dezvoltarea gândirii computaționale. În acest scop, am împărțit cadrele didactice în trei grupuri: (1) cei cu mai puțin de 10 ani de experiență, (2) cei cu 10-19 ani de experiență, (3) cei cu mai mult de 19 ani de experiență. Răspunsurile celor trei grupuri au fost comparate utilizând testul ANOVA, rezultatele fiind prezentate în tabelul 8. Rezultatele arată că nu există o diferență statistic semnificativă între răspunsurile celor trei grupuri.

Tabelul 8. Opiniile privind dezvoltarea gândirii computaționale în funcție de experiență

<i>Afirmație</i>	<i>Mai puțin de 10 ani de experiență</i>		<i>10-19 ani de experiență</i>		<i>Mai mult de 19 ani de experiență</i>		<i>F</i>	<i>p</i>
	<i>Medie</i>	<i>Abatere standard</i>	<i>Medie</i>	<i>Abatere standard</i>	<i>Medie</i>	<i>Abatere standard</i>		
Dezvoltarea gândirii computaționale necesită întotdeauna un calculator	2,72	1,43	2,47	1,18	2,43	1,33	,431	,651
Gândirea computațională poate fi	3,24	1,48	3,47	1,18	3,08	1,45	,550	,579

dezvoltată/învățată și fără a utiliza calculatorul.								
Gândirea computațională poate fi dezvoltată și prin sarcini bazate pe hârtie și creion.	3,48	1,26	3,71	1,16	3,40	1,31	,389	,679

În ceea ce privește necesitatea dezvoltării gândirii computaționale în grădiniță, 69% dintre educatori (79 persoane) consideră că este importantă dezvoltarea gândirii computaționale încă din perioada preșcolară.

La întrebarea despre modul în care poate fi dezvoltată gândirea computațională în grădiniță, au sosit răspunsuri foarte diferite. Prelucrarea răspunsurilor la întrebarea deschisă a fost realizată în Maxqda. Figura 1 prezintă diferitele categorii de răspunsuri, cu subcategoriile aferente, detaliate. Răspunsurile la întrebarea noastră (cum poate fi dezvoltată gândirea computațională?) au fost grupate în șase categorii principale: joc, dezvoltarea abilităților matematice, utilizarea instrumentelor digitale, jocuri speciale pentru dezvoltarea gândirii algoritmice, activități cotidiene, dezvoltarea gândirii științifice. În cadrul acestora, am definit mai multe subcategorii, iar subcategoriile au fost detaliate în subunități.

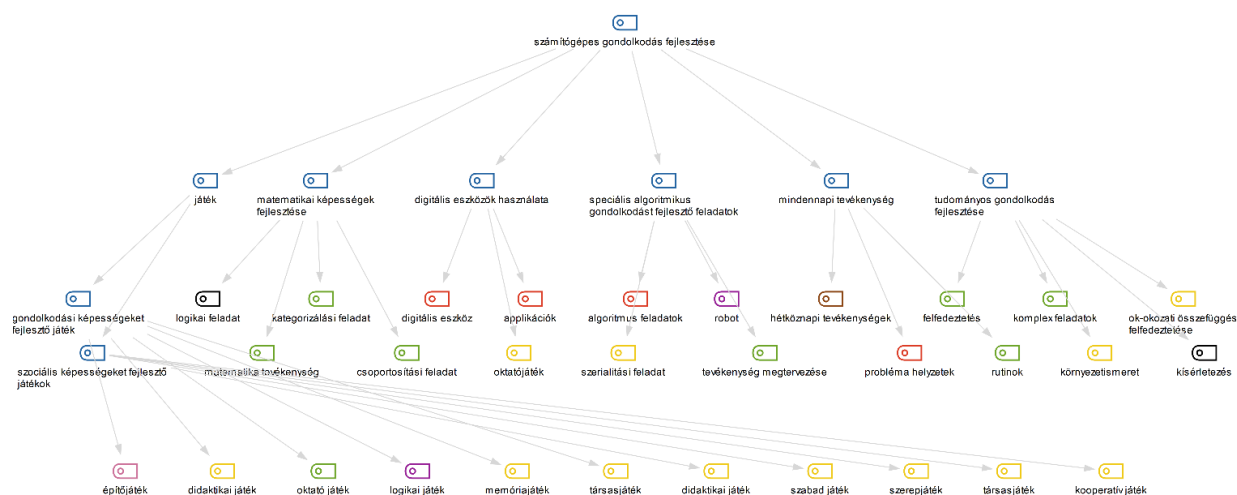


Figura 1. Model ierarhic de cod-subcod în MAXQDA

În cazul categoriei jocurilor, am definit două subcategorii, pentru aceste două subgrupuri

identificând și subcategorii comune. În categoria principală a dezvoltării abilităților matematice, am identificat patru subgrupuri. În cadrul subcategoriilor, cele mai multe mențiuni (7 persoane) au fost pentru sarcinile logice, considerate activități potrivite pentru dezvoltarea gândirii computaționale. Grupul instrumentelor digitale a fost divizat în trei subcategorii, dintre care utilizarea instrumentelor digitale a fost considerată de cele mai multe persoane (7 persoane) ca fiind potrivită pentru dezvoltarea gândirii computaționale. Categoria sarcinilor speciale care dezvoltă gândirea algoritmică are patru subcategorii. Subcategoria sarcinilor de serialitate a fost menționată de cele mai multe ori (8 persoane) ca fiind o activitate potrivită pentru dezvoltarea gândirii computaționale. Un rezultat interesant este că, din toți respondenții, doar un educator a sugerat roboții ca o posibilitate de dezvoltare a gândirii computaționale. Din categoria principală a activităților cotidiene, subcategoria situațiilor problematice a fost menționată de cei mai mulți (5 persoane) ca fiind o activitate potrivită pentru dezvoltarea gândirii computaționale. Al șaselea grup mare este dezvoltarea gândirii științifice, dintre subcategoriile căruia explorarea a fost menționată de cele mai multe ori (4 persoane) ca fiind o activitate adecvată în dezvoltarea gândirii computaționale.

3. Utilizarea roboților educaționali în grădiniță

Doar 37% dintre educatoarele chestionate (42 de persoane) au auzit despre roboții de podea. Dintre roboți, cel mai cunoscut este Blue-bot, urmat de Cubetto, Ozobot și șoricelul robot Colby. La întrebarea dacă au folosit vreunul dintre acești roboți în activitățile lor, doar 3,4% dintre ele (4 persoane) au răspuns afirmativ. Când le-am cerut să descrie o astfel de activitate, o educatoare a menționat robotul Cubetto, dar nu a descris modul de utilizare, două au menționat robotul Bee-bot, dintre care o educatoare doar a achiziționat instrumentul, dar nu l-a folosit încă, iar una a relatat în detaliu o activitate. Prin urmare, pe baza acestor date, nu se pot face constatări relevante despre experiențele legate de utilizarea roboților de podea în grădiniță.

1. CREAREA ȘI PRETESTAREA TESTULUI DE GÂNDIRE COMPUTAȚIONALĂ ALGOPAINT

4.1. Elaborarea testului de gândire computațională AlgoPaint

Pe baza cercetărilor internaționale, toate testele prezentate în cel de-al doilea capitol au una sau mai multe dintre caracteristicile pe care am încercat să le eliminăm din testul nostru de gândire computațională creat de noi. Unele dintre testele prezentate nu răspund nevoilor noastre deoarece necesită cunoștințe prealabile de programare, presupun utilizarea unui instrument digital, pot fi efectuate online sau presupun abilități de orientare spațială și rotație mentală. Pentru a le exclude, am luat în considerare următoarele criterii la conceperea testului nostru:

- să corespundă copiilor preșcolari (4-7 ani)
- să nu presupună folosirea unui ecran sau a unui calculator
- să nu depindă de instrumente digitale și tehnice adică, să nu utilizeze un robot educațional, deoarece în testele în care sunt utilizați roboți educaționali pentru evaluare, copiii care au o experiență anterioară cu roboții educaționali au un avantaj. Pe de altă parte, dacă testul urmează să fie utilizat ca pre-test și post-test în cadrul unei evaluări a eficacității unei intervenții bazate pe roboți educaționali, grupul experimental poate obține rezultate mai bune la post-test doar pentru că în cadrul intervenției sunt utilizați roboți educaționali.
- sarcinile testului să nu necesite cunoștințe prealabile de programare
- sarcinile să nu conțină direcții, deoarece copiii care nu au o orientare spațială suficient de dezvoltată, în special rotația mentală, pot fi dezavantajați în rezolvarea sarcinilor.

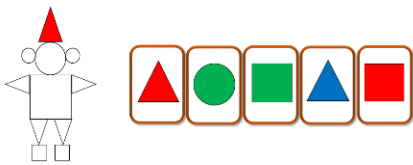
Caietul de testare AlgoPaint pe care l-am dezvoltat se bazează pe o poveste concepută pentru a stimula și a menține motivația și interesul copiilor. Sarcina este de a-l ajuta pe Spiridușul Pictor să creeze forme (roboți, viermi) pornind de la diferite forme geometrice. Instrucțiunile sunt date pe cartonașe ilustrate. Aceste cartonașe ilustrate sunt similare cartonașelor din cadrul proiectului Algolittle Erasmus+, concepute pentru desen creativ bazat pe gândire algoritmică.

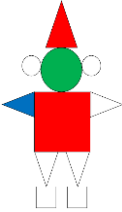
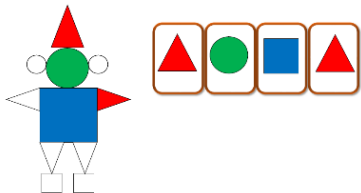
Conform poveștii, spiridușul pictor desenează conturul unei forme geometrice și apoi pictează aceste forme una câte una cu culori diferite. El începe întotdeauna să picteze de la o margine, deci pictează forma geometrică pe care se află, apoi trece la forma care este în

contact cu cea pe care a pictat-o deja. Procedează cu formele în succesiune, una după alta, și trebuie să aibă grijă să nu calce înapoi pe o formă care a fost deja pictată, deoarece aceasta se usucă lent și, prin urmare, se poate lipi. Conform acestei reguli, se poate lucra cu formele doar în ordinea dată, dar nu există o singură ordine corectă în care să se lucreze. Ordinea aleasă în soluție depinde dacă algoritmul poate fi continuat sau nu. Regula conform căreia nu vă puteți întoarce la o formă care a fost deja pictată vă permite să înțelegeți că fiecare instrucțiune simplă se referă la forma geometrică curentă de la care pictorul spiriduș merge mai departe, astfel încât fiecare formă geometrică are o culoare asociată.

Testul AlgoPaint măsoară următoarele abilități mentale computaționale: aplicarea algoritmului, proiectarea algoritmului, depistarea erorilor de către algoritm. Tabelul 9 oferă o listă a abilităților mentale computaționale măsurate, împreună cu itemii aferenți măsurării fiecărei abilități.

Tabelul 9. Abilitățile mentale algoritmice măsurate cu testul AlgoPaint

<i>Abilitatea mentală computațională</i>	<i>Descriere</i>	<i>Exemple pe itemi</i>
<i>Aplicarea algoritmului</i>	Capacitatea de a identifica rezultatul unui anumit șir de instrucțiuni	Itemul 1: „Ajută-l pe Spiridușul Pictor, colorează robotul. Ți-a înșirat instrucțiunile, urmează-le. Sunt curios dacă poți să colorezi robotul pe baza cartonașelor.” 

Proiectarea algoritmului	Abilitatea de a identifica șirul de instrucțiuni necesare pentru a obține un anumit rezultat	Itemul 6: „Ajută-l pe Spiridușul Pictor să pună instrucțiunile în ordine! Spiridușul a colorat robotul, iar ție ți-a trimis cartonașele cu instrucțiuni. Trebuie să așezi cartonașele în ordine, așa cum au fost pictate de Spiridușul Pictor. Așază cartonașele într-un rând, unul lângă celălalt. Sunt curios dacă poți aranja cartonașele conform desenului.” 
Depistarea erorii în algoritm	Capacitatea de a compara un șir de instrucțiuni și rezultatul asociat/ dat	Itemul 4: „Spiridușul Pictor a colorat robotul. A înșirat instrucțiunile, iar sarcina ta este doar să verifici dacă a colorat exact conform cartonașelor. A ordonat corect cartonașele?” 

Dintre itemii testului, nu doar sarcinile concepute pentru a găsi erori conțineau erori, dar pentru fiecare item le-am cerut copiilor să verifice dacă pot găsi greșeli în exercițiu.

Broșura testului a inclus, de asemenea, instrucțiuni specifice privind modul de utilizare a testului, trei exemple rezolvate, un sistem de punctare, soluții la probleme, cartonașe cu imagini pentru exerciții.

4.2. Pre-testarea testului de gândire computațională AlgoPaint

AlgoPaint a fost testat în anul școlar 2021-2022, cu 11 educatoare în propria lor grupă, cu un total de 56 de copii în vârstă de 5-6 ani. Profesorii de învățământ preșcolar au fost selectați din cadrul cursului STEM al programului de masterat. În plus, broșura de testare a fost trimisă profesorilor universitari care sunt experți în domeniu. Experții au fost selectați din 6 țări diferite și au fost rugați să completeze un chestionar online anonim despre

broșura de testare. Chestionarul a cuprins 9 întrebări, 5 întrebări închise (cu alegere multiplă și scalare) și 4 întrebări deschise. Scopul chestionarului a fost de a afla dacă aceștia consideră că exercițiile sunt adecvate pentru grupul de vârstă țintă, dacă sunt potrivite pentru măsurarea abilităților de gândire algoritmică și, de asemenea, de a obține opinia lor cu privire la claritatea explicațiilor din broșura de testare. Rezultatele testelor celor 56 de preșcolari care au luat parte la pretestare sunt rezumate în tabelul 10.

Tabelul 10. Rezultatele copiilor în Testul de măsurare a abilităților de gândire computațională AlgoPaint

<i>Item</i>	<i>Abilitate de gândire computațională-</i>	<i>Introducere a unei erori în exercițiu</i>	<i>Numărul copiilor care au răspuns corect</i>	<i>Procentul copiilor care au răspuns corect</i>	<i>Cele mai frecvente erori</i>	<i>Numărul copiilor care au greșit</i>
1	Aplicare	nu	22	39,29	Colorează mai multe forme, decât în instrucțiuni	15
					Nu alege forma alăturată corespunzătoare	12
2	Aplicare	da	23	41,07	Nu depistează greșeala	32
3	Aplicare	da	25	44,64	Nu depistează greșeala	28
4	Detectarea greșelii	nu	49	87,50		
5	Detectarea greșelii	da	33	58,93	Nu depistează greșeala	23

6	Proiectare	nu	44	78,57	Înșiruire greșită a instrucțiunilor	10
7	Proiectare	da	30	53,57	Nu depistează greșeala	22

Rezultatele arată că cele mai dificile sarcini au fost exercițiile de aplicare (41,6% rezolvate corect), urmate de sarcinile de proiectare (66,07% rezolvate corect), iar cele mai ușoare sarcini au fost sarcinile de depistare a greșelilor (73,22% rezolvate corect).

Pe lângă rezultatele copiilor, la evaluarea testului am luat în considerare și opiniile și sugestiile educatoarelor. Răspunsurile primite de la 11 cadre didactice au arătat că testul li s-a părut interesant și antrenant, povestea din spatele lui fiind deosebit de interesantă. Unii au considerat testul dificil, în timp ce alții l-au găsit potrivit pentru nivelul copiilor preșcolari. Educatoarele au subliniat, de asemenea, că sunt necesare instrucțiuni mai precise pentru realizarea testului, deoarece multe dintre ele l-au utilizat în grup, ceea ce a îngreunat notarea scorurilor, deoarece nu au putut fi atente la răspunsurile tuturor copiilor în același timp. De asemenea, s-a dovedit că instrucțiunile pentru sarcini trebuiau să fie clarificate și standardizate, deoarece unele educatoare au furnizat informații suplimentare cu fiecare sarcină, facilitând rezolvarea sarcinilor de către copii. În plus față de acestea, educatoarele de grădiniță au menționat, de asemenea, că mulți copii nu au fost capabili să se concentreze asupra mai multor lucruri/obiective în același timp, de exemplu, au urmat instrucțiunile, au colorat forma potrivită cu culoarea potrivită, dar nu au luat în considerare condiția de adiacență. S-a sugerat, de asemenea, că povestea ar trebui să menționeze și faptul că Spiridușul Pictor ar putea greși.

Pe lângă feedback-ul educatoarelor, am analizat și evaluările experților. Majoritatea experților consideră că testul este adecvat pentru copiii de 5-6 ani, cu un număr suficient de sarcini pentru a evalua competențele vizate. Cu toate acestea, am încercat, de asemenea, să evaluăm măsura în care testul măsoară aplicarea algoritmilor, planificarea și depistarea greșelilor: pe o scară de la 1 la 5, le-am cerut să evalueze gradul de adecvare al sarcinilor. Media evaluărilor experților a fost de 4,83 pentru aplicarea algoritmilor, 4,33 pentru depistarea greșelilor și 4,33 pentru proiectarea algoritmilor. Majoritatea acestora au

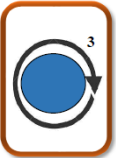
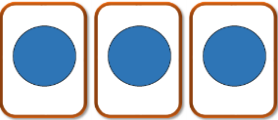
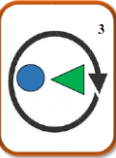
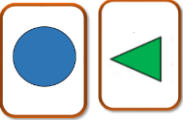
considerat că testul a fost ușor de înțeles pentru copii, dar au recomandat ca în broșura de testare să fie inclusă o explicație mai precisă a conceptului de adiacență.

Pe baza rezultatelor testului pilot și a feedback-ului din partea experților și a educatoarelor, testul a fost corectat. Următoarele modificări au fost adăugate la cea de-a doua versiune a broșurii de test AlgoPaint Computer Reasoning.

- În broșura de testare au fost adăugate încă 5 elemente, care măsoară mai multe/alte concepte de gândire algoritmică
 - În prima versiune, a fost inclus doar un exemplu de structuri de repetiție, fără o sarcină corespunzătoare. În noua versiune, au fost incluse două sarcini de repetiție simplă și un element de repetiție multiplă
 - Un alt concept de gândire algoritmică este cel al structurilor condiționale, pentru care am dezvoltat itemi de măsurare: au fost incluse un model și o sarcină asociată.
 - Am elaborat mai detaliat sistemul de punctare
 - În poveste am menționat că Spiridușul Pictor uneori greșește
 - Instrucțiunile de utilizare a broșurii de testare sunt detaliate, până în punctul în care acestea trebuie doar citite utilizatorului. Scopul a fost acela de a se asigura că modul de utilizare este consecvent și că rezultatele nu sunt distorsionate de instrucțiunile date copiilor.
 - Am adăugat și alte forme la broșură: pe lângă roboți, am făcut forme de viermi
- Tabelul 11 cuprinde conceptele de gândire computațională utilizate în a doua versiune

Tabelul 11. Concepte de gândire computațională în varianta Algopaint 2.

<i>Concepte de gândire computațională</i>	<i>Descriere</i>	<i>Exemple de cartonașe</i>	<i>Explicația cartonașelor</i>
Instrucțiuni simple	Colorarea unei forme geometrice cu o anumită culoare		Triunghiul trebuie colorat cu roșu

Repetiție simplă	O structură care permite ca o instrucțiune simplă să fie repetată de mai multe ori		Înseamnă același lucru ca următorul set simplu de instrucțiuni : 
Repetiție multiplă	O structură care permite ca o secvență de instrucțiuni simple să fie repetată de mai multe ori		În acest caz, următoarea serie simplă de instrucțiuni trebuie repetată de trei ori: 
Condiții/ structură condiționată	O structură care permite executarea unei instrucțiuni simple pe baza unei condiții		Ne uităm la condiția din partea de sus a cartonașului: forma e un cerc? Dacă da, o colorăm în albastru, dacă nu, o colorăm în verde.

Versiunea revizuită include acum patru modele rezolvate, două pentru instrucțiuni simple, unul pentru repetiție simplă și unul pentru structură condițională. Exemplele rezolvate sunt scrise textual, astfel încât este suficientă o lectură atentă în timpul punerii în aplicare.

Acest lucru permite tuturor copiilor să primească corect aceleași explicații și, în același timp, administratorul testului nu trebuie să adauge alte explicații la sarcini. Caietul de testare AlgoPaint conține soluțiile corecte și cartonașele cu imagini necesare pentru rezolvarea fiecărei sarcini. De asemenea, am completat și sistemul de notare.

Concluzii

Pre-testarea AlgoPaint la preșcolari a arătat că testul este adecvat pentru grupul de vârstă țintă și este potrivit pentru măsurarea abilităților de gândire computațională, cum ar fi aplicarea, depistarea greșelilor și planificarea algoritmilor. Rezultatele obținute de copii au arătat că testul a fost adecvat pentru nivelul lor de competență. Reacțiile educatoarelor care au efectuat evaluarea și ale profesorilor universitari, experți în dezvoltarea competențelor computaționale au oferit idei valoroase pentru îmbunătățirea testului. Pe baza rezultatelor,

a fost elaborată o versiune îmbunătățită a instrumentului de evaluare a gândirii computaționale AlgoPaint.

5. REALIZAREA ȘI PRE-TESTAREA PROGRAMULUI DE INTERVENȚIE

5.1. Proiectarea și prezentarea programului de intervenție

Pentru a încorpora și aplica competențele tehnice, am elaborat un program de dezvoltare a gândirii computaționale la copiii preșcolari. Pentru intervenție, am folosit șoricelul robot de podea Colby. Activitățile au constat în sarcini progresiv mai dificile, care au fost concepute într-o manieră ludică, ținând seama de caracteristicile specifice vârstei. Ordinea de dificultate s-a bazat pe trei niveluri.

Nivelul I:

- a. Copilul duce șoricelul în mână până la capăt, pas cu pas
- b. Copilul duce șoricelul în mână și între timp îl programează

Nivelul II:

- a. Copilul duce șoricelul în mână până la capăt și între timp așază cartonașele care indică direcțiile
- b. Așază cartonașele care indică direcțiile și îl programează pe baza acestuia

Nivelul III:

- a. Doar programează fără cartonașe cu imagini. La acest nivel, există mai multe sarcini de dificultate variabilă.

Cadrul de activități a fost următorul: *Șoricelul este un explorator, curios de orice, mereu dornic să vadă locuri noi și suficient de curajos să meargă în aventuri. Atunci când îi vizitează pe copiii de la grădiniță, este mereu pregătit pentru o nouă provocare, așa că de fiecare dată când ei se joacă cu șoricelul, va trebui să-l învețe lucruri noi.*

În planul de intervenție, pentru fiecare activitate, am identificat instrumentele necesare, forma de organizare și o scurtă descriere a pașilor de urmat. Odată ce programul a fost elaborat, l-am încercat și l-am testat în propria grupă de grădiniță.

5.2. Pre-testarea programului de intervenție

În cadrul cercetării noastre, am dorit să aflăm dacă intervenția noastră va schimba gândirea algoritmică/computațională a copiilor preșcolari. Am emis ipoteza că, după finalizarea activităților, rezultatele post-test vor fi mai bune comparativ cu cele din pre-test.

Participanți

Am încercat programul cu 6 copii, cu copii de grădiniță din grupele mijlocie și mare: doi copii din grupa mijlocie (4 ani): un băiat și o fată, patru copii din grupa mare (5 ani): doi băieți și două fete. Intervenția a avut loc în decembrie 2022.

Instrumente de evaluare

Înainte de intervenție, am folosit două instrumente pentru a evalua gândirea computațională a participanților. Ambele instrumente au fost administrate individual. Primul test este un instrument de evaluare dezvoltat de noi: AlgoPaint (Algopictură), un instrument bazat pe hârtie și creion care măsoară urmărirea pașilor algoritmului, crearea și depistarea greșelilor algoritmului (Bálint-Svella și Zsoldos-Marchis, 2023). Cu toate acestea, include, de asemenea, sarcini de măsurare a structurilor repetitive simple și multiple și a structurilor condiționale de tip „dacă-atunci”.

Celălalt test este The competent Computational Thinking Test (cCTt), dezvoltat de El-Hamamsy și colegii săi în 2022. Din acesta, am evidențiat elemente care se regăsesc și în caietul nostru de teste și anume: urmărirea algoritmilor, sarcini de structură repetitivă simplă și multiplă și o sarcină de structură condițională „dacă-atunci”.

Procesul de intervenție

După ce au învățat povestea, au efectuat 9 sarcini diferite cu robotul cu roțițe.

Rezultate

Evaluarea rezultatelor pre-test și post-test ale participanților la testul AlgoPaint

În primele trei sarcini (I.1.a, I.1.b, I.1.c) a trebuit să coloreze conform instrucțiunilor date. În doi dintre itemi, ordinea instrucțiunilor era incorectă, iar copilul trebuia să observe acest lucru. La pre-test, primul item a fost rezolvat corect de 3 copii, al doilea de 2 și al treilea de 2. Motivele erorilor au fost: unii copii au colorat la întâmplare sau nu au luat în considerare condiția de adiacență. La post-test, numărul de soluții corecte la aceiași itemi: primul și al doilea item au fost rezolvate corect de 4 copii, iar al treilea de 5 copii. Motivul erorilor a fost nerespectarea condiția de adiacență.

La următoarele două exerciții (I.2.a, I.2.b), a trebuit să decidă dacă algoritmul a fost corect sau incorect, adică dacă forma pre-colorată a fost colorată în conformitate cu instrucțiunile adiacente. La pre-test, 5 copii au răspuns corect la prima sarcină și doar 1 copil a răspuns corect la a doua. Cei care nu au răspuns corect nu au observat ordinea inversată, ci pur și

simplu au asociat cărțile cu formele. În cadrul post-testului, numărul de răspunsuri corecte a fost următorul: toți copiii au răspuns corect la primul item și 3 au răspuns corect la al doilea item.

Următoarele două sarcini (I.3.a, I.3.b) le cereau copiilor să creeze algoritmi: ei trebuiau să pună instrucțiunile în ordinea corectă pentru diagramele pe care le creaseră. La pre-test, 4 copii au rezolvat corect primul item și niciunul nu a rezolvat corect al doilea item (în care trebuiau să observe eroarea din diagramă). Motivul a fost că nu au luat în considerare condiția conform căreia nu se poate reveni la o formă care a fost deja colorată. La post-test, primul item a fost rezolvat corect de toți copiii, iar al doilea item a fost rezolvat corect de un singur copil.

Următorii trei itemi (II.a, II.b, II.c) au cuprins sarcini repetitive simple și multiple: copiii a trebuit să recunoască repetiția pe instrucțiune și să coloreze ca atare. La pre-testare, 4 copii au răspuns corect la primul și al doilea item și 5 copii au răspuns corect la al treilea item. Erorile s-au datorat faptului că nu au recunoscut cartonașul cu structura repetiției. La post-test, toți copiii au rezolvat corect toți itemii.

În ultimele două sarcini (III.a, III.b), a fost utilizată structura condițională „dacă-atunci”: copiii a trebuit să coloreze cartonașele pe baza recunoașterii lor și să decidă dacă instrucțiunile pentru modelul colorat au fost plasate corect. În pre-test, niciun copil nu a putut rezolva primul item, dar un copil a răspuns corect la al doilea, însă nu a putut explica de ce setul de instrucțiuni era corect. În post-test, însă, au existat 4 răspunsuri corecte pentru primul item și 3 răspunsuri corecte pentru al doilea.

Compararea rezultatelor pre-testului și post-testului

La câteva săptămâni după intervenție (în ianuarie 2023), cele două teste au fost administrate din nou, în același format și în aceleași condiții ca prima dată, la pretestare. Rezultatele sunt rezumate în tabelul de mai jos.

Tabelul 12: rezultatele pre-test și post-test în cele două intervenții

Participanți	Rezultate Algopaint			Rezultate test Cct		
	în %			în %		
	Pre-test	Post-test	Diferență	Pre-test	Post-test	Diferență
1.	33,34	41,67	8,34	33,34	16,67	-16,67
2.	66,67	83,34	16,67	33,34	66,67	33,34
3.	25,00	75,00	50,00	83,34	50,00	-33,34

4.	58,34	83,34	25,00	66,67	100,00	33,34
5.	41,67	83,34	41,67	16,67	83,34	66,67
6.	33,33333	83,33333	50	66,66667	83,33333	16,66666

Datele arată că au existat diferențe între rezultatele pre- și post-test pentru ambele teste. La testul AlgoPaint, toți copiii au obținut scoruri mai mari, adică au dat mai multe răspunsuri corecte în post-test. La testul cCTt, 4 copii și-au îmbunătățit, de asemenea, rata răspunsurilor corecte, dar doi copii au dat mai puține răspunsuri corecte în post-test. Acești doi copii au obținut rezultate mai bune la testul Algofest în post-test: unul a dat cu 8,33% mai multe răspunsuri corecte, iar celălalt copil a avut cu 50% mai multe soluții corecte în post-test decât în prima evaluare. În ceea ce privește procentul rezultatelor, putem spune că rezultatele la testul Algofest au fost influențate pozitiv de intervenția noastră, cu o tendință de creștere a numărului de răspunsuri corecte.

Elaborarea versiunii finale a programului de intervenție

După testarea pilot a intervenției, am revizuit programul de intervenție, deoarece a devenit clar în repetate rânduri în timpul activităților că acestea nu erau adecvate pentru grupul de vârstă preșcolară. În versiunea revizuită, am luat în considerare rezultatele propriilor mele observații, am adăugat informații noi și am descris activitățile mult mai detaliat. În total, au fost elaborate nouă activități. În versiunea finală, am păstrat cele trei niveluri de dificultate bazate pe principiul gradualității, precum și povestea de pornire. Fiecare descriere a activității a fost precedată de o imagine care arăta forma traseului pe care urma să-l facă. În noua versiune, fiecare activitate a fost elaborată în detaliu. La sfârșitul broșurii programului, toate instrumentele ilustrative au fost pregătite ca anexe. Nivelul de detaliu din broșura de program revizuită permite utilizatorului să fie obiectiv și să îndeplinească sarcina cu precizie.

6. EVALUAREA IMPACTULUI PROGRAMULUI DE INTERVENȚIE

6.1. METODOLOGIE

Ipoteze:

În studiul nostru, am investigat dacă programul de intervenție are efecte pozitive asupra gândirii computaționale a copiilor de grădiniță.

H1. Să presupunem că există diferențe în ceea ce privește abilitățile de gândire computațională între copii începând cu vârsta preșcolară, în funcție de vârstă.

H2. Abilitățile de gândire computațională ale copiilor preșcolari sunt dezvoltate prin intermediul unui program bazat pe activități STEM, care utilizează roboți educaționali.

H3. Prin utilizarea roboților educaționali cu activități STEM, programul poate ajuta, de asemenea, preșcolarii să învețe concepte de programare mai puțin familiare pentru ei în mediul preșcolar.

Participanți

La studiu au participat 41 de preșcolari din trei grupe de grădiniță. Toți copiii au făcut parte din grupul experimental. Deoarece dezvoltarea abilităților de gândire computațională nu este o prioritate în învățământul preșcolar, nu am apelat la comparații cu grupul de control. Programul de intervenție cu roboți nu a fost comparat cu alte metode de dezvoltare, dar dezvoltarea inițială și post-intervenție a copiilor a fost examinată în raport cu ei înșiși. În ceea ce privește vârsta, copiii în vârstă de 6 ani au fost majoritari, cu 59% (23) din copiii în vârstă de 6 ani și 41% (16) din copiii având 5 ani împliniți. Educatoarele au fost rugate, în prealabil, să nu selecteze copii mai mici de 5 ani pentru studiu. Două dintre grupele de grădiniță incluse în studiu au fost grupe de grădiniță din mediul rural și o grupă de grădiniță din mediul urban. Grupele au fost eterogene din punctul de vedere al vârstei.

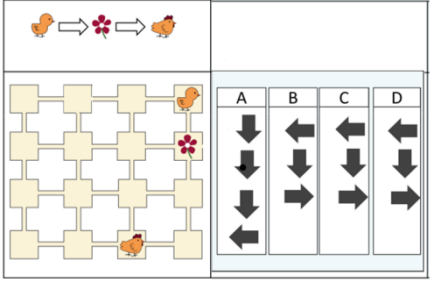
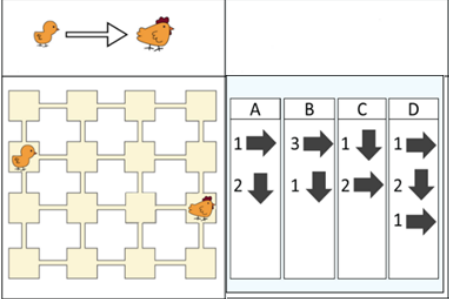
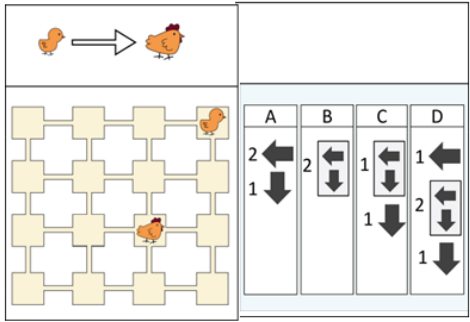
Instrumente de evaluare:

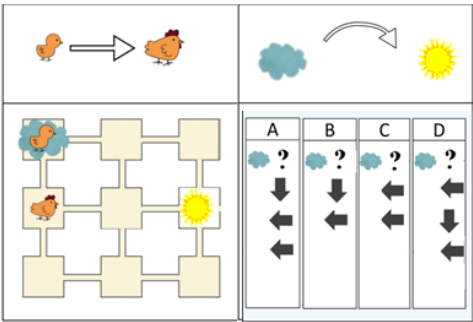
1. AlgoPaint: testul pe care l-am dezvoltat are ca scop evaluarea gândirii computaționale a copiilor cu vârste cuprinse între 4 și 7 ani (Zsoldos-Marchis și Bálint-Svella, 2023).

2. cCTt: El-Hamamsy și colegii (2022) au validat testul cCTt care reprezintă un instrument de evaluare a gândirii computaționale la copii cu vârste cuprinse între 7-9 ani. Am adaptat câțiva itemi din acest test, creând o versiune prescurtată pe care am folosit-o în cercetarea noastră. Broșura de test pe care am întocmit-o conține 6 activități. Itemii acopereau aceleași concepte de gândire computațională ca și sarcinile din broșura de testare AlgoPaint: instrucțiuni simple, repetiții simple, repetiții multiple și structuri condiționale. Iar pentru abilitățile de gândire computațională, am evaluat modul de aplicare

a algoritmului. Caietul de testare începe cu un ghid care oferea o descriere detaliată a ceea ce trebuie făcut. În broșura de test, sarcinile sunt tip grilă, adică trebuia să găsească și să selecteze opțiunea corectă din cele patru. Scorul putea fi 0 sau 1 pentru fiecare sarcină. Tabelul 13 prezintă conceptele de gândire computațională pentru care am elaborat itemii, precum și exemple ale acestor itemii.

Tabelul 13. Concepte de gândire computațională în versiunea adaptată a testului

Concepte de gândire computațională	Descriere	Exemple
Instrucțiuni simple	Un șir de instrucțiuni pentru a conduce puiul către mama sa pe grila pătrată folosind săgețile .	
Repetiție simplă	O structură care permite ca o instrucțiune să fie repetată de mai multe ori.	
Repetiție multiplă	O structură care permite ca un șir de instrucțiuni să fie repetat de mai multe ori	

Condiții/ Structură condițională	O structură care permite executarea unei instrucțiuni pe baza unei condiții	
--	---	--

Procesul de intervenție

Programul de intervenție a fost pus în aplicare în anul școlar 2022-2023. Ca prim pas, ne-am întâlnit cu cadrele didactice de la grădiniță, cărora li s-au prezentat caiete de teste detaliate pentru a fi utilizate în grupele lor. Caietele de teste au fost prezentate unul câte unul, cu instrucțiuni detaliate cu privire la modul și momentul efectuării acestora. Pentru fiecare copil au fost pregătite dosare separate cu ambele teste, care au fost livrate grădinițelor în toamna anului 2022. După pretestarea fiecărui grup, ne-am întâlnit din nou cu fiecare educatoare și am prezentat programul de intervenție în detaliu, parcurgând toate activitățile și demonstrând materialul vizual. Toate educatoarele au avut ocazia să încerce roboții, să monteze pista și să efectueze diferite sarcini de programare. Am livrat roboții fiecărui grup și am păstrat legătura cu educatoarele pe parcursul intervenției. După intervenție, în mai 2023, am distribuit fiecărei grădinițe caietele de teste necesare pentru completarea post-testului, într-un dosar separat pentru fiecare copil. După efectuarea post-testelor, am colectat caietele de teste completate de la toate cele trei grupuri.

6.2. REZULTATE

Am verificat mai întâi dacă au existat diferențe semnificative din punct de vedere statistic între diferitele grupe, atât inițial, cât și după intervenție, pentru a exclude efectul diferiților educatoarelor care au efectuat intervenția în cadrul grupelor. A fost utilizat un test ANOVA pentru a explora acest aspect. Tabelul 14 prezintă aceste rezultate.

Tabelul 14. Diferențele de performanță dintre cele trei grupe

Test	Grupa 1		Grupa 2		Grupa 3		F	p
	Media	Abatere	Media	Abatere	Media	Abatere		
Pre-test AlgoPaint	23,231	6,894	26,250	6,962	22,417	7,025	1,214	,308

Post-test AlgoPaint	30,308	7,052	32,313	4,222	34,667	2,774	2,376	,107
Pre-test cCTt	4,538	1,506	4,313	1,621	4,167	1,267	0,199	,820
Post-test cCTt	5,846	0,376	5,813	0,403	5,917	0,289	0,284	,754

Tabelul arată că nu există nicio diferență semnificativă între rezultatele pre- și post-test ale celor trei grupe, nici pentru testul AlgoPaint, nici pentru testul cCTt. Acest rezultat indică faptul că rezultatele nu au fost afectate chiar dacă intervenția a fost pusă în aplicare de educatoare diferite.

Rezultatele au fost analizate în continuare în conformitate cu ipotezele noastre.

H1. *Se presupune că există diferențe legate de vârstă în ceea ce privește abilitățile de gândire computațională între copii încă din perioada preșcolară.*

Am comparat rezultatele la teste ale copiilor de vârste diferite utilizând un test t pentru eșantioane independente pentru a vedea cum variază performanța copiilor în funcție de vârstă. Rezultatele sunt prezentate în tabelul 15.

Tabelul 15. Rezultatele pre- și pot-test pe vârste

Test	5 ani			6 ani			t	p
	Media	Abatere	N	Media	Abatere	N		
Pre-test AlgoPaint	23,944	6,620	18	24,348	7,420	23	-0,181	,857
Post-test AlgoPaint	31,111	6,211	18	33,348	4,052	23	-1,392	,722
Pre-test cCTt	4,389	1,145	18	4,304	1,690	23	0,182	,857
Post-test cCTt	5,833	0,383	18	5,870	0,344	23	-0,318	,752

Rezultatele arată că scorurile medii ale copiilor în vârstă de 6 ani sunt mai mari la testul AlgoPaint atât la pre-test, cât și la post-test, dar nu există nicio diferență semnificativă în ceea ce privește performanța copiilor în vârstă de 5 și 6 ani.

În primul rând, analizăm rezultatele testului **AlgoPaint**.

În cele ce urmează, comparăm rezultatele pre-test și post-test pentru diferite tipuri de sarcini.

Diferența dintre mediile pre-testului și post-testului pentru sarcinile privind *structura repetiției* este de 0,24. Deoarece testul Shapiro-Wilk arată o abatere de la distribuția normală, a fost utilizat testul Wilcoxon, care indică o diferență semnificativă din punct de vedere statistic între pre-test și post-test. Mărimea efectului pentru acest tip de sarcină este scăzută, deoarece valoarea d-Cohen este de 0,462.

Pentru sarcinile de *structură condiționată*, media scorurilor post-test este de 1,1 ori mai mare decât media scorurilor pre-test. De asemenea, pentru acest tip de sarcină, testul Shapiro-Wilk arată o abatere de la distribuția normală și, prin urmare, a fost utilizat testul Wilcoxon, care arată o diferență semnificativă din punct de vedere statistic între pre-test și post-test. Pentru structura condițională, mărimea efectului este ridicată.

Tabelul 16. Comparație între rezultatele pre-test și post-test ale testului AlgoPaint

Algo- Paint	Pre-test		Post-test		Test normalitate Shapiro-Wilk		Test	t / W (statis- tics)	p	Mări mea efec- tului
	Medi a	Abate re	Med ia	Abatere	W	p				

Media pe activități a scorului total al testului	2,00	0,57	2,67	0,44	0,982	,753	Student	6,990	<.001	1,079
Instrucțiuni simple	2,04	0,68	2,78	0,39	0,983	,778	Student	6,373	<.001	0,983
Repetiții	2,34	0,59	2,58	0,72	0,917	,005	Wilcoxon	133,5	0,024	0,462
Structuri condiționale	1,34	1,19	2,44	0,83	0,933	,017	Wilcoxon	52	<.001	0,815

În cele ce urmează, sunt examinate rezultatele testului **cCTt adaptat și revizuit**. Și în acest caz, scorul total al testului a fost împărțit la numărul de activități, astfel încât media scorului total al testului pe activitate a fost utilizată pentru prelucrarea statistică. Media scorurilor post-test este cu 0,25 mai mare (tabelul 17) decât media scorurilor pre-test. Deoarece testul Shapiro-Wilk arată o abatere de la distribuția normală, a fost utilizat testul Wilcoxon, care arată o diferență semnificativă din punct de vedere statistic între pre-test și post-test. Valoarea d-Cohen în acest caz este de 0,913, indicând o mărime a efectului ridicată.

Ca în cazul testului AlgoPaint, am analizat rezultatele pre- și post-test pentru testul cCTt pentru diferite tipuri de sarcini. Pentru toate cele trei tipuri de sarcini, testul Shapiro-Wilk a indicat o distribuție anormală, astfel încât am utilizat un test Wilcoxon. Rezultatele arată că pentru toate cele trei tipuri de sarcini există o diferență semnificativă între rezultatele pre-test și post-test, iar mărimea efectului este ridicată.

Tabelul 17. Compararea rezultatelor pre-test și post-test ale testului cCTt

cCTt	Pre-test		Post-test		Test normalitate Shapiro-Wilk		Test	W (statistics)	p	Mărimea efectului
	Media	Abater e	Media	Abat ere	W	p	Wilcoxon			

Media pe activități a scorului total al testului	0,71	0,24	0,96	0,11	0,932	,015	Wil-coxon	26,000	<.001	0,913
Instrucțiuni simple	0,85	0,23	0,97	0,11	0,680	<.001	Wil-coxon	10,000	,011	0,780
Repetiții	0,61	0,32	0,96	0,13	0,832	<.001	Wil-coxon	11,000	<.001	0,937
Structuri condiționale	0,50	0,50	0,90	0,29	0,702	<.001	Wil-coxon	10,000	<.001	0,895

H3. Prin utilizarea roboților educaționali pentru activități STEM, programul poate ajuta, de asemenea, preșcolarii să învețe concepte de programare mai puțin familiare pentru ei în mediul preșcolar.

În cele ce urmează, vom compara dificultatea a trei concepte de programare (structură simplă, structură iterativă și structură condiționată) pentru care am aplicat o analiză a varianței pe eșantioane grupate.

Începem prin a studia rezultatele pretestării colectate cu ajutorul testului AlgoPaint. Deoarece testul Mauchly arată că sfericitatea nu este îndeplinită ($\chi^2(2) = 10.470$, $p = .005$), am apelat la corecția Greenhouse-Geisser. Rezultatul arată că există o diferență semnificativă din punct de vedere statistic între rezultatele celor 3 concepte de programare ($F(1.626) = 20.321$, $p < .001$). Analiza Holm Post hoc arată că structura condițională este semnificativ mai dificilă din punct de vedere statistic decât instrucțiunile simple și structura iterativă (tabelul 18).

Tabelul 18. Rezultatele pre-testelor Algotaint pentru diferite tipuri de activități

	Media	Abatere		Media	Abatere	t	p
Instrucțiuni simple	2,04	0,68	Structuri iterative	2,34	0,59	-1,820	,072
Instrucțiuni simple	2.04	0,68	Structuri condiționale	1,34	1.19	4,381	<.001
Structuri	2,34	0,59	Structuri	1,34	1.19	6,201	<.001

iterative			condiționale				
-----------	--	--	--------------	--	--	--	--

În continuare, sunt analizate rezultatele *post-test* colectate prin testul *AlgoPaint*.

Deoarece conform testului Mauchly, sfericitatea nu este îndeplinită nici pentru *post-test* ($\chi^2(2) = 5,584$, $p = 0,061$), am folosit din nou corecția Greenhouse-Geisser. Rezultatele obținute arată o diferență semnificativă din punct de vedere statistic în rezultatele pentru conceptele de programare ($F(1,769) = 4,593$, $p < 0,05$).

Analiza Holm Post hoc arată că structura condiționată ($M = 2,440$, $SD = 0,835$) este semnificativ mai dificilă decât instrucțiunile simple, dar nu există nicio diferență semnificativă din punct de vedere statistic între structura condiționată și structura iterativă (tabelul 19). Acest rezultat arată că intervenția i-a ajutat pe copii să înțeleagă utilizarea structurii condiționale.

Tabelul 19. Rezultatele *post-testului* Algopaint pentru diferite tipuri de activități

	Media	Abatere		Media	Abatere	t	p
Instrucțiuni simple	2,78	0,39	Structuri iterative	2,58	0,72	1,808	,149
Instrucțiuni simple	2,78	0,39	Structuri condiționale	2,44	0,83	3,011	,010
Structuri iterative	2,58	0,72	Structuri condiționale	2,44	0,83	1,203	,232

La fel ca în cazurile de mai sus, pentru testul *cCTt adaptat*, am comparat, de asemenea, rezultatele *pre-testului* obținute pentru cele trei concepte de programare utilizând o analiză a varianței pe eșantioane grupate. Deoarece sfericitatea nu a fost îndeplinită prin testul Mauchly ($\chi^2(2) = 8,170$, $p=0,017$), a fost aplicată o corecție Greenhouse-Geisser. Rezultatele arată o diferență semnificativă între rezultatele referitoare la diferitele concepte de programare ($F(1,688) = 15,928$, $p < 0,001$). Analiza Holm Post hoc (tabelul 20) arată că instrucțiunile simple sunt semnificativ mai ușoare decât structura condițională și structura iterativă. Cu toate acestea, nu există nicio diferență semnificativă din punct de vedere statistic între structura iterativă și structura condițională.

Tabelul 20. Rezultatele pretestare ale testului cCT pentru diferite tipuri de sarcini

			Media	Abatere		Media	Abatere	t	p
Instrucțiuni simple			0.858	0,234	Structuri iterative	0,619	0,328	3,698	<.001
Instrucțiuni simple			0,858	0,234	Structuri condiționale	0,500	0,506	5,542	<.001
Structuri iterative			0,619	0,328	Structuri condiționale	0,500	0,506	1,844	,069

Am efectuat aceleași proceduri statistice în cadrul *post-testului cCTt adaptat* ca și în cadrul pre-testului, și anume am comparat rezultatele obținute pentru cele trei concepte de programare utilizând o analiză a varianței pe eșantioane grupate. Deoarece sfericitatea nu a fost îndeplinită conform testului Mauchly ($\chi^2(2) = 33,314$, $p < 0,001$), s-a aplicat corecția Greenhouse-Geisser. Rezultatele nu arată nicio diferență semnificativă între rezultatele diferitelor concepte de programare ($F(1,278) = 2,091$, $p = 0,150$).

Rezultatele anterioare arată că programul de intervenție i-a făcut pe copii să înțeleagă structura condițională, care inițial s-a dovedit dificilă, și că diferențele dintre rezultatele obținute în diferitele tipuri de activități au fost mai mici decât la post-test.

7. CONCLUZII ȘI POSIBILITĂȚI ULTERIOARE DE STUDIU

Teza mea este o prezentare a cercetării mele și a rezultatelor cercetării mele privind gândirea copiilor preșcolari. Mai exact, am investigat abilitățile de gândire computațională și potențialul lor de dezvoltare la copiii preșcolari.

1. Chestionare:

Deoarece ne-am propus să dezvoltăm activități legate de domeniul tehnic al activităților STEM, am elaborat mai întâi un chestionar privind aceste activități, care a fost completat de 85 de studenți care studiază la secția de Pedagogia Învățământului Primar.

Răspunsurile la întrebări au arătat că studenții integrează diferite domenii de experiență în practica lor didactică, ceea ce este, de asemenea, o așteptare curriculară. De asemenea, s-a constatat că au avut cel mai mare succes în combinarea activităților de matematică și de științe ale naturii, ceea ce este interesant având în vedere că ambele domenii fac parte din activitățile STEM. O constatare surprinzătoare a fost că mai mult de jumătate dintre studenți nu erau familiarizați cu termenul STEM și, prin urmare, nu aveau nicio experiență personală cu acest tip de activitate. Cu toate acestea, majoritatea dintre ei cred că

activitățile STEM încurajează copiii să gândească creativ în domeniul științei. Potrivit studenților, activitățile STEM dezvoltă o serie de competențe și abilități cognitive, emoționale, sociale și fizice, cum ar fi gândirea creativă și exploratorie, munca în echipă, precum și independența, explorarea senzorială, abilitățile motorii fine și grosiere. În ceea ce privește potențialul activităților STEM în grădiniță, majoritatea au subliniat că acestea le permit copiilor să dobândească experiență practică, să își îmbogățească cunoștințele despre mediul tehnic și natural și să ofere emoții pozitive și o motivație ridicată pentru diferite domenii ale științei.

A doua parte a chestionarului se referă la cunoștințele tehnologice și aplicarea lor în grădiniță, precum și la posibilitățile de dezvoltare a gândirii algoritmice/computaționale. În plus față de cei 85 de studenți, 115 cadre didactice din învățământul preșcolar au completat această parte a chestionarului. A existat un consens în rândul studenților și al cadrelor didactice din învățământul preșcolar conform căruia integrarea cunoștințelor tehnice în activitățile preșcolare ar dezvolta atitudinea pozitivă a copiilor față de acest domeniu. În răspunsurile studenților, cât și ale educatoarelor, lipsa unei pregătiri metodologice adecvate și lipsa unor instrumente didactice adecvate au fost identificate drept principalele obstacole în planificarea și implementarea acestor activități. În ceea ce privește gândirea algoritmică/computațională, atât studenții, cât și cadrele didactice au considerat importantă introducerea dezvoltării acestor abilități în grădiniță. Cu toate acestea, știau pre puțin despre modul în care să le dezvolte, roboții educaționali ca opțiuni de dezvoltare fiind menționați doar de câțiva dintre ei, niciunul dintre studenți nu folosisese vreodată astfel de dispozitive în practica lor de grădiniță și doar patru dintre profesorii de grădiniță au declarat că au folosit roboți de podea în activitățile lor de grădiniță. Rezultatele evidențiază necesitatea unor cursuri și ateliere de formare menite să integreze cunoștințele tehnologice în practica din grădiniță.

2. *Caietul de teste AlgoPaint:*

Pentru a evalua gândirea algoritmică și computațională a copiilor preșcolari, am dezvoltat un instrument cu hârtie și creion care nu necesită utilizarea unui computer sau a unui dispozitiv tehnologic. După testarea pilot a instrumentului de evaluare cu 56 de copii de grădiniță (5-6 ani) de către 11 educatoare, am trimis testul spre evaluare profesorilor universitari din diferite țări care sunt specialiști în domeniu. Pe baza rezultatelor educatoarelor, a copiilor și a opiniilor experților, testul a fost considerat adecvat și potrivit pentru evaluarea abilităților de gândire algoritmică și computațională ale grupeii de vârstă preșcolară. Încorporând sugestiile și opiniile cadrelor didactice de grădiniță și ale

profesorilor universitari cu experiență în dezvoltarea gândirii algoritmice și computaționale, am revizuit și îmbunătățit broșura de testare și am creat astfel Instrumentul de evaluare a gândirii computaționale AlgoPaint. Acest instrument este de nișă și poate fi utilizat în context preșcolar în orice țară, doar prin traducerea instrucțiunilor.

3. *Program de dezvoltare:*

Testarea prealabilă a planului de intervenție elaborat, deși a fost efectuată cu un grup mic de copii, s-a dovedit a fi un studiu de impact adecvat pe baza rezultatelor copiilor. Erorile detectate în timpul pre-testării au fost corectate, iar sarcinile care s-au dovedit dificile au fost revizuite. Datele pre- și post-test au arătat că testul AlgoPaint a fost sensibil la modificările intervenției, programul având un efect pozitiv asupra scorurilor post-test. Rezultatele au arătat că dezvoltarea gândirii computaționale poate fi introdusă la copiii preșcolari prin programe specifice. Studiul nostru de impact a arătat că roboții educativi de podea pot fi un instrument eficient pentru dezvoltarea gândirii computaționale la copiii preșcolari.

Limitările studiului:

Am identificat unele limitări ale cercetării, care sunt rezumate mai jos:

- Datorită numărului relativ mic de studenți și educatoare care au participat la sondajul prin chestionar (85 de elevi și 115 educatoare), rezultatele noastre nu pot fi generalizate la sistemul educațional din România. Cu toate acestea, în ceea ce privește chestionarul, ne-am dat seama că au fost adresate întrebări relativ puține cu privire la roboții educativi.
- Prin absența unui grup de control ar exista posibilitatea să excludem dezvoltarea naturală ca un posibil factor de influență în dezvoltarea intervenției. Cu toate acestea, considerăm că, întrucât dezvoltarea gândirii algoritmice și computaționale nu face parte din programul de grădiniță, nu abordăm în mod specific dezvoltarea direcționată a competențelor în acest domeniu, iar natura inovatoare a concepției intervenției și a instrumentului utilizat poate explica diferențele dintre rezultatele.
- În ceea ce privește instrumentele de sondaj, din cauza utilizării incomplete a testului cCTt, nu putem considera versiunea noastră adaptată drept un comparator validat.
- doar 41 de copii de grădiniță au participat la activitate, deci un eșantion relativ redus.

Inventarierea lacunelor oferă oportunități pentru îmbunătățiri suplimentare. În cazul nostru, apar următoarele posibilități:

- să punem în aplicare procedura de validare a broșurii de testare AlgoPaint pe un nou eșantion, utilizând testul cCT complet ca instrument de evaluare validat.
- dezvoltarea unei versiuni diferite a broșurii de testare AlgoPaint cu exerciții/itemi care ar inhiba efectul de transfer.

BIBLIOGRAFIE

1. Ackermann, E. (2001). Piaget's Constructivism, Papert's Constructionism: What's the Difference?
<http://learning.media.mit.edu/content/publications/EA.Piaget%20%20Papert.pdf>
2. Aho, V. A. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832–835. Oxford University Press.
3. Aknai, D., & Fehér, P. (2021). Barriers and challenges of the integration of robots in K-12 classrooms [Conference presentation]. *ECER 2021*, Geneva, Switzerland (online). <https://eera-ecer.de/ecer-programmes/conference/26/contribution/51492>
4. Aknai, D., & Fehér, P. (2022). Robotok alkalmazásának legújabb eredményei az általános iskolában – nemzetközi kitekintés. In G. Molnár & E. Tóth (Eds.), *Új kutatások a neveléstudományokban 2021– A neveléstudomány válaszai a jövő kihívásaira* (pp. 149–163). Szegedi Tudományegyetem Neveléstudományi Intézet, Magyar Tudományos Akadémia Pedagógiai Tudományos Bizottsága.
5. Aleksieva, L., Mirtschewa, I., & Radeva, S. (2021). Preschool teachers' knowledge, perspectives, and practices in STEM education: An interview study. *Mathematics and Informatics*, 64(6), 617–633
6. Algolittle. (2020). Algorithmic thinking skills through play-based learning for future's code literates. Erasmus+ project, 2020-1-TR01-KA203-092333.
7. Alimisis, D., & Kynigos, C. (2009). Constructionism and robotics in education: Teacher education on robotics-enhanced constructivist pedagogical methods.
8. Ananiadou, K., & Magdolean, C. (2009). 21st century skills and competences for new millennium learners in OECD countries. *OECD Education Working Papers*, No. 41.
9. Ari, F., Arslan-Ari, I., & Vasconcelos, L. (2022). Early childhood preservice teachers' perceptions of computer science, gender stereotypes, and coding in early childhood education. *TechTrends*, 66(3), 539–546. <https://doi.org/10.1007/s11528-022-00725-w>
10. Aubrey, C., & Dahl, S. (2014). The confidence and competence in information and communication technologies of practitioners, parents, and young children in the early years foundation stage. *Early Years*, 34(1), 94–108. <https://doi.org/10.1080/09575146.2013.792789>
11. Avcı, C., & Deniz, M. N. (2022). Computational thinking: Early childhood teachers' and prospective teachers' preconceptions and self-efficacy. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-022-11078-5>

12. Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, 29.
13. Balanskat, A., & Engelhardt, K. (2015). Computing our future. Computer programming and coding priorities, school curricula, and initiatives across Europe. *European Schoolnet*, 4–7.
14. Baltsavias, A., & Kyridis, A. (2020). Preschool teachers' perspectives on the importance of STEM education in Greek preschool education. *Journal of Education and Practice*, 11(14), 1–10. <https://doi.org/10.7176/JEP/11-14-01>
15. Bay, D. N. (2022). The perspective of preschool teachers on the use of digital technology. *Southeast Asia Early Childhood Journal*, 11(2), 87–111. <https://doi.org/10.37134/saecj.vol11.2.6.2022>
16. Bálint-Svella, É. K., & Zsoldos-Marchis, I. (2023). Óvodapedagógusok véleménye a technológiai területhez kapcsolódó képességek fejlesztéséről az óvodában. *Magyar Pedagógia*, 123(4). <https://doi.org/10.14232/mped.2023.4.209>
17. Bálint-Svella, É. K., & Zsoldos-Marchis, I. (2022). Pre-service teachers' knowledge and opinion about STEM education in preschool. *Studia Universitatis Psychologia-Paedagogia*, LXVII(1), 113–123. <https://doi.org/10.24193/subbypsyped.2022.1.07>
18. Bálint-Svella, É. K., & Zsoldos-Marchis, I. (2022). Preservice teachers' opinion about developing computational thinking in preschool. *Pedacta*, 12(1), 7–15. <https://doi.org/10.24193/PedActa.12.1.2>
19. Bálint-Svella, É. K. (2023). Az algoritmikus gondolkodás fejlesztése oktatási robotokkal óvodában: A fejlesztési program pilot tesztelése. *Pedacta*, 13(1), 1–8. <https://doi.org/10.24193/PedActa.13.1.1>
20. Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. <https://doi.org/10.4324/9781315398945>
21. Bers, M. U., Flannery, L., Kazakoff, R. E., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>
22. Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, 138, 130–145. <https://doi.org/10.1016/j.compedu.2019.04.013>

23. Bers, M. U., & Kazakoff, E. R. (2019). Engaging preschoolers in computational thinking through robotics. *Frontiers in Psychology*, 10, 2398. <https://doi.org/10.3389/fpsyg.2019.02398>
24. Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education: Implications for policy and practice. *JRC Research Reports*, JRC104188, Joint Research Centre. <https://doi.org/10.2791/792158>
25. Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, BC, Canada, 1. http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
26. Brennenman, K. (2011). Assessment for preschool science learning and learning environments. *Early Childhood Research & Practice*, 13(1).
27. Brown, T. T., & Jernigan, L. T. (2012). Brain development during the preschool years. *Neuropsychology Review*, 22, 313–333. Springer. <https://doi.org/10.1007/s11065-012-9214-1>
28. Bryant, E., R, Sutner, K., Stehlik J., M. (2010). *Introductory Computer Science Education, at Carnegie Mellon University: A Deans' Perspective*, CMU-CS-10-140
29. Bruner, J. S. (1966). *Toward a theory of instruction*. Harvard University Press.
30. Çakir, R., Korkmaz, Ö., Idil, Ö., Erdogmus, F., & Ugur, M. (2021). The effect of robotic coding education on preschoolers' problem solving and creative thinking skills. *Thinking Skills and Creativity*, 40. <https://doi.org/10.1016/j.tsc.2021.100812>
31. Carale, L. R., & Campo, P. C. (2003). *Concept development in Filipino children: The circulatory system*. University of the Philippines, National Institute of Science and Mathematics Education.
32. Castro, A., Aguilera, C., Yang, W., & Urrutia, B. (2024). High-capacity robots in early education: Developing computational thinking with a voice-controlled collaborative robot. *Educ. Sci.*, 14(8), 856. <https://doi.org/10.3390/educsci14080856>
33. Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education*, 22(4), 711–722.
34. Chaparro, L. C. D. (2020). *Algorithmic thinking: Central concepts, elements and pedagogical considerations* [Dissertation].

35. Chen, Y. L., Huang, L. F., & Wu, P. C. (2021). Preservice preschool teachers' self-efficacy in and need for STEM education professional development: STEM pedagogical belief as a mediator. *Early Childhood Education Journal*, 49, 137–147. <https://doi.org/10.1007/s10643-020-01055-3>
36. Ching, Y.-H., Hsu, Y.-Ch., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, 62(6), 563–573. <https://doi.org/10.1007/s11528-018-0292-7>
37. Clements, D. H., & Sarama, J. (2011). Early childhood mathematics intervention. *Science*, 333(6045), 968–970. <https://doi.org/10.1126/science.1204537>
38. Conezio, K., & French, L. (2002). Science in the preschool classroom: Capitalizing on children's fascination with the everyday world to foster language and literacy development. *Young Children*, 57, 12–18.
39. CSTA & ISTE. (2011). *Operational definition of computational thinking for K-12 education*. <http://www.iste.org/docs/pdfs/Operational-Definition-of-ComputationalThinking.pdf>
40. Curzon, P., Dorling, M., Selby, C., & Woollard, J. (2014). Developing computational thinking in the classroom: A framework. *Computing at School*. <http://stuartfrost.me/wp-content/uploads/2014/09/0.-Developing-ComputationalThinking-In-The-Classrooma-Framework.pdf>
41. De Bono, E. (1967). *The use of lateral thinking*.
42. DeJarnette, N. K. (2018). Implementing STEAM in the early childhood classroom. *European Journal of STEM Education*, 3(3), 18. <https://doi.org/10.20897/ejsteme/3878>
43. Di Lieto, M. C., Inguaggiato, E., Castro, E., Cecchi, F., Cioni, G., Dell'Omo, M., Laschi, C., Pecini, C., Santerini, G., Sgandurra, G., Dario, P. (2017). Educational robotics intervention on executive functions in preschool children: A pilot study. *Computers in Human Behavior*, 71, 16–23. <https://doi.org/10.1016/j.chb.2017.01.018>
44. Driscoll, A., & Nagel, G. N. (2005). *Early childhood education: Birth - 8: The world of children, families, and educators* (3rd ed.). MyLab School Edition.
45. Duncan, G. J., Dowsett, C. J., Claessens, A., Magnuson, K., Huston, A. C., Klebanov, P., Pagani, L. S., Feinstein, L., Engel, M., Brooks-Gunn, J., Sexton, H., Duckworth, K., & Japel, C. (2007). School readiness and later achievement. *Developmental Psychology*, 43(6), 1428–1446. <https://doi.org/10.1037/0012-1649.43.6.1428>
46. Davidov, V. V. (1992). The genesis and development of the individual in childhood. *Questions of Psychology*, 1, 22–33.

47. Edwards, S., Mantilla, A., Grieshaber, S., Nuttall, J., & Wood, E. (2020). Converged play characteristics for early childhood education: Multi-modal, global-local, and traditional-digital. *Oxford Review of Education*, 46(5), 637–660. <https://doi.org/10.1080/03054985.2020.1750358>
48. El-Hamamsy, L., Zapata-Cáceres, M., Martín Barroso, E., Mondada, F., Dehler Zufferey, J., & Bruno, B. (2022). The competent computational thinking test (cCTt): Development and validation of an unplugged computational thinking test for upper primary school. *Journal of Educational Computing Research*, 60(7). <https://doi.org/10.1177/07356331221081753>
49. Elkonin, B. D. (1999). *Game-psychology*. Moscow: Vlosos.
50. English, L. D. (2016). STEM education K-12: Perspectives on integration. *International Journal of STEM Education*, 3(1), art. 3. <https://doi.org/10.1186/s40594-016-0036-1>
51. Ferragina, P., & Luccio, F. (2018). Computational thinking. In *Algorithms and coding* (pp. 11-12). https://doi.org/10.1007/978-3-319-97940-3_3
52. Fosnot, C. T. (1996). *Constructivism: Theory, perspectives, and practice*. Teachers College Press.
53. Fridberg, M., Redfors, A., Greca, I. M., & García Terceño, E. M. (2022). Spanish and Swedish teachers' perspective of teaching STEM and robotics in preschool – Results from the botSTEM project. *International Journal of Technology and Design Education*. <https://doi.org/10.1007/s10798-021-09717-y>
54. Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. In R. T. Mittermeir (Ed.), *Informatics education – The bridge between using and understanding computers* (Lecture Notes in Computer Science, vol. 4226, pp. 159–168). Springer. https://doi.org/10.1007/11915355_15
55. Futschek, G., & Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. In *Proceedings of the 5th ISSEP, informatics in schools—Contributing to 21st century education* (pp. 155–164). Springer. https://doi.org/10.1007/978-3-642-24722-4_14
56. Gallant, G. (2002). Professional development for web-based teaching: Overcoming innocence and resistance. *New Directions for Adult and Continuing Education*, 88, 69–78. <https://doi.org/10.1002/ace.8807>
57. Gadzikowski, A. (2019). Planting the seeds of computational thinking in early childhood.

58. Glasersfeld, E. (1995). *Radical constructivism: A way of knowing and learning*. Falmer Press.
59. Hall, J. A., & McCormick, K. I. (2022). “My cars don’t drive themselves”: Preschoolers’ guided play experiences with button-operated robots. *TechTrends*, 66(3), 510–526. <https://doi.org/10.1007/s11528-022-00727-8>
60. Hódi, Á., Tóth, E., B. Németh, M., & Fáyine Dombi, A. (2019). Óvodások IKT-használata otthon – Szülői minta és szerepvállalás. *Neveléstudomány*, 6(2), 22–41. <https://doi.org/10.21549/NTNY.26.2019.2>
61. Hmelo-Silver, C. E., & Barrows, H. S. (2006). Goals and strategies of a problem-based learning facilitator. *Interdisciplinary Journal of Problem-based Learning*, 1, 21–39. <http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1004&context=ijpbl>
62. Hromkovic, J., Kohn, T., Komm, D., & Serafini, G. (2017). Algorithmic thinking from the start. *Bulletin of the European Association for Theoretical Computer Science*, 121. <http://bulletin.eatcs.org/index.php/beatcs/article/view/478>
63. Ishii, H. (2007). Tangible user interfaces. In *The Human-Computer Interaction Handbook* (2nd ed.).
64. Jack, C., & Higgins, S. (2019a). What is educational technology and how is it being used to support teaching and learning in the early years? *International Journal of Early Years Education*, 27(3), 222–237. <https://doi.org/10.1080/09669760.2018.1504754>
65. Jack, Ch., & Higgins, S. (2019b). Embedding educational technologies in early years education. *Research in Learning Technology*, 27, 2033. <https://doi.org/10.25304/rlt.v27.2033>
66. Jipson, J. L., Callanan, M. A., Schultz, G., & Hurst, A. (2014). Scientists not sponges: STEM interest and inquiry in early childhood. In *Ensuring STEM Literacy: A National Conference on STEM Education and Public Outreach* (Vol. 483, pp. 149–156).
67. József, I. (2012). *Fejlődéslélektan*. Kaposvári Egyetem
68. K4K. (2020). Teachers about STEM education on the preschool level. *Kitchen Lab for Kids Erasmus+ Project*.
69. Kalyenci, D., Metin, Ş., & Başaran, M. (2022). Test for assessing coding skills in early childhood. *International Journal of Technology and Design Education*, 27, 4685–4708. <https://doi.org/10.1007/s10639-021-10803-w>

70. Kara, N., & Cagiltay, K. (2017). In-service preschool teachers' thoughts about technology and technology use in early educational settings. *Contemporary Educational Technology*, 8(2), 119–141. <https://doi.org/10.30935/cedtech/6191>
71. Kara, N., & Cagiltay, K. (2020). Smart toys for preschool children: A design and development research. *Electronic Commerce Research and Applications*, 39. <https://doi.org/10.1016/j.elerap.2019.100909>
72. Karademir, A., & Yildirim, B. (2021). A different perspective on preschool STEM education: STEM education and views on engineering. *Journal of Turkish Science Education*, 18(3), 338–350. <https://doi.org/10.36681/tused.2021.77>
73. Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2012). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41, 245–255. <https://doi.org/10.1007/s10643-012-0554-5>
74. Keengwe, J., & Onchwari, G. (2009). Technology and early childhood education: A technology integration professional development model for practicing teachers. *Early Childhood Education Journal*, 37, 209–218. <https://doi.org/10.1007/s10643-009-0341-0>
75. Kourti, Z., Michalakopoulos, Ch-A., Bagos, P. G., & Efrosyni-Alkisti Paraskevopoulou-Kollia, E-A. (2023). Computational thinking in preschool age: A case study in Greece. *Education Sciences*, 13(2), 1–13. <https://doi.org/10.3390/educsci13020157>
76. Kovácsné Pusztai, K. (2016). *Számítógépes gondolkodás a felsőoktatásban*. ELTE IK.
77. Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal*, 47, 709–716. <https://doi.org/10.1007/s10643-019-00967-z>
78. Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37. <https://doi.org/10.1145/1929887.1929902>
79. LEGO Education. (2019). *The impact of early STEM education on children's learning outcomes*.
80. Levy, S. T., & Mioduser, D. (2008). Does it “want” or “was it programmed to...”? Kindergarten children's explanations of an autonomous robot's adaptive functioning. *International Journal of Technology and Design Education*, 18, 337–359. <https://doi.org/10.1007/s10798-007-9032-6>

81. Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research*, 3(1), 1–18. <https://doi.org/10.1007/s41979-020-00030-2>
82. Lippard, C., Lamm, M. H., Tank, K. M., & Choi, J. Y. (2019). Pre-engineering thinking and the engineering habits of mind in preschool classrooms. *Early Childhood Education Journal*, 47(2), 187–198. <https://doi.org/10.1007/s10643-018-0898-6>
83. Lindberg, R. S. N., Laine, T. H., & Haaranen, L. (2018). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology (BJET)*, 50(4), 1979–1995. <https://doi.org/10.1111/bjet.12685>
84. Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, 28(6), 651–676. <https://doi.org/10.1007/s10956-019-09799-3>
85. Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In *Proceedings of the working group reports of the conference on innovation & technology in computer science education (ITiCSE-WGR '14)* (pp. 1–29). ACM. <https://doi.org/10.1145/2713609.2713610>
86. McClure, E. R., Guernsey, L., Clements, D. H., Bales, S. N., Nichols, J., Kendall-Taylor, N., & Levine, M. H. (2017). *STEM starts early: Grounding science, technology, engineering, and math education in early childhood*. Joan Ganz Cooney Center at Sesame Workshop.
87. Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018). Unravelling the cognition of coding in 3-to-6-year olds: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. In *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER'18* (pp. 133–141). <https://doi.org/10.1145/3230977.3230984>
88. Meeteren, V. B., & Zan, B. (2010). Revealing the Work of Young Engineers in Early Childhood Education. SEED (STEM in Early Education and Development) Collected Paper, Conference.
89. Metin, O. (2020). Cognitive Networking for Next Generation of Cellular Communication Systems. Doctoral Thesis.

90. Ministerul Educației Naționale. (2019). *Koragyermekkorai Nevelés Curriculumuma*.
91. Mioduser, D., & Levy, S. T. (2010). Making Sense by Building Sense: Kindergarten Children's Construction and Understanding of Adaptive Robot Behaviors. *International Journal of Computers for Mathematical Learning*, 15, 99–127. <https://doi.org/10.1007/s10758-010-9163-9>
92. Mittermeir, T. R. (2013). Algorithmics for Preschoolers – A Contradiction? *Creative Education*, 4(9), 557–562. <https://doi.org/10.4236/ce.2013.49081>
93. Nagy, E. (2020). Robotok az oktatási-nevelési folyamatokban. *Képzés és Gyakorlat*, 18(3–4), 176–186. <https://doi.org/10.17165/TP.2020.3-4.18>
94. National Science Foundation. (2019). Women, Minorities, and Persons with Disabilities in Science and Engineering.
95. National Research Council. (2014). *STEM Integration in K-12 Education: Status, Prospects, and an Agenda for Research*. Washington, DC: The National Academies Press. <https://doi.org/10.17226/18612>
96. Nardelli, E. (2019). Do We Really Need Computational Thinking? *Communications of the ACM*, 62(2), 32–35. <https://doi.org/10.1145/3231587>
97. NGSS Lead States. (2013). *Next Generation Science Standards: For States, by States*. Washington, DC: The National Academies Press. <https://doi.org/10.17226/18290>
98. Németh, M., Hódi, Á., Juhász, F., Sárík, A., & Tóth, E. (2021). Szülők véleménye az óvodáskorú gyermekek IKT-eszköz használatának negatív és pozitív hatásairól. *Gyermeknevelés Tudományos Folyóirat*, 9(1), 8–38. <https://doi.org/10.31074/gyntf.2021.1.8.38>
99. Ortiz-Revilla, J., Greca, I. M., & Meneses-Villagrà, J. A. (2021). Effects of an Integrated STEAM Approach on the Development of Competence in Primary Education Students. *Journal for the Study of Education and Development*, 1–33. <https://doi.org/10.1080/02103702.2021.1925473>
100. Otterborn, A., Schönborn, K. J., & Hultén, M. (2020). Investigating Preschool Educators' Implementation of Computer Programming in Their Teaching Practice. *Early Childhood Education Journal*, 48, 253–262. <https://doi.org/10.1007/s10643-019-00976-y>
101. Palmér, H. (2017). Programming in Preschool—with a Focus on Learning Mathematics. *International Research in Early Childhood Education*, 8(1).

102. Pantoya, M. L., Aguirre-Munoz, Z., & Hunt, E. M. (2015). Developing an Engineering Identity in Early Childhood. *American Journal of Engineering Education*, 6(2), 61-68.
103. Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.
104. Papert, S. (1991). Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism*. (pp. 1–11). Norwood, NJ: Ablex Publishing Corporation
105. Papert, S. (1993). *The Children's Machine: Rethinking School in the Age of the Computer*. New York: Basic Books.
106. Passe, J. (1999). *Elementary School Curriculum* (2nd ed.). Boston: McGraw-Hill College.
107. Pawilen, G. T., & Yuzon, M. R. (2019). Planning a Science, Technology, Engineering, and Mathematics (STEM) Curriculum for Young Children: A Collaborative Project for Pre-service Teacher Education Students. *International Journal of Curriculum and Instruction*, 11(2), 130–146.
108. Pawilen, G. T., & Manzano, V. U. (2007). Integration of Science and Mathematics in the Grade I Curriculum. *Education Quarterly Journal*, College of Education, University of the Philippines, Diliman, 65(1), 4–18.
109. Phillips, A. C., Lewis, L. K., McEvoy, M. P., Galipeau, J., Glasziou, P., Moher, D., Tilson, J. K., & Williams, M. T. (2016). Development and Validation of the Guideline for Reporting Evidence-Based Practice Educational Interventions and Teaching (GREET). *BMC Medical Education*, 16(1), Article 237. <https://doi.org/10.1186/s12909-016-0759-1>
110. Piaget, J. (1952). *The Origins of Intelligence in Children*. New York: International Universities Press.
111. Piaget, J. (1970). *The Child's Conception of the World*.
112. Racsó, R. (2017). *Digitális átállás az oktatásban*. Iskolakultúra-könyvek 52. Gondolat Kiadó. <https://doi.org/10.17717/IQKONYV.Racsko.2017>
113. Relkin, E. (2018). Assessing Young Children's Computational Thinking Abilities (Master's Thesis). Retrieved from ProQuest Dissertations and Theses Database (UMI No. 10813994).
114. Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early

- Childhood Education. *Journal of Science Education and Technology*, 29, 482–498.
<https://doi.org/10.1007/s10956-020-09831-x>
115. Relkin, E., & Bers, M. (2021). TechCheck-k: A Measure of Computational Thinking for Kindergarten Children. In T. Klinger, C. Kollmitzer, & A. Pester (Eds.), *Proceedings of the 2021 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1709–1715). Institute of Electrical and Electronics Engineers.
<https://doi.org/10.1109/EDUCON46332.2021.945392>
 116. Resnick, M., & Siegel, D. (2015). A Different Approach to Coding. *International Journal of People-Oriented Programming*, 4(1), 1–4.
 117. Resnick, M. (2017). *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*. MIT Press.
 118. Rogers, X. (1972). *Theory of Recursive Functions and Effective Computability*. Mir.
 119. Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2018). Coding in K-8: International Trends in Teaching Elementary/Primary Computing. *TechTrends*, 63(3), 311–329. <https://doi.org/10.1007/s11528-018-0295-4>
 120. Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation. *Proceedings of EDULEARN15 Conference*, 2436–2444.
 121. Saçkes, M., Trundle, K. C., Bell, R. L., & O'Connell, A. A. (2010). The Influence of Early Science Experience in Kindergarten on Children's Immediate and Later Science Achievement: Evidence from the Early Childhood Longitudinal Study. *Journal of Research in Science Teaching*. <https://doi.org/10.1002/tea.20395>
 122. Sadykova, O. V., & Il'bahtin, G. G. (2020). The Definition of Algorithmic Thinking. *Proceedings of the International Session on Factors of Regional Extensive Development (FRED 2019)* (pp. 419–422). Irkutsk State Transport University.
<https://doi.org/10.2991/fred-19.2020.85>
 123. Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In M. Khine (Ed.), *Computational thinking in the STEM disciplines* (pp. 127–146). Springer.
https://doi.org/10.1007/978-3-319-93566-9_8
 124. Selby, C., & Woollard, J. (2014). Computational thinking: The developing definition. *SIGCSE Proceedings*, 1-6.

125. Simmering, R. (2012). The development of visual working memory capacity during early childhood. *Journal of Experimental Child Psychology*, 111(4), 695-707. <https://doi.org/10.1016/j.jecp.2011.10.007>
126. Scharf, F., Winkler, T., & Herczeg, M. (2008). Tangicons: Algorithmic reasoning in a collaborative game for children in kindergarten and first class. In *IDC '08: Proceedings of the 7th International Conference on Interaction Design and Children* (pp. 242-249). <https://doi.org/10.1145/1463689.1463762>
127. Siu, K. W. M., & Lam, M. S. (2005). Early childhood technology education: A sociocultural perspective. *Early Childhood Education Journal*, 32, 353-358. <https://doi.org/10.1007/s10643-005-0003-9>
128. Stephen, C., & Edwards, S. (2018). *Young children playing and learning in a digital age: A cultural and critical perspective*. Routledge
129. Sternberg, R. J. (1986). *Intelligence applied: Understanding and increasing your intellectual skills*. Harper & Row.
130. Sullivan, A. M., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grades. *International Journal of Technology and Design Education*, 26, 3-20. <https://doi.org/10.1007/s10798-015-9304-5>
131. Sullivan, A. A., Bers, M. U., & Mihm, C. (2017). Imagining, playing, and coding with KIBO: Using robotics to foster computational thinking in young children. The Education University of Hong Kong, China.
132. Sung, J. (2022). Assessing young Korean children's computational thinking: A validation study of two measurements. *Education and Information Technologies*, 27, 12969–12997. <https://doi.org/10.1007/s10639-022-11137-x>
133. Szántó, S. (2002). Az algoritmikus gondolkodás fejlesztése általános iskolában. *Új pedagógiai szemle*, 52(5), 84-91.
134. Tai, R. H., Liu, C. H. Q., Maltese, V. A., & Fan, X. (2006). Planning early for careers in science. *Science*, 312(5777), 1143-1144. <https://doi.org/10.1126/science.1128690>
135. Tolman, M. N. (1995). *Discovering elementary science: Method, content, and problem-solving activities*. Allyn & Bacon.
136. Türk, A., & Akcanca, N. (2021). An example implementation of STEM in preschool education: Magnets. *Journal of Educational Leadership and Policy Studies*.

137. Uğraş, M., & Genç, Z. (2018). Investigating preschool teacher candidates' STEM teaching intention and the views about STEM education. *Bartın University Journal of Faculty of Education*, 7(2), 724-744. <https://doi.org/10.14686/buefad.408150>
138. Undheim, M. (2022). Children and teachers engaging together with digital technology in early childhood education and care institutions: A literature review. *European Early Childhood Education Research Journal*, 30(3), 472–489. <https://doi.org/10.1080/1350293X.2021.1971730>
139. Ültay, N., & Ültay, E. (2020). A comparative investigation of the views of preschool teachers and teacher candidates about STEM. *Journal of Science Learning*, 3(2), 67-78.
140. Voronina, V. L., Sergeeva, N. N., & Utyumova, A. E. (2016). Development of algorithm skills in preschool children. *Procedia - Social and Behavioral Sciences*, 233, 155–159. <https://doi.org/10.1016/j.sbspro.2016.10.176>
141. Vidal-Hall, C., Flewitt, R., & Wyse, D. (2020). Early childhood practitioner beliefs about digital media: Integrating technology into a child-centred classroom environment. *European Early Childhood Education Research Journal*, 28(2), 167–181. <https://doi.org/10.1080/1350293X.2020.1735727>
142. Vizner, M. Z. (2017). Big Robots for Little Kids: Investigating the Role of Scale in Early Childhood Robotics Kits. Thesis.
143. Vygotsky, L. S. (1934). *Thought and language*. MIT Press.
144. Vujičić, L., Jančec, L., & Mezak, J. (2021). Development of algorithmic thinking skills in early and preschool education. In L. Gómez Chova, A. López Martínez, & I. Candel Torres (Eds.), *13th International Conference on Education and New Learning Technologies* (pp. 8152–8161). IATED Academy. <https://doi.org/10.21125/edulearn.2021.1650>
145. Wadsworth, B. J. (1996). *Piaget's theory of cognitive and affective development*. Longman.
146. Wai, J., Lubinski, D., Benbow, C. P., & Steiger, J. H. (2010). Accomplishment in science, technology, engineering, and mathematics (STEM) and its relation to STEM educational dose: A 25-year longitudinal study. *Journal of Educational Psychology*, 102(4), 860–871. <https://doi.org/10.1037/a0019454>
147. Walsh, C., & Campbell, C. (2018). Introducing coding as a literacy on mobile devices in the early years. In *Mobile technologies in children's language and literacy*

- (pp. 51-66). Emerald Publishing Limited. <https://doi.org/10.1108/978-1-78714-879-620181004>
148. Watts, T. W., Duncan, G. J., Siegler, R. S., & Davis-Kean, P. E. (2014). What's past is prologue: Relations between early mathematics knowledge and high school achievement. *Educational Researcher*, 43(7), 352-360. <https://doi.org/10.3102/0013189X14553660>
149. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
150. Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>
151. Wing, J. (2011). Research notebook: Computational thinking—what and why? *The Link*. <https://www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why>. 10.1016/j.compedu.2019.103607
152. Wing, J. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2).
153. Wyeth, P., & Wyeth, G. (2008). Robot building for preschoolers. *Conference paper*, 124–135.
154. Yadav, A., Mayfeld, C., Zhou, N., Hambruch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1-16. <https://doi.org/10.1145/2576872>
155. Yang, W., Ng, D. T. K., & Gao, H. (2022). Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. *British Journal of Educational Technology*, 10, 1-25. <https://doi.org/10.1111/bjet.13215>
156. Yelland, N. (2017). Teaching and learning with tablets: A case study of twenty-first century skills and new learning. In N. Kucirkova & G. Falloon (Eds.), *Apps, technology and younger learners: International evidence for teaching* (pp. 57–72). Routledge.
157. Yu, J., & Roque, R. (2019). A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction*, 21, 17-36. <https://doi.org/10.1016/j.ijcci.2019.04.001>

158. Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. In *2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1905–1914). IEEE. <https://doi.org/10.1109/EDUCON45650.2020.9125368>
159. Zhang, L., & Nouri, J. J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>
160. Zhang, S., & Wong, G. K. W. (2023). Development and validation of a computational thinking test for lower primary school students. *Educational Technology Research and Development*. <https://doi.org/10.1007/s11423-023-10231-2>
161. Zsoldos-Marchis, I., & Ciascai, L. (2019). The opinion of primary and preschool pedagogy specialization students about the teaching approaches related with STEM/STEAM/STREAM education. *12th International Conference of Education Research and Innovation (ICERI2019)*, 7269-7275.
162. Zsoldos-Marchis, I., & Bálint-Svella, É. K. (2023). Development and preliminary testing of the AlgoPaint unplugged computational thinking assessment for preschool education. *Acta Didactica Napocensia*, 16(1), 32–50. <https://doi.org/10.24193/adn.16.1.3>