Developing computational thinking in preschool with educational robots

1. INTRODUCTION

Given my profession as a preschool teacher, it was an obvious choice for me to conduct my research with preschool children. This was important for me not only because I am around them every day and watch the way they work, but also because this age (3-6 years) is a particularly exciting period that is defined by substantial change.

According to Brown et al. (2012), preschool age is a period of intense mental development, during which many psychological abilities emerge and continue to develop and refine until the onset of young adulthood. Similarly, the brain development that occurs during this stage can be characterized as a period of “blossoming”, during which the most dynamic anatomical and psychological changes take place. In light of the above, I have chosen to focus my thesis on the assessment and development of thinking skills in preschool children. Of the various thinking skills, I chose to measure the ones that are currently being heavily researched. This prompted me to investigate the characteristics of algorithmic/computational thinking skills.

The planning of our daily lives is based on algorithms; it is therefore important to teach children from an early age to follow a logical path and adopt a systematic approach when solving a problem (Szántó, 2002). The term “computational thinking” was first introduced by two researchers, Jeannette Wing and Seymour Papert. As early as 1980, Papert (1980) – in his constructionist approach – emphasised the crucial role of social and emotional engagement in the utilisation of programming as an interdisciplinary instrument. In 2006, Wing stressed the pivotal role of computational thinking in the development of technical competences; additionally, he proposed the idea that computational thinking will emerge as the fourth essential skill in the 21st century, alongside literacy and numeracy. Computational thinking is a complex skillset that extends beyond using a computer. Computational thinking requires an understanding of problem-solving, systems design, and human behaviour (Wing, 2008). Bers (2018) defined computational thinking in preschool children as the ability to abstract computational behaviours and identify errors. Those who are critical of computational thinking (e.g. Mannila et al., 2014) argue that computer scientists are attempting to transform all children into software developers. This, however, is a gross misrepresentation of Wang’s position. The objective is not to teach software development; rather, it is to provide instruction in computing with the aim of equipping people with the necessary tools to navigate and comprehend our digital world. Despite the extensive research conducted on this subject, there is still no consensus on a single definition of computational thinking. Selby and Woollard (2014) define algorithmic thinking as an essential subset of computational thinking. While algorithmic thinking is concerned with the development of step-by-step procedures in solving specific problems, computational thinking involves a broader set of skills that go beyond specific algorithms.

The structure of my thesis is based on the chronological sequence of findings from previous research, with the objective of investigating and analysing the selected topic. The section that follows the introduction contains a literature review, which provides an explanation of key concepts, such as STEM activities and algorithmic/computational thinking. This is followed, within the same section, by a presentation of the assessment instruments for algorithmic/computational thinking, as well as an overview of the most frequently used methods and tools of development.

The third section presents three research projects. The first project discusses STEM activities: it focuses on the integration of technological skills, as well as the development of algorithmic and computational thinking skills. The questions on STEM activities were answered by 85 students enrolled in the Pedagogy of Primary and Preschool Education programme at the Babeș-Bolyai University (Bálint-Svella and Zsoldos-Marchis, 2022). In addition to the 85 university students, 115 practicing preschool teachers also responded to the questions on the inclusion of technological literacy and algorithmic and computational thinking (Bálint-Svella és Zsoldos-Marchis, 2022, Bálint-Svella and Zsoldos-Marchis, 2024). The aim of these surveys was to evaluate the awareness of practicing preschool teachers and those in training regarding the potential of utilising technology in STEM activities within the preschool context, as well as the competences that can be developed through such activities. This situation assessment was followed by the development, pre-testing and revision of the AlgoPaint Test of Computational Thinking (Zsoldos-Marchis és Balint-Svella, 2023). In the third study, we devised a development plan that employed the use of educational robots to facilitate the advancement of computational thinking in preschool-aged children (Bálint-Svella, 2023). The intervention programme was pre-tested, revised and subsequently implemented in three kindergarten groups.

It is not only the development of the assessment tool that is innovative, but also the intervention plan, which is based on educational robots, and targets the preschool age group. The development of educational robots is not a standard practice within the

Romanian education system. A variety of clubs and extracurricular activities offer opportunities to do this, with a particular focus on school-age children. As a wide range of skills and abilities develop during the preschool years, research indicates that this is also the period during which computational thinking skills begin to emerge. Our study was designed to address this research gap, as no existing examples could be found in the Romanian scholarly literature.

During my analysis, I found that floor robots can serve as an effective educational tool for children, as they tend to find them appealing. Another significant advantage of these playful tasks is the ability to verify the children's level of proficiency and receive immediate feedback on their performance. In this manner, robots not only provide an experience, but also develop the child. While doing the tasks, the children learn that it is possible to make mistakes, that they can be corrected, that it is possible to try and start again. The children have the opportunity to try repeatedly and to practice until the solution is correct, while accurately identifying the specific error committed. This is a very important experience, because already at this age they are able to approach tasks with a problem-solving focus, rather than measuring their performance against the endpoints of right or wrong, success or failure. Most of the planned tasks were organised in pairs or groups, so collaborative problem solving, mutual support, choosing the appropriate way and means of communication, cooperation and brainstorming were all part of the process.

My research was conducted with objectivity and applying a scientific approach, and my thesis covers the work of the last four years, presenting a small segment of preschool education in Romania.

2. THEORETICAL BACKGROUND

2.1. STEM activities in kindergarten

2.1.1. The presentation of STEM activities

STEM is an acronym consisting of the first four letters of four words which combine knowledge in science, technology, mathematics and engineering (English, 2016). STEM activities encompass the fields of science, technology, engineering and mathematics and incorporate this information into the existing knowledge base in a playful way at the preschool level (Pantoya et al., 2015, DeJarnette, 2018). Science, mathematics, technology and engineering are constantly intertwined in everyday contexts. They provide knowledge, skills and tools to improve the quality of human life. Over the past decade, there has been growing concern about the state of science, technology, engineering and mathematics education in the United States (English, 2016, National Research Council [NRC], 2014). Data show that only 16% of US high school students are proficient in mathematics, and the United States ranks 17th among industrialised nations in science. It is very important for everyone, especially young people, to have an adequate level of scientific and technological literacy in order to become a creative and active member of society (English 2016, NGSS lead states, 2013).

2.1.2. Using STEM activities in a preschool setting

Today, STEM (Science, Technology, Engineering and Mathematics) education has a very important role to play in meeting the challenges of the modern world. Not only are STEM skills essential for many jobs, but they are also essential in everyday life. Research has shown that children need to be involved in STEM activities from an early age, as the preschool years are a sensitive period for the development of basic thinking skills (Driscoll and Nagel, 2008). A longitudinal study by Wai et al. (2010) found that those exposed to STEM activities in their early years of life had high mathematical skills in adulthood. Children's early positive experiences with STEM are important for developing the skills they need to cope with life's challenges (Lippard et al., 2019), but these early experiences also have a significant impact on their performance at school (Watts et al., 2014). Early STEM education should be child-centred and problem-focused (Fridberg et al., 2022). It should be implemented primarily through hands-on activities that positively influence children's attitudes towards STEM (Ortiz-Revilla et al., 2021).

Introducing and practising STEM activities in preschool is crucial for children's development. A number of scientific studies have shown that these activities have significant benefits for the cognitive, social and emotional development of children. STEM activities develop children's problem-solving and critical thinking skills. One study shows that early STEM education contributes to an understanding of basic mathematical and scientific concepts, which leads to an advantage in later school performance (Clements and Sarama, 2011). Preschoolers who participate in such activities perform better in mathematics and science at school (Duncan et al, 2007). In addition, STEM activities encourage children to work in teams and to cooperate, as such activities often entail collaborative work, which develops social skills and emotional intelligence (Brenneman, 2011). For example, when children build a simple machine or carry out experiments together, they learn how to communicate effectively, share ideas and support each other. STEM activities stimulate creativity and innovation. Children are exposed to problems that require creative thinking for solving them, which helps children develop their future innovation skills (Resnick, 2017). Research has shown that children who receive early STEM education are more likely to choose STEM careers later in life, which is very important in the modern labour market (Tai et al., 2006). This is particularly important for gender and social equity, as early exposure to these areas can facilitate the reduction of the gender gaps in STEM fields (National Science Foundation, 2019).

In their research, Pantoya and colleagues (2015) developed a programme for children between the ages of 3 and 7 to help them develop an engineering identity. The researchers developed and tested an engineering storybook to increase children's knowledge of engineering, as well as their creativity and interest in engineering careers. The results showed that this storybook and the related activities can support children's understanding of STEM concepts and the development of their engineering identity. The authors suggest that more emphasis should be placed on engineering and STEM content in preschool and early school education to support children's future success in these areas.

In Romania, the curriculum for early childhood education (Ministerul Educației Naționale, 2019) includes several experience areas, of which natural sciences and mathematics can be considered part of STEM. The activities bring together several different experience areas. The development of skills and competences in engineering and technology as a part of STEM education is missing from the curriculum of early childhood education.

2.2. Algorithmic/computational thinking in preschool

Algorithms are part of our lives; we use algorithms in our daily activities. We carry out our daily activities according to specific algorithms that bring stability, security and order to our lives. The algorithm is a deterministic procedure that can be applied to any element of a class of symbolic inputs and that produces the corresponding symbolic output for each such input (Rogers, 1972). "Algorithmic thinking is a way of getting to a solution through clear definition of the steps" (Curzon et al, 2014, 2). More generally, "algorithmic thinking is a system of thinking methods that is necessary to build a sequence of obtaining intermediate results, planning the structure of actions and its implementation, leading to the achievement of the goal" (Sadykova and Il'bahtin, 2019, 421).

Regarding the notions of algorithmic thinking and computational thinking, Hromkovich and colleagues (2017) consider them to be two completely equivalent and interchangeable formulations of the same concept, despite the fact that they were introduced in different eras, since the notion of algorithmic thinking has been in use for several decades, while computational thinking is a much newer term. The authors believe that algorithmic thinking is rooted in the scientific core of computing, as indicated by Aho's (2012) definition: computational thinking can be seen as a thought process that is involved in formulating problems so that their solutions can be represented as computational steps and algorithms.

The concept of computational thinking is not entirely new: it has evolved over a long period of time. First mentioned by Seymour Papert (1980), the developer of the LOGO programming language, the term was popularised by Wing (2006), who introduced it as shorthand for "thinking like a computer scientist". Informally, computational thinking describes the mental activity of formulating a problem in order to find a computational solution. Computational thinking is not just about solving problems. It is also about formulating the problem. In fact, computational thinking is a process in which an individual develops a set of thinking strategies in order to approach a given problem. The main and highest level of computational thinking is abstraction. We use abstraction to identify patterns and also to highlight important common features while hiding irrelevant features between them. According to Wing, there are two fundamental aspects of computational thinking: creating abstractions (the forms of computational abstraction: algorithmic formulations and modularity) and implementing abstractions. Wing's (2006) article arguing that computational thinking is a universally applicable attitude and skill set, as important as literacy and numeracy, has sparked a major debate. Although there is no single agreed definition, Brennan and Resnick's (2012) is one of the most widely accepted. The authors break computational thinking into the following parts: computational concepts (the concepts most closely related to computing and programming, i.e. "sequences, loops, parallelism, events, etc."), computational practices (the strategies and practices needed to apply these concepts), and computational perspectives ("the perspectives designers form about the world around them and about themselves").

Selby and Woollard (2014) followed the development and evolution of Wing's definition of computational thinking, and reviewed the literature. According to their recommendation, computational thinking is a cognitive or thinking process that reflects the ability to think in abstractions, the ability to think in decomposition, the ability to think in algorithms, the ability to think in evaluations and generalisations. This definition includes only those terms for which there is a consensus in the literature.

According to Gadzikowski's (2019) research, computational thinking is generally a combination of four skill categories: pattern recognition, creating and using algorithms, decomposition, and understanding abstractions. Pattern recognition is the process of identifying, defining, extending and creating patterns; algorithms are a series of steps in solving a problem; decomposition is the process of breaking something down into its elementary parts; abstraction is the process of understanding an abstraction through generalisation, inference and other problem-solving thinking processes to imagine something that cannot be seen or touched. Although there is a growing recognition of the importance of computational thinking, its conceptual boundaries are not yet clear. There are various definitions. Computational thinking has undergone a major expansion in content since its conception in the 1950s, as it has evolved from its roots in computer science to a broader, more complex concept that focuses on the cognitive and metacognitive processes that occur when individuals process information. Thus, computational thinking is the ability to use computational concepts to formulate and solve problems.

Chaparro (2020) compared the concepts of computational and algorithmic thinking based on a review of the scholarly literature. According to him, the difference between the two concepts is that algorithmic thinking focuses on algorithms, while computational thinking focuses on computational methods. He summarises the results of previous research and includes algorithmic design, an ordered sequence of elementary operations designed to solve similar problems, as a key concept in computational thinking.

2.2.1. Assessing computational thinking in preschool

Despite the growing interest in the study of computational thinking, there is still a lack of research on how to teach and assess this area of thinking, particularly in young children (Rich et al., 2018). The assessment of computational thinking would provide useful feedback to teachers, students and researchers evaluating the effectiveness of educational programmes, curricula or interventions. Over the past two decades, many assessment tools have been developed to measure computational thinking, but few have focused on young children (between 4 and 9 years of age). Most of the previous work used interviews in their surveys or carried out project-based coding assessment. While interview and project-based surveys provide insights into children’s thinking, the format and time-consuming nature of these surveys make them unsuitable for use outside of a research setting.

According to Lee et al. (2011), “because CT is not evaluated by standardized testing, it is difficult in the current educational climate for teachers to teach CT concepts directly. (...) the field needs systematic assessment procedures” (p. 36). This, however, is a big challenge for researchers, especially as preschool children’s cognitive abilities are still limited. Of course, many measurement tools have been developed over the last two decades, but only a few of them have focused on young children, i.e., the 3-7 age group (El-Hamamsy, 2022, Marinus et al., 2018, Relkin, 2018, Relkin et al., 2020, Relkin and Bers, 2021, Zapata-Cáceres, 2020, Zhang and Wong, 2023).

Below, we present some of the assessment tools designed to measure young children’s computational thinking. Some of the measurement tools developed include tasks to be performed with educational robots (Marinus, 2018, Relkin, 2018).

Marinus and colleagues (2018) undertook the development of a standardised assessment tool to measure coding skills in children between the ages of 3 and 6. The test was developed for the Cubetto robot. The Cubetto is a simplified version of the LOGO turtle programming exercise developed by Seymour Papert. The tests were carried out on a sample of 18 children. At the task level, a lot of things were modified and simplified in comparison with what the Cubetto robot was originally capable of doing.

Relkin et al. (2018) developed another measuring tool for children aged 5 to 7, also using an educational floor robot. The test was named TACTIC-KIBO (Tufts Assessment of Computational Thinking in Children). The KIBO robot platform is used worldwide to teach coding skills to young children. The test was developed on the basis of the seven powerful ideas of computational thinking as set forth by Bers (2018). These concepts are: algorithm, modularity (iterative structures), control structures, representation, hardware/software, design process, and debugging. The other starting point is the programming development model proposed by Vizner (2017), which identifies four distinct levels of programming development in children: the proto-programming level, the early programming level, the programming level and the fluent programming level. The TACTIC-KIBO tasks and questions were developed on the basis of these two theories, with questions created for each of the four levels for each concept. The results indicate that TACTIC-KIBO is a promising instrument for assessing young children’s computational thinking, which can be used to determine the four levels of programming proficiency.

The three measuring tools assume prior coding knowledge, so children's previous knowledge affects the results. In addition, other researchers have worked on developing tools that do not require prior programming knowledge or any technological/digital tools. The tools presented below are multiple-choice, which means that the child has to choose the correct answer from a set of answers. One of such an assessment tool is the Bebras Unplugged Computational Thinking Cards, developed by the international community of educators of the Bebras Challenge (www.bebras.org), with the intention of promoting computer science and computational thinking in schools. The test is for children aged 3 to 10, with different levels of difficulty for each age group. Each card measures a computational thinking concept: patterns, algorithms, logic and abstraction. The 48 cards are divided into three difficulty levels: easy (16), medium (17) and hard (15). Sung and colleagues (2022) validated the cards with Korean children. The results show that the Bebras cards have acceptable psychometric properties and are suitable for measuring different computational thinking skills in young children.

The TechCheck Computational Thinking (CT) assessment, developed by Relkin and colleagues in 2020, measures computational thinking domains similar to Bebras. The TechCheck measures the computational thinking concepts/areas described by Bers (2018): algorithm, modularity (iterative structures), control structures, representation, hardware/software, and debugging. One of the seven areas was not included: the design process is not measured in the test. The online tool contains 15 multiple-choice questions with four possible answers. Each correct answer is worth 1 point (15 points in total). A total of 768 children (5 to 9 years old) were tested and the instrument was administered in groups. Ther results were compared with the results of TACTIC-KIBO for validation purposes. Overall, the TechCheck has moderately good psychometric properties and has been shown to be a valid and reliable tool for measuring computational thinking skills in children between the ages of 5 and 9. Researchers have developed a version of TechCheck for younger age groups: the TechCheck-K (Relkin and Bers, 2021), a measurement tool for preschoolers. The most important change is that instead of four possible answers, there were only three possible answers kept for each question, of which the children had to choose one. The rationale for this change was that previous research has shown that preschool children (5-year-olds) have limited working memory (the ability to hold three items at once) (Simmering, 2012), which may affect their performance on multiple-choice tasks (when there are more than three answer options). The test was online and the children had to click/choose the correct answer from three. The question was read out aloud to the children by the experimenter two times in a row. The study involved 89 kindergarten children between the ages of 5 and 6. According to the researchers, the advantage of the test is that no prior programming knowledge is required to complete the tasks, but they also noted some limitations: its multiple-choice nature precludes creative self-expression and open-ended problem solving, which are important/significant parts of computational thinking.

Zapata-Cáceres and colleagues (2020) developed the Beginners Computational Thinking Test (BCTt), which is designed for children between the ages of 5 and 12. The BCTt measures the following computational thinking concepts: sequences, loops (simple and nested), conditionals (if-then, if-else, while statements). The test was designed for young children who could not read or write, so they used symbols and drawings that were self-explanatory. Two types of tasks were designed: line-tracing tasks (where the children had to draw a pattern) and maze or matrix tasks. The 25-question test required the correct answer to be chosen from four possible answers, which were represented by a series of steps in the form of arrows, symbols and numbers. The first version was sent to 45 professionals who were asked to rate the items on the basis of 66 questions. Per their suggestions, a second version was developed and completed with 299 children. The results show that the level of difficulty of the test is best suited to pupils in the primary school age group (5- to 10-year-olds).

A test similar to the previous one was developed by Zang and Wong (2023), which they called the Computational Thinking Test for Lower Primary (CTtLP). The computational thinking concepts included in the test were: sequences, directions, loops and conditionals. As in the case of the previous tool, two types of tasks were developed: drawing tasks and driving a character on a square grid. For the square grid driving task, the character was adapted from the Román-Gonzáles (2015) test, and the square grid was adapted from the Zapata-Cáceres et al. (2020) test. A total of 30 items were developed according to three scenarios: Drawing, Pac-Man and Hungry Snake. For validation, the test was evaluated by 20 experts using an online questionnaire, 6 students were interviewed to analyse the questions, and a pilot test was conducted with 72 primary school students. Based on feedback from experts and students, several changes were made, including the addition of a small arrow indicating the direction of departure and the replacement of the term “90 degrees” with a visual representation, as this concept is still unfamiliar to second graders. Each correct answer was worth 1 point. The results show that the test is well adapted to the measurement objectives and that, overall, the test is appropriate for the age group.

2.2.2. Developing computational thinking in preschool

Wing (2017) emphasizes that the concept that he introduced influenced many disciplines in the more than 10 years since its emergence. The development of this skill is also increasingly promoted in education. The teaching of computational thinking is widespread at the international level, with many countries already including it in their curricula for primary school children. The UK Department for Education changed the national curriculum to include the teaching of computational thinking to all students from 2014. (UK Department for Education, 2013). In a 120-page report, the Danish Growth Council summarised the importance of teaching computational thinking at all levels of education and made recommendations to the Danish government in this respect. (The Danish Growth Council, 2016).

Kazakoff, Sullivan and Bers (2012) investigated changes in preschool children's sequencing abilities after they had participated in an intensive robot programming task. Their main hypothesis was that sequencing is the component of the processes involved in robot programming that is also present when children put the events of a story into a logical order. The differences between the pre- and post-test results showed a significant difference in the image alignment tasks for the experimental group. They concluded that the children involved in robot programming had improved their ability to create a sequence.

Bers and colleagues (2014) applied the 'TangibleK' Robotics Programme to three kindergarten groups, using the framework of constructionism and Positive Technological Development as a basis/starting point. The 'TangibleK' curriculum promotes the development of computational thinking. The programme focuses on the following computational thinking skills: defining a problem, consistency in generating and applying solutions, exploring multiple possible solutions, multi-level problem solving, being creative in the face of failure and identifying misconceptions/misunderstandings on the way to a successful project, applying strategies to tackle difficult problems. The results show that the curriculum can be used by kindergarten teachers and that preschool children are interested and able to learn and apply many aspects of robotics, programming and computational thinking.

Lindenberg and colleagues (2019) point out that there is a growing global trend in education to teach computational thinking, which predicts the importance of introducing computational education from preschool onwards. As preschool children have limited cognitive abilities, teaching them the logic of programming and developing an appropriate curriculum became a necessity. Building on this, Ching and colleagues (2018) in their research set out to create a framework to help preschool children develop computer skills, increase interest in learning, and improve learning outcomes. A game-based learning approach was integrated with a Tangible User Interface (TUI) – an interface where digital information can be interacted with using physical tools (such as a mouse) – to develop computational thinking skills in preschool children. The learning outcomes were measured with questions of different levels of difficulty and, based on their analysis, it was concluded that the game-based learning method combined with a tangible user interface is effective in improving the learning performance of kindergarten children and enhancing their computational thinking skills.

2.3. Educational robots in kindergarten

2.3.1. Educational robots

Since Seymour Papert developed the LOGO programming language in 1967, educational robots have been introduced into the educational process as didactic tools and are still the subject of much research interest. The use of robots has been on the increase for the past two decades. There are now many robots designed especially for young children. Some robots, such as the Bee-bot/Blue-bot and the Colby mouse robot, can be programmed using buttons on the back. These robot sets also come with command cards (go forward, go back, turn left, turn right) so that the child can create a sequence of commands (algorithm – code) before programming the robot. These robots also come with square grid boards: the robot moves one square forward or backward on command (see the Colby mouse robot board in Figure 3).

Another type of robot, the KIBO, can be programmed to fit wooden cubes/blocks together (using holes and handles) (Bers, 2018). Each block represents a command (walk forward, walk backward, turn left, turn right, turn around, turn on lights, whistle, wait for applause, etc.). This robot can perform more commands than the button robots presented above. It also has structures such as repetition and the ‘if’ condition. Research has shown that preschool children are able to understand and use repetition and numerical parameters (Bers et al., 2014), which can be programmed with the KIBO robot.

A third type of screenless robot is Cubetto, which comes with a board and instruction blocks to place on the board. There are no buttons on the back of the robot; it can only be programmed with the help of the circuit board. An important feature of the robot is the function line, which allows the use of subprograms (Gadzikowski, 2018). There are four places in a special area of the board where up to four blocks can be placed, and these blocks form a subprogram. Using these, when Cubetto is programmed, only the subprogram block needs to be placed in the instruction line. The function line helps children practice abstraction and modularisation (Yu and Roque, 2019).

2.3.2. Using educational robots in kindergarten

The latest generation of robotics kits for young children allow for manipulative learning, but robotics is generally a non-screen-based activity that promotes teamwork and collaboration (Sullivan and Bers, 2016).

Research has shown that children as young as 4 to 6 years old can design and build simple robots (Cejka et al., 2006), while acquiring knowledge in engineering, technology and programming and developing their computational thinking skills. Robotics activities develop children’s fine motor skills and hand-eye coordination while they cooperate and work in teams. They also experiment with/explore engineering concepts and practice storytelling/story building by attaching stories/narratives/plots to projects (Bers, 2008).

In a study conducted in Sweden, Palmér (2017) investigated the acquisition of the use of computational concepts in problem-solving activities and, more specifically, the relationship between spatial reasoning and programming among preschoolers’ mathematical skills. Children between the ages of 3 and 5 were engaged in a series of programming activities as part of their daily preschool curriculum over a four-month period, and the impact of these activities on their spatial reasoning skills was studied. Spatial reasoning involves the mental comparison, rotation and memorisation of relationships, conceptualisation, as well as the transformation of objects. The research involved children working with a robot called the Bee-Bot. Their results showed that the children took part with a great deal of enthusiasm and that all of them were able to complete the tasks. In addition, the children understood the meaning of the symbols (arrows pointing in different directions), applied them correctly and used hand and body movements to demonstrate how the robot would move based on the arrows.

Otterborn and colleagues (2019) also studied kindergarten teachers' views on the content, aims and methods of programming activities based on their own teaching practice in Sweden. Based on the results, two main approaches to the use of programming in kindergartens were identified: unplugged programming, i.e., programming without any digital tools, and digital programming. One way of programming without digital tools is for one child to take the role of the robot and another child to give different commands to move and navigate the "robot". Two types of digital programming were distinguished: direct programming through various applications/programs (on a computer/tablet) and hands-on programming through manipulative tools, e.g. Blue-Bot and other robots. The results show that the kindergarten teachers embedded programming in different projects and themes, and that it strengthened cooperation, problem solving and the children's belief in their own abilities.

To investigate computational thinking using the KIBO robot kit, Bers and colleagues (2019) conducted a study with preschool children (3 to 5 years old). The results show that robotics helped children develop high levels of coding and computational thinking skills.

Çakır (2021) and colleagues investigated the effects of programming and robotics activities on preschool children's problem-solving skills and creativity. The Problem Solving Skill Scale (PSSS) was used to assess problem solving skills and the Integrative Creativity Test was used to assess creativity. During the intervention, the experimental group participated in programming and robotics activities, while the control group was given paper-and-pencil tasks. The results showed that the experimental group performed significantly better on the post-test in both areas.

Bakala (2021) and colleagues conducted a systematic review of published research on the effects of robotics/programming activities on pre-school children's computational thinking. They sought answers to four questions: What kind of robots were used in the studies, and how can they be classified? What are the characteristics of the activities that aim to stimulate the development of computational thinking? How was computational thinking evaluated? Which individuals and countries are most active and influential in research on computational thinking development for preschool children mediated by robots, and what have been their motivations for conducting research in this area? Ultimately, 15 relevant articles remained in the study. The review found that all the studies used commercially available robots; there was no consensus on the use of these activities; the structure, duration and design were also very different; the survey method used was also very heterogeneous. The results suggest that there is a need for more rigorous research reporting in this area.

3. VIEWS AND EXPERIENCES OF PROSPECTIVE AND CURRENT PRESCHOOL TEACHERS ON THE USE OF STEM ACTIVITIES AND THE DEVELOPMENT OF ALGORITHMIC/COMPUTATIONAL THINKING

3.1. The views and experiences of students of preschool and primary education on the use of STEM activities and the development of algorithmic/computational thinking in preschool

Method

This survey was carried out in the first semester of the 2021-2022 academic year, among students of elementary and preschool education. The survey had several aims: one was to explore students' views and experiences of using STEM activities in kindergarten, and the other was to explore students' views and experiences of using the technology part of STEM activities in kindergarten. At the same time, we also wanted was to explore students' views on algorithms and the importance of developing algorithmic and computational thinking in pre-school.

Research questions

1. What experience areas and related activities can you successfully integrate when planning preschool activities?
2. Do you know the term STEM and what it means?
3. In your opinion, what is the purpose of STEM activities, what opportunities do they offer and what cognitive, social, physical, emotional skills/abilities do they develop?
4. What personal experience have you gained in applying STEM activities in your teaching practice?
5. What would encourage more STEM activities in kindergartens in your opinion?
6. What is your opinion and experience of technology activities as part of STEM?
7. What prior knowledge/experience do you have with algorithms?

8. What knowledge and experience do you have on developing digital literacy in preschool?

Participants

85 students of the Pedagogy of Primary and Preschool Education programme at the Babeș-Bolyai University took part in the research: 51 second-year students and 34 third-year students. In terms of gender distribution, 1 (1.2%) of the respondents was male, which can be explained by the fact that women are generally interested in the programme and the majority of applicants are women.

Research tool

The survey was carried out using an online Google Forms questionnaire containing a total of 36 questions. Of these 36 questions, 5 were related to demographics, 3 questions were related to the way preschool activities are integrated, 10 questions were related to their knowledge and experience of STEM activities, 7 questions were related to the integration of technology as part of STEM in preschool and 11 questions were related to the importance of developing computational thinking in preschool. The questionnaire consisted of both open-ended questions and closed-ended questions (multiple-choice questions and statements that were measured on a 5-point Likert scale). The questions on STEM activities were taken from another questionnaire used in a multi-country ERASMUS+ project coordinated by the Universitat Internacional de Catalunya, called "Kitchen Lab for Kids". (K4K, 2020).

Results

1. **Question: What experience areas and related activities can you successfully integrate when planning preschool activities?**

95.3% of the respondents integrate activities from experience areas. This result confirms that the principle of integration is also implemented in planning, as recommended in the curriculum for early childhood education (Ministerul Educației Naționale, 2019). The second question was about how to integrate: they had to choose the type of activity that would be the most optimal to be integrated. The results show that both second- and third-year students most frequently associated mathematics (40) and mother tongue (17) with natural science education, and that mathematics was also most frequently associated with natural sciences (38). In the case of mathematics, the activities chosen to accompany natural science education differed between second- and third-year students: second-year students tended to integrate it with visual arts (8) and crafts (6), while third-year students linked it with mother tongue (4) and physical education (4). Both groups combined mother tongue activities with music (25), natural science (14) and visual arts (16) education in varying proportions. Romanian (which is the official state language but is taught as a foreign language), was most frequently combined by both groups with music (28) and visual arts (17). For the integration of musical activities, second-year students preferred physical education (16) and third-year students preferred visual arts (12), but physical education was also the second most common response (9) here as well. Both groups would associate visual arts education with the same activities, but to different extents: while the second-year students would associate it with natural science education (16), the third-year students chose mother tongue (15) in greater numbers. Both groups integrated civic education and household activities with mother tongue activities (39). The biggest difference was observed in craft activities, with the groups choosing completely different activities: the second-year group preferred to combine it with natural science activities (12) and mathematics (9), while the third-year group integrated it with mother tongue (11) and music (5). Physical education activities were mostly combined with musical education (37) in both grades.

2. **Question: Do you know the term STEM and what it means?**

67.4% of respondents were not aware of what the term STEM meant: 64.7% of second-year students and 73.52% of third-year students were not familiar with the STEM acronym. It is noteworthy that the results for third-year students are worse, indicating that the subjects they have studied at university have not facilitated a deeper understanding of STEM education. Comparing these results with those of other countries, it appears that the STEM knowledge of students of preschool teacher education and practicing preschool teachers varies from country to country and even depends on the higher education institution where they study. For example, Karademir and Yıldırım (2021) found in their study that final year preschool teacher education students could define what STEM education meant, whereas Baltsavias and Kyridis (2020) found that practicing preschool teachers were less familiar with STEM education. Furthermore, preschool teachers participating in the pilot study by Aleksieva and colleagues (2021) were completely unaware of the concept of STEM and what it meant before the intervention. These results suggest that the competence of educators in the teaching of STEM subjects depends on the curricula of the institutions that prepare them.

After a definition and brief description of what STEM education is, students were asked to decide whether some of the statements were true or false. The first false claim was that STEM activities were included in the current Romanian national curriculum. 7.84% (4) of second-year respondents thought this statement was true, while none of the third-year respondents marked it as true. The second incorrect claim was that STEM activities can only be successfully applied in schools. In this regard, 11.76% (6) of second-year students thought the statement was true, and 5.88% (2) of third-year students had the same opinion. The third false claim was that STEM activities always require digital tools. 18.82% (16) of respondents thought this false statement was true: 21.56% (11) of second-year students thought it was true, while 14.7% (5) of third-year students had the same opinion. An interesting finding is that despite the fact that third-year students were less familiar with STEM education, after a short explanation, they had a better understanding of what STEM was. It is possible that they already had some knowledge of STEM education, but had not yet had exposure to the STEM acronym itself.

3. **Question: What is the purpose of STEM activities, what opportunities do they offer and what cognitive, social, physical, emotional skills/abilities do they develop?**

To explore students' perceptions of the purpose of STEM education and the opportunities it can offer in preschool education, two sets of statements were formulated and the results were measured using a 5-point Likert scale. A two-sample t-test was used to compare the responses for each of the statements (Table 1). Although there were small differences between the means for some of the statements, no statistically significant differences were found for any of those listed. With regard to the point of STEM education, most of the students agreed with the statement that it encourages children to think creatively in the natural sciences. Other studies (Karademir and Yıldırım, 2021) have also highlighted creative thinking as a key benefit of STEM education. Most students agreed with three statements about the most important opportunities provided by STEM education: STEM activities develop children's knowledge of the social, natural and technical world; they provide practical experience; and they provide positive emotions and motivation to learn science.

Table 1. The comparison of second- and third-year students' responses to statements about the purpose and opportunities of STEM activities in kindergarten

Statement	*Second year*		*Third year*		*p*	*t*
	Mean	*SD*	*Mean*	*SD*		
The purpose of STEM education						
1. Encouraging children to learn through direct and personal experiences	4.24	0.66	4.26	1.11	.890	0.138
2. Encouraging children to think creatively in the natural sciences	4.55	0.49	4.32	0.77	.215	-1.254
3. Designing an active learning-teaching process	4.16	0.69	4.24	0.91	.698	0.390
4. Identifying and solving problems in natural, everyday situations	4.12	0.83	3.94	1.33	.456	-0.750
5. Developing an integrated, holistic world view in the child's mind	3.94	0.74	3.91	1.05	.891	-0.138
6. Encouraging the holistic development of children	4.02	0.62	3.85	0.86	.392	-0.862
7. Developing the teaching-learning process by including at least two STEM areas	4.16	0.77	4.00	1.09	.473	-0.721
Opportunities in STEM education						
1. Developing a positive self-image	3.61	0.92	3.59	1.16	.932	-0.086
2. Developing positive emotions and motivation to learn science	4.53	0.41	4.38	0.55	.348	-0.946
3. Developing children's knowledge of the social, natural and technological world	4.63	0.32	4.38	0.55	.106	-1.641
4. Self-directed and independent learning	4.04	0.80	4.18	0.82	.493	0.690
5. Cooperative learning	3.90	0.93	4.03	0.70	.519	0.648
6. Gaining practical experience	4.59	0.33	4.47	0.68	.472	-0.723
7. Encouraging learning through	4.47	0.61	4.29	0.88	.368	-0.906

play						
8. Asking questions and finding answers through experimentation	4.41	0.73	4.35	0.72	.756	-0.312

In the next four questions, we asked the students what kind of cognitive, emotional, social and physical skills/abilities they think are developed through STEM activities. In the area of cognitive skills, the majority, 75.6% (56) believe that exploratory and creative thinking is best developed through the early application of STEM activities. This result is in line with the results from countries participating in the K4K Erasmus+ project (K4K, 2020). Turkish preschool teacher candidates also believe that STEM activities contribute to the development of children's creativity (Karademir and Yıldırım, 2021, Ültey and Ültey, 2020). 81.4% (70) of respondents felt that teamwork was the social skill/ability most developed through STEM activities. This result is also consistent with the findings of previous studies (K4K, 2020, Karademir and Yıldırım, 2021). In terms of emotional skills/abilities, 68.6% of respondents said that STEM activities were the most effective in developing independence. This response was the second highest in the results of the research on the basis of which we developed and adapted our questionnaire (K4K, 2020). According to students, the two physical skills/abilities most developed by STEM activities are: experiencing the world through the senses (60.5%, 52 responses) and fine and gross motor skills/abilities (59.3%, 51 responses). These results are consistent with those obtained by the K4K project (2020).

4. **Question: What personal experience have you gained in applying STEM activities in your teaching practice?**

60.5% (52) of the respondents reported having no experience in this area (Table 2). This finding is somewhat puzzling as most of the preschool teachers and preschool teacher students in the K4K project already had some kind of personal experience with STEM activities. The most frequently mentioned area (by 18 students) in which they had experience was carrying out scientific observations and experiments.

Table 2. Respondents' personal experience with STEM activities

Type of experience	*Second-year students*	*Third-year students*	*Total (frequency)*

	(frequency)	*(frequency)*	
1. Leading/conducting scientific observations and experiments	11	7	18
2. Studying the physical characteristics of the world	8	3	11
3. IT-related workshop exercises (e.g., how mini robots work)	5	0	5
4. Interdisciplinary projects combining at least two STEM fields	1	3	4
5. Forest kindergarten/field trips	7	3	10
6. Visiting science centres, university laboratories	8	3	11
7. I have no experience of using STEM activities in early childhood	29	23	52

5. **Question: What would encourage more STEM activities in kindergartens in your opinion?**

Respondents stated that in order to increase the frequency of STEM activities in kindergartens, preschool teachers' knowledge of STEM content and methodological training related to these activities should be improved (64%, 56 responses). Research has shown that training in STEM education and supporting students to plan and implement STEM activities in kindergarten groups are effective ways to increase students' and preschool teachers' confidence in their STEM knowledge and change their attitudes towards STEM activities (Aleksieva et al., 2021, Fridberg et al., 2022). In addition, disciplines related to STEM education should be included in the training of future preschool teachers. Even the introduction of a single discipline related to STEM education can have a significant impact on the tendency of preservice kindergarten teachers to integrate/apply STEM activities in their work (Uğraş and Genç, 2018). However, the implementation of STEM activities also requires long-term professional support. The most effective professional development programmes are those that provide mentored internships and experiences in STEM education (Chen et al., 2021).

11.6% of respondents felt that changing the motivation of preschool teachers to engage in STEM activities would also help improve the situation. Research shows that motivation to teach STEM increases as confidence and methodological knowledge of

STEM content increases (Aleksieva et al., 2021); this means that providing appropriate training may be the answer. 9.3% of respondents believed that providing kindergartens with STEM tools would also contribute to increasing the frequency of STEM activities, as in many cases one of the main barriers to introducing STEM activities in kindergartens is the lack of the necessary tools (Ültey and Ültey, 2020).

6. **Question: What is your opinion and experience of technology activities as part of STEM?**

To explore respondents' views on the use of technological knowledge in preschool activities, four claims/statements were formulated. Students were asked to rate the statements on a 5-point Likert scale (where 1=strongly disagree and 5=strongly agree).

The students agree that a positive attitude towards technological tasks can be established in kindergarten. They are, however, not convinced that technological skills should be taught in kindergarten. They disagree even less with the claim that the technological tasks carried out in preschool serve as a basis for programming (design, implementation). This can be explained by their lack of experience in integrating technological knowledge or skills into preschool activities, as 89.5% (77) of them have never tried to develop children's technological competences. We then asked respondents who had used technological skills to describe (in an open question) which skills they had focused on and how they had used them in their activities. The analysis of the responses showed that many confused technological education with the use of technology, as most of them understood technological education to mean the use of technological tools. This conceptual confusion/misunderstanding may explain why students are not that positive about the integration of technological education in kindergarten. From the answers to the next question, we wanted to find out what was preventing them from planning activities based on technological knowledge. 40.3% of respondents (31) said that they were not methodologically prepared enough to develop such activities, 32.5% (25) said that the lack of appropriate didactic tools was the biggest barrier, and 28.6% (22) identified low confidence as a barrier. 95.3% of respondents (82) would like to know more about activities to introduce technological skills in kindergartens. This result is consistent with the findings of Zsoldos-Marchis and Ciascai (2019), and highlights the importance of integrating technology activities – as part of STEM activities – in the training of preservice preschool and primary school teachers. The methodological skills needed to implement

STEM activities can be effectively acquired through training and mentoring programmes (Uğraş and Genç, 2018, Chen et al. 2021).

7. **Question: What prior knowledge/experience do you have with algorithms?**

In the first question of this section, the students were asked to give examples of activities that could be used to develop algorithmic thinking in preschool. 73.3% of the respondents (63) gave examples that could make a real contribution to the development of algorithmic thinking. Some of the examples were very general in the sense that they were just lists of experience areas that use algorithms, such as mathematics, science and art. Others gave examples from kindergarten routines such as morning or mealtime routines. These were accompanied by correct, specific examples, such as using robots, building with blocks, making a piece of craftwork following the steps given by the teacher, etc.

To assess students' algorithm writing skills, they were asked to describe the steps involved in building a snowman. The following solution was considered correct and accepted: 1. make three snowballs of different sizes, 2. put the middle snowball on top of the biggest, then the smallest on top of the middle, 3. put two eyes in the middle of the smallest, a nose and a hat on top. Of course, these three main steps have many sub-steps, which can also be seen as sub-programs. There are also steps that are interchangeable, so the order is not important, e.g. someone could do the nose of the snowman first, then the eyes. Only 32.94% (28) were able to formulate the steps of the algorithm for building a snowman correctly. Most respondents made the mistake of not describing the sequence of steps in sufficient detail.

Regarding students' previous experience with algorithms, 70.9% (61) had only heard of the term algorithm in mathematics class, 15.1% (13) had studied programming and 9.3% (8) said they had never studied algorithms. The students involved therefore had very limited experience of algorithms and coding. However, 93% of students (80) felt that there was a need for early development of algorithmic thinking.

8. **Question: What knowledge and experience do you have on developing digital literacy in preschool?**

Respondents were asked to rate three statements about the types of activities associated with developing computational thinking on a 5-point Likert scale, according to how much they agreed or disagreed with the statement (1 - strongly disagree, 5 - strongly agree). The means and standard deviations regarding the claims are summarised in Table 3.

Table 3. Opinions on the development of computational thinking in terms of experience

Statement	*Mean*	*SD*
For the development of computational thinking, there is always a need for a computer	2.84	1.204
Computational thinking can be developed/taught without the use of computers.	3.57	1.189
Computational thinking can also be developed with paper-and-pencil exercises.	3.39	1.256

The responses showed that most participants agreed with the statement that "computational thinking can be developed without the use of computers". This is a promising finding, as one of the main reasons why preschool teachers do not consider it appropriate to develop computer skills at this age is that, at least in preschool, screen time should be limited. In their study, Yavadav and colleagues (2014) found that students enrolled in preschool education programmes who had not received training in computational thinking were convinced that computational thinking required the use of computer technology. 67.4% of respondents (58) consider the development of computer skills in kindergartens to be important. The students were then asked to give examples of activities that could be used to develop computational thinking in kindergarten. 47.67% (41) gave good examples of how to develop computational thinking. According to them, computational thinking can be developed through tasks based on algorithms, tasks based on problem-solving, games based on sequential steps, etc. Two students also mentioned the use of educational robots as a possible tool for development. However, there were also students who gave incorrect answers, such as saying that computational thinking could be developed by showing children the computer and its parts. Some knowledge of hardware and software is part of computational thinking, but it is not a competence that should be developed in early childhood education. Another misconception of respondents about developing computational thinking was that it can be done with mathematics, which is not consistent with the concept of computational thinking. Computational thinking as an approach to solving mathematical problems was also mentioned by Sands and colleagues (2022), as well as Avci and Deniz (2022) in their research. Computational thinking is in fact more about algorithms and coding, and almost half of the respondents understood this, as shown by the examples they gave. Research by Ari and colleagues (2022) also found that preschool teacher candidates did not integrate coding into their kindergarten activities.

As robots are often used in kindergarten activities to teach algorithms and programming, the students were asked if they had ever heard of educational robots. Only 22.1% (19) gave a positive answer. The robots they were familiar with included the Blue-bot/Bee-bot robot (20.9%, 18 students), Ozobot, Vex and Cubetto – mentioned by 5 students (5.8%). The other respondents were not aware of any educational robots. None of the respondents had ever used educational robots in the classroom. This finding is consistent with the fact that learning with educational robots is not yet widespread in Romania, and the few such extracurricular activities that do exist tend to be offered only to school-age children. Taking into account the research results from other countries, it would be appropriate to include computational thinking as a possible area for development in the preschool curriculum requirements. At the same time, appropriate tools should be made available, such as educational robots.

3.2. Preschool teachers' views on developing technology-related skills in kindergarten

The aim of the study

The aim of the study was to gain insight into the perceptions and experiences of preschool educators in Romania regarding the integration of technological knowledge and tools in early childhood education.

Research questions

The objective of the research was to address the following questions:

1. What is the attitude of the preschool teachers towards the use of technology in kindergarten?
2. Have preschool teachers tried to integrate technological knowledge into the preschool teaching and learning activities?
3. What knowledge and experience do they possess regarding the conceptualisation and implementation of algorithmic and computational thinking, and the development of these competencies in kindergarten?
4. What is the attitude of preschool teachers towards the integration of computational thinking into the preschool curriculum?
5. Are they familiar with educational floor robots?
6. What are their experiences with using educational floor robots?

Method

The research was conducted during the 2021-2022 and 2022-2023 academic years in Romania.

Participants

The survey was conducted on 115 kindergarten teachers, all of whom were practising preschool teachers within the public education system in Romania, from Harghita, Covasna, Mureș, Cluj, Sibiu, Bihor, Sălaj, Satu Mare, Timis, Bistrița-Năsăud Counties. All respondents were women. In terms of educational qualifications, 81% of the preschool teachers (93) have obtained a Bachelor's degree, while the remaining 19% have obtained a Master's degree (MSc). In terms of the didactic degree achieved thus far, 49% (56) of the respondents are Level I teachers, representing the highest level of qualification. 20% (23)

are Level II teachers, 14% (20) have passed the so-called *definitivat* or teacher-confirmation examination, and 17% (16) are early career teachers. In terms of work environment, 60% of respondents (69 individuals) work in urban areas, while 40% (46 individuals) teach in rural settings.

Research tools

The survey was carried out using an online Google Forms questionnaire containing a total of 23 questions. There were six questions on various demographics, seven questions on the use of technology in kindergarten, and ten questions on the importance of algorithmic and computational thinking and its potential for development, including the use of floor robots in an educational context. The questionnaire consisted of 5 open-ended questions; the rest were closed-ended questions (multiple-choice questions and statements that were measured on a 5-point Likert scale). The questionnaire included questions on algorithmic and computational thinking in separate sections. In order to guarantee that all respondents had a uniform understanding of the terms "algorithmic thinking" and "computational thinking," we introduced the questions with our agreed-upon definitions of both terms.

Results

The results are organised into three topics, which combine two research questions each: (1) technological knowledge in kindergarten; (2) developing algorithmic/computational thinking; (3) using floor robots in kindergarten.

1. **Technological skills in preschool teaching and learning activities**

In order to find out what kindergarten teachers think about the introduction of technological knowledge in kindergarten, they were asked to rate four statements on a 5-point Likert scale (where 1=strongly disagree and 5=strongly agree). The mean and standard deviation of the responses are presented in Table 4.

Table 4. Mean and standard deviation per statement on the use of technological skills in kindergarten

Statement	*Mean*	*SD*
It is necessary to introduce technological skills in early childhood.	3.05	1.234
There is no place for technological skills in early	2.18	1.288

childhood.		
Early childhood can help shape how we approach technological activities.	3.41	1.199
Kindergarten technology tasks involve laying the foundations for programming (design, implementation).	2.91	1.274

Based on the responses, kindergarten teachers mostly agree that a positive attitude towards technological activities can be established in kindergarten. At the same time, most also agree that technology skills need to be introduced in early childhood. They agree less with the statement that kindergarten technology tasks involve laying the foundations for programming (design, implementation).

The study also examined whether there were notable differences in the responses of preschool teachers in rural and urban settings. As the sample was not normally distributed according to the Shapiro-Wilk test, a Mann-Whitney test was employed for comparison purposes. This revealed no statistically significant difference between the views of preschool teachers working in rural and urban settings with regard to technological literacy. However, it should be noted that urban teachers scored higher on all statements than their colleagues working in rural educational institutions (see Table 5).

Table 5. Comparison of the responses of preschool teachers in rural and urban areas on the use of technology in kindergarten using the Mann-Whitney test

Statement	*Rural area N=41*		*Urban area N=66*		*W*	*p*
	Mean	*SD*	*Mean*	*SD*		
It is necessary to introduce technological skills in early childhood.	2.98	1.18	3.10	1.27	1510.00	.652
There is no place for technological skills in early childhood.	2.26	1.34	2.13	1.26	1666.50	.636
Early childhood can help shape how we approach technological activities.	3.26	1.10	3.52	1.25	1362.50	.184
Kindergarten technology tasks involve laying the foundations for programming (design, implementation).	2.85	1.10	2.96	1.39	1526.00	.722

The question was raised whether the educational level of the preschool teachers influences their opinion on the introduction of technological activities in kindergartens. As the sample is not normally distributed according to the Shapiro-Wilk test, the Mann-Whitney test was employed for comparison purposes. The results are shown in Table 6. Based on the first two statements, preschool teachers with a master's degree perceive a significantly greater need to introduce technological skills into kindergarten activities.

Table 6. Comparison of the responses of preschool teachers with a bachelor's and master's degree on the use of technology in preschool education using the Mann-Whitney test

Statement	*Bachelor's degree N=93*		*Master's degree N=14*		*W*	*p*
	Mean	*SD*	*Mean*	*SD*		
It is necessary to introduce technological skills in early childhood.	3.01	1.24	3.71	1.12	440.00	.045
There is no place for technological skills in early childhood.	2.20	1.28	1.36	0.63	894.50	.017
Early childhood can help shape how we approach technological activities.	3.37	1.22	3.86	1.17	495.00	.137
Kindergarten technology tasks involve laying the foundations for programming (design, implementation).	2.86	1.30	3.36	1.15	501.50	.156

The next question was whether or not they had tried to integrate technological skills into kindergarten activities. 55% of respondents (63 individuals) had not tried to integrate technology into their preschool activities, while 45% (52 individuals) had. However, the following open-ended question suggested that there were some conceptual problems. We asked preschool teachers to give us specific examples of how they had used technological skills in their kindergarten activities, and the responses showed that many confused technological literacy with the use of technological tools and digital aids. Respondents tended to interpret technological literacy as the use of technological tools (e.g. learning to use online platforms, apps, smartboards, tablets, phones, computer mice). Others considered experimental activities as part of technological education: observing how different tools work or discovering cause and effect relationships. This skews the answer to the previous question, with less than 45% of respondents using integrated technology skills.

Most of the teachers who had tried to incorporate technological skills said that the activities they organized were enjoyed by the children and that they were actively involved in the tasks. The next question was about the difficulties they encountered in the planning of activities. 26.9% (31) of the preschool teachers blamed the lack of pedagogical tools for the difficulties they had encountered. 16.5% of respondents (19 individuals) said that insufficient methodological preparation made it difficult to plan such activities, 14.7% (17 individuals) believed that the kindergarten environment was not suitable for such activities and 12.1% (14 individuals) cited insufficient theoretical preparation as a reason.

The next question was about the barriers that prevent preschool teachers from integrating technological skills into kindergarten activities. 37.3% of respondents (43 individuals) cited insufficient methodological preparation as the main obstacle, the same number (37.3%) cited the lack of appropriate didactic tools as the main reason, while 30.4% (35 individuals) cited insufficient theoretical preparation as the main obstacle. However, when asked if they would like to learn about activities that are based on the introduction of technological skills and to plan activities that introduce technological skills, 91% (105 individuals) of the respondents said yes.

2. Developing algorithmic/computational thinking in preschool

After reading the definition of algorithmic thinking, preschool teachers were asked to formulate their opinion on the activities that could be used to develop preschool children's algorithmic thinking. The responses were very diverse, with 76.5% (88 individuals) finding examples of activities that contribute to the development of algorithmic thinking. Of these, 37.5% (33 individuals) mentioned mathematical activities in general, highlighting grouping, classification and logical tasks. In addition, 19.3% (17 individuals) mentioned seriality/sequencing (e.g. following the steps of completing a task or determining the order of events). Closely related is following the steps of craft/art activities, chosen by 10.2% (9 respondents), and the following of daily routines, chosen by 13.6% (12 respondents). In addition, 10.2% (9 individuals) mentioned building as an activity to develop algorithmic thinking, 9.09% (8 individuals) considered rule-based games and 10.2% (9 individuals) considered science/environmental activities as activities to develop algorithmic thinking.

Floor robots and programming were only mentioned by 5.6% (5 respondents). The obvious reason for this is that robots and the programming of robots for educational purposes are alien to preschool teachers.

Regarding their previous experience with algorithms as preschool teachers, 77.3% (88 individuals) said that they had only heard of the term in mathematics lessons, 12.1% (14 individuals) had never studied algorithms, 4.3% (5 individuals) had studied programming and 2.6% (3 individuals) had attended a training course on algorithmic thinking. The results reflect a lack of awareness in regards of the developments in this area, as the teachers themselves have not been trained in this area and most of them do not even have sufficient experience of the concept. Nevertheless, 91% of preschool teachers (105 individuals) think it is important to develop algorithmic thinking in kindergarten.

We focused on computational thinking in the following questions. In the first question, preschool teachers were asked to rate three statements about developing computational thinking on a Likert scale according to how much they agreed with them (where 1=strongly disagree and 5=strongly agree). The means of their responses and the standard deviation results are shown in Table 7.

Table 7. Mean and standard deviation per statement on the development of computational thinking

Statement	*Mean*	*SD*
For the development of computational thinking, there is always a need for a computer	2.49	1.327
Computational thinking can be developed/taught without the use of computers.	3.217	1.412
Computational thinking can also be developed with paper-and-pencil exercises.	3.48	1.272

Based on the responses, we can see that the majority of preschool teachers mostly agree with the statement that the development of computational thinking can be developed with paper-and-pencil tasks, i.e., it does not require the use of a computer. This is an encouraging result, as for a long time the development of computational thinking was equated with computer use.

We also wanted to find out whether there was a difference between the answers given by preschool teachers in rural areas and those in urban areas. Based on the Mann-Whitney test, there is no statistically significant difference between the two groups’ views on the development of computational thinking. Both groups prefer paper-and-pencil activities to develop computational thinking. The average for this statement is slightly higher for teachers in rural areas.

At the same time, it was interesting to examine the impact of educational experience on perceptions of computational thinking development. For that purpose, the teachers were split into three groups: (1) those with less than 10 years' experience, (2) those with 10 to 19 years' experience, (3) those with more than 19 years' experience. The responses of the three groups were compared using the ANOVA test and the results are included in Table 8. The results show that there is no statistically significant difference between the answers given by the three groups of respondents.

Table 8. Opinions on the development of computational thinking in terms of experience

Statement	*Less than 10 years' experience*		*10 to 19 years' experience*		*More than 19 years' experience*		*F*	*p*
	Mean	*SD*	*Mean*	*SD*	*Mean*	*SD*		
For the development of computational thinking, there is always a need for a computer	2.72	1.43	2.47	1.18	2.43	1.33	.431	.651
Computational thinking can be developed/taught without the use of computers.	3.24	1.48	3.47	1.18	3.08	1.45	.550	.579
Computational thinking can also be developed with paper-and-pencil exercises.	3.48	1.26	3.71	1.16	3.40	1.31	.389	.679

Regarding the need to develop computational thinking in preschool, 69% of kindergarten teachers (79 respondents) consider it important to develop computational thinking already in kindergarten.

When asked how computational thinking can be developed in kindergartens, the answers varied widely. The answers to the open-ended question were processed in Maxqda. Figure 1 shows the different response categories and their corresponding subcategories. The answers to our question (How can computational thinking be developed?) were grouped into six main categories: games, developing mathematical skills, using digital tools, games that develop specific algorithmic thinking, everyday activities and developing scientific thinking. Within these, several subcategories were defined, and the subcategories were further subdivided into subunits.

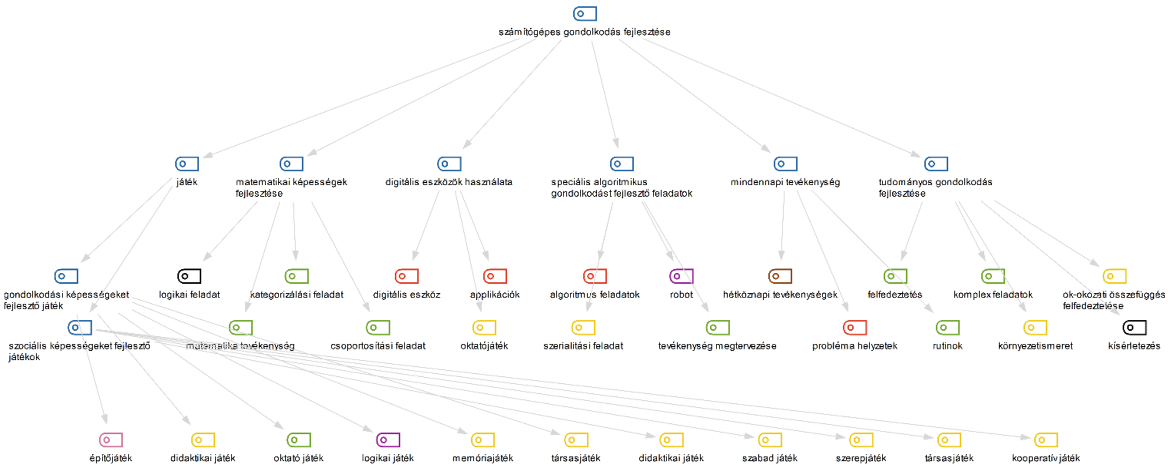


Figure 1. MAXQDA hierarchical code-subcode model

In the game category, two subcategories were defined, and common subcategories were identified for these two subcategories as well. In the main category of development of mathematical skills, four subgroups were identified. Within the subcategories, logical tasks were most frequently mentioned (by 7 respondents) as a way of developing computational thinking. The group of digital tools was divided into three subcategories, of which the majority (7 respondents) considered the use of digital tools to be suitable for developing computational thinking. The category of advanced algorithmic thinking tasks is divided into four subcategories. The subcategory of serialization tasks was most frequently referenced (cited by 8 respondents) as a way to develop computational thinking. It is noteworthy that among the respondents, only one preschool teacher proposed the use of robots as a means of developing computational thinking. In the main category of daily activities, the subcategory of problem situations was most frequently mentioned (by 5 respondents) as an activity suitable for developing computational thinking. The sixth major group is the development of scientific thinking, of which exploration was the subcategory most often identified (by 4 respondents) as an appropriate activity for developing computational thinking.

3. Using floor robots in preschool

Only 37% (42) of the preschool teachers surveyed had heard of floor robots. The best-known robot is Blue-bot, followed by Cubetto, Ozobot and Colby, the robot mouse. When asked if they had used any of these robots in their activities, only 3.4% (4 respondents) said yes. When asked to describe such an activity, one teacher mentioned the Cubetto robot but did not describe how it was used, two mentioned the Bee-bot robot, one of whom had just purchased the device but had not yet used it, and one described an activity in detail. Therefore, no relevant conclusions can be drawn from the experience of the use of floor robots in kindergarten.

4. DEVELOPMENT AND PRELIMINARY TESTING OF THE ALGOPAINT COMPUTATIONAL THINKING ASSESSMENT TEST

4.1. Developing the AlgoPaint computational thinking assessment test

In light of the findings from international research, it can be concluded that all the assessment tests presented in the second chapter exhibit one or more of the features that we have sought to eliminate in the computational thinking assessment test that we planned. Some of the tests presented above are not suitable for our purposes because they require prior programming knowledge, the use of a digital tool, they need to be taken online, or require spatial orientation and mental rotation skills. In order to exclude these, the following aspects were taken into consideration during the design of the test:

- the test should be suitable for preschool children (ages 4 to 7);
- the test should be screen-free, it should not require the use of a computer;
- the test should be independent of digital and technological devices, i.e. it should not require an educational robot, because in tests where educational robots are used for assessment, children who have previous experience with educational robots have an advantage. However, if the test is to be used as a pre- and post-test to evaluate the effectiveness of an intervention based on educational robots, the experimental group may achieve better results in the post-test due to the use of educational robots in the intervention.
- the tasks should not require prior programming knowledge;

- tasks should not contain directions, as children who do not have a sufficiently developed spatial orientation, especially mental rotation skills, may be at a disadvantage when solving tasks.

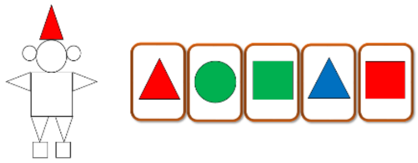
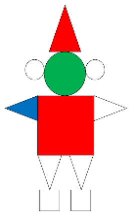
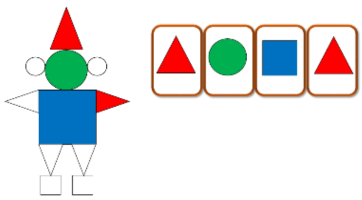
The AlgoPaint test book that we have developed is based on a storyline that is designed to motivate and sustain the motivation and interest of the children. The task is to help the Painter Elf to create shapes (robots, worms) from different geometric shapes. Instructions are given on picture cards. These picture cards are similar to the cards of the Algolittle Erasmus+ project, which are designed for creative drawing based on algorithmic thinking.

According to the storyline, the Painter Elf draws the outline of a shape made up of geometric shapes, and then paints these shapes one by one with different colours. The Elf always starts from an edge, painting the geometric shape he is standing on, then moves on to the shape that is adjacent to the one he has already painted. He moves from one shape to another, being careful not to step on a shape he has already painted as it dries slowly and he could get stuck. According to this rule, he can only go through the shapes in the order given. However, there are more correct orders in solving the task. The order chosen to solve is usually influenced by whether the algorithm can be continued or not. The rule that he cannot go back to a shape that has already been painted allows children to understand that each simple instruction refers to the current geometric shape that Painter Elf is moving on from, so that each geometric shape has a colour associated to it.

The AlgoPaint test measures the following computational thinking skills: applying an algorithm, designing an algorithm, debugging an algorithm. Table 9 provides a list of the computational thinking skills that were measured and the associated items that were used to measure each of these skills.

Table 9. Computational thinking skills measured by the AlgoPaint test

Computational thinking skill	*Description*	*Examples of items*

Applying an algorithm	The ability to identify the result of a given set of instructions	Item 1.: “Help Painter Elf, colour the robot. He laid out the instructions for you, follow them. I wonder if it is possible to paint the robot based on the cards.” 
Designing an algorithm	The ability to identify the sequence of instructions needed to achieve a given output	Item 6.: “Help Painter Elf to lay out the instructions! Painter Elf painted the robot and he sent you the picture cards with the instructions. Your task is to lay out the cards in order, just like Painter Elf painted. Here, next to him, put the cards in order in a row. I wonder if it is possible to lay out the cards as it is shown in the drawing.” 
Debugging an algorithm	The ability to compare a given set of instructions and their associated output	Item 4.: “Panter Elf has already painted the robot. He lay out the instructions. All you have to do is check that he painted the same as what is laid out on the cards. Did he lay out the cards correctly?” 

Of the items in the test, not only did the debugging tasks contain errors, but for each item we asked the children to check whether they could find an error in the task.

The test book also contained specific instructions on how to apply the rules, three solved examples, a scoring system, solutions to the tasks and picture cards for the tasks.

4.2. Preliminary testing of the AlgoPaint computational thinking assessment test

The AlgoPaint was tested in the school year 2021-2022, with 11 preschool teachers using it in their own groups, with a total of 56 children between the ages of 5 and 6. The preschool teachers were selected from the STEM course of the Master's programme. The test book was also sent to university professors who are experts in the field. We selected experts from 6 different countries to fill in an online, anonymous questionnaire about the test book. The questionnaire contained 9 questions, 5 closed questions (multiple choice and scaled) and 4 open-ended questions. The aim of the questionnaire was to find out whether the respondents thought the test items were appropriate for the target age group, whether they were suitable for measuring computational thinking, and also to get their opinion on the clarity of the explanations in the test book. The test results of the 56 preschool children who took part in the preliminary test are summarised in Table 10.

Table 10. Children's results on the AlgoPaint Computational Thinking Assessment

Item	*Computational thinking skill*	*Inclusion of errors in the task*	*Number of children who solved it correctly*	*Percentage of children who solved it correctly*	*Most frequent errors*	*Number of children committing errors*
1	Application	no	22	39.29	Painting more shapes than specified in the instructions	15
					Choosing an incorrect adjacent shape	12
2	Application	yes	23	41.07	Overlooking the error	32
3	Application	yes	25	44.64	Overlooking the error	28
4	Debugging	no	49	87.50		

5	Debugging	yes	33	58.93	Overlooking the error	23
6	Design	no	44	78.57	Putting the instructions in the incorrect order	10
7	Design	yes	30	53.57	Overlooking the error	22

The results show that the most difficult tasks were the algorithm application tasks (41.6% solved correctly), followed by the design tasks (66.07% solved correctly) and the easiest tasks were the debugging tasks (73.22% solved correctly).

In addition to the children's results, we also took into account the opinions and suggestions of the preschool teachers in the evaluation of the test. Responses from 11 educators indicated that they found the test intriguing and enjoyable, with the storyline being particularly interesting. Some found the test difficult, while others said it was suitable for preschool children. The kindergarten teachers also pointed out that more precise instructions were needed on how to administer the test, as many of them used it in groups, which made it difficult to score the results as they could not pay attention to all the children's answers at once. They also reported that the instructions for the tasks needed to be clarified and standardised, as some of the participants provided additional information with each task, making it easier for the children to complete the tasks. The preschool teachers also mentioned that many children were not able to focus on several things/objects at the same time, for example, they followed the instructions, they coloured the right shape with the right colour, but they did not take into account the rule of adjacent shapes. It was also suggested that the storyline should include the fact that the Painter Elf could make mistakes.

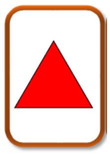
In addition to the feedback from the preschool teachers, we also analysed the review of the experts. The majority of experts believe that the test is suitable for children between the ages of 5 and 6 and that the number of items is sufficient to assess the targeted skills. At the same time, we wanted to see how well the test measured algorithm application, design and debugging: the appropriateness of the tasks was rated on a scale of 1 to 5. The average scores of the review formulated by experts were as follows: applying algorithms 4.83, debugging 4.33, algorithm design 4. One of the experts suggested using other shapes in the test, besides robots. Most of them found the test understandable for children, but recommended that a more precise explanation of the concept of adjacency should be included in the test book.

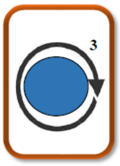
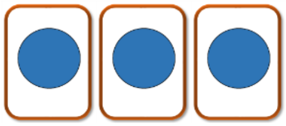
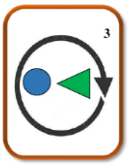
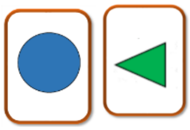
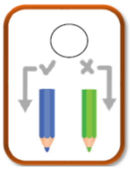
Based on the results of the preliminary test and the feedback from experts and preschool teachers, the test was redesigned. The following modifications were made to create the second version of the AlgoPaint computational thinking assessment test:

- An additional 5 items were added to the test book, measuring more / other computational thinking concepts.
- In the first version, only an example of loops was included, without a corresponding task. The new version also includes two tasks with simple loops and one task with nested loops.
- Another computational thinking concept included in the revised version of the test is the conditional: one example and a related task have been introduced.
- The scoring system has been developed in more detail
- We have included in the storyline that the Painter Elf is sometimes wrong
- We added more detail to the instructions for using the test book, so that they only need to be read out loud by the one applying the test. Our aim was to ensure that the way it was applied was consistent and that the results were not biased by the instructions given to the children.
- We also added other shapes to the book: besides robots, we created worm shapes.

Table 11 shows the computational thinking concepts used in the second version

Table 11. Computational thinking concepts in AlgoPaint version 2

Computational thinking concepts	*Description*	*Examples for cards*	*Explanation of cards*
Simple instructions	Colouring a geometric shape with a given colour		Colour the triangle red

Simple loop	A structure that allows a simple instruction to be repeated multiple times		It means the same as the following simple sequence of instructions: 
Nested loop	A structure that allows a sequence of simple instructions to be repeated multiple times		In this case, the following simple sequence of steps should be repeated three times: 
Conditionals/ conditional structure	A structure that allows the performance of a simple instruction based on a condition		We examine the condition on the top part of the card: Is the shape a circle? If yes, colour it blue, if not, colour it green.

The revised version now includes four solved examples, two for simple instructions, one for simple loops and one for conditionals. The solved examples are described verbatim, so it is sufficient to read them aloud precisely during the application of the test. This allows all children to get the same correct explanations and at the same time the test administrator does not have to add other explanations to the tasks. The AlgoPaint test book contains the correct solutions and the picture cards needed to solve each problem. The scoring system has been developed.

Conclusions

Preliminary testing of the AlgoPaint test with preschool children has shown that the test is appropriate for the target age group and suitable for measuring computational thinking skills such as algorithm application, debugging and planning. The results obtained by the children demonstrate that the test is appropriate for the level of competence of the children. Feedback from the preschool teachers who administered the test and from university professors who are experts in the development of computational thinking provided valuable ideas for improving the test. Based on the results, an improved version of the AlgoPaint Computational Thinking Assessment Tool was developed.

5. DEVELOPMENT AND PRELIMINARY TESTING OF THE INTERVENTION PROGRAMME

5.1. Developing and presenting the intervention programme

In order to integrate and apply technological skills, we have designed a programme for the development of computational thinking in preschool children. The tasks of the intervention plan focused on simple instructions, simple and nested loops, as well as conditional structures. The intervention was carried out using the Colby mouse floor robot. The activities consisted of progressively more challenging tasks, which were designed in a playful way, taking into account the specific characteristics of the age group. There were three levels of difficulty.

Level 1.

a. The child takes the mouse in his or her hand to the finish line, step by step
b. The child takes the mouse in his or her hand to the finish line, programming it

Level 2:

a. The child takes the mouse in his or her hand to the finish line, while laying out the cards with the directions on them
b. The child lays out the direction cards and uses them to program the mouse

Level 3:

a. The child only programs, without using the picture cards. There are several tasks of varying difficulty at this level.

The activities were framed by the following back story: *Mousey loves to explore, is curious about everything, always wants to see new places and is brave enough to go on adventures. When he visits the children in the kindergarten, he is always looking for a new challenge, so every time the children play with Mousey, he has to be taught a new task.*

In the intervention plan, we identified the tools needed, the form of organisation and a brief description of the steps to be taken for each activity. Once the programme was developed, I tried it out and tested it with my own kindergarten group.

5.2. Pretesting the intervention programme

The aim of our research was to find out whether our intervention would have an impact on the algorithmic/computational thinking of preschool children. The assumption was that the post-measurement results would be better than the pre-measurement results after the activities had been carried out.

Participants

We tested the programme with 6 preschool children from the middle and upper group: two children from the middle group (4 years old): one boy and one girl, and four children from the upper group (5 years old): two boys and two girls. The intervention took place in December 2022.

Assessment tools

Before the intervention, we used two instruments to assess the participants' computational thinking. Both tools were applied individually. The first assessment tool, the AlgoPaint (Algofest), is a paper-and-pencil-based instrument designed to evaluate three key competences: algorithm step tracking, algorithm creation and debugging (Bálint-Svella and Zsoldos-Marchis, 2023). It also includes tasks that measure simple and nested loops and if-then conditionals.

The second test is The Competent Computational Thinking Test (cCTt), developed by El-Hamamsy and colleagues in 2022. From the latter, we included items that are also found in our own test book: algorithm tracking, simple and nested loop tasks and an if-then conditional structure task.

The process of the intervention

After learning the back story, the children performed 9 different tasks with the floor robot.

Results

Evaluation of participants' pre- and post-test scores on the AlgoPaint test

In the first three tasks (I.1.a, I.1.b, I.1.c) the children had to colour according to the instructions. In two of the items, the order of the instructions was not correct, and the participants had to notice this. In the pre-test, 3 children solved the first item correctly, 2 the second and 2 the third. Reasons for errors: some children coloured at random or did not take into account the adjacency clause. In the post-test, the number of correct solutions to the same items is as follows: the first and second items were solved correctly by 4 and the third by 5. The reason for error here was the disregard of the adjacency clause.

In the next two exercises (I.2.a, I.2.b) they had to decide whether the algorithm was right or wrong, i.e. whether the pre-coloured shape was coloured according to the instructions given. In the pre-test, 5 children answered the first task correctly, and only 1 child answered the second correctly. Those who did not get it right did not notice that the order had been changed, and they simply matched the cards to the shapes. In the post-test, the number of correct answers was as follows: the first item was solved correctly by all children and the second item was solved correctly by 3 children.

The next two tasks (I.3.a, I.3.b) asked the children to create algorithms: they had to put the instructions in the right order for the diagrams they had created. During the pre-test, the first item was solved correctly by 4 children, while the second item (where the error in the diagram had to be noticed) was not solved correctly by any child. The reason for this is that they did not take into account the rule that it was not possible to return to a shape that had already been coloured. In the post-test, all the children were able to solve the first item correctly and one child was able to solve the second item correctly.

The next three items (II.a, II.b, II.c) consisted of simple and nested loop tasks: the children were required to identify the repetition in the instructions and colour accordingly. In the pre-test, 4 children answered the first and second items correctly, and 5 children answered the third item correctly. The error was due to a failure to recognise the loop structure card. In the post-test, all the children solved all the items correctly.

The last two tasks (III.a, III.b) contained if-then conditional structures: the children had to colour on the basis of their recognition of the cards and decide whether the instructions for the coloured pattern were laid out correctly. In the pre-test, no child could solve the first item; one child gave the correct answer to the second item without being able to explain why the set of instructions was correct. However, in the post-test there were 4 correct answers to the first item and 3 correct answers to the second item.

Comparing pre- and post-test scores

A few weeks after the intervention (in January 2023), the two tests were administered again, in the same format and under the same conditions as the first time, during the pre-test. The results are summarised in the table below.

Table 12. Pre- and post-test scores in the two tests

Participants	*Algopaint* *results in %*			*Cct-test* *results in %*		
	Pre-test	Post-test	Difference	Pre-test	Post-test	Difference
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.	33.33	83.33	50.00	66.66	83.33	16.66

The data shows that there were differences between the pre- and post-test scores for both tests. In the AlgoPaint test, all of the children performed better, i.e. they gave more correct answers in the post-test. In the case of the cCTt test, 4 of the children also improved their rate of correct answers, but 2 of the children gave fewer correct answers in the post-test. These two children performed better on the AlgoPaint test in the post-test: one gave 8.33% more correct answers and the other had 50% more correct answers in the post-test than in the first assessment. In terms of the percentage of results, it can be stated that the results of the AlgoPaint test were positively influenced by our intervention, with an increasing trend in the number of correct answers.

Development of the final version of the intervention programme

After piloting the intervention, we revised the intervention programme as it became clear on several occasions during the activities that they were not appropriate for the preschool age group. In the revised version, we took into account the results of my own observations, we completed the process and described the tasks in much more detail. A total of nine activities were developed. In the final version, we kept the three levels of difficulty based on the principle of gradience. We also kept the back story. Each description of the activity starts with a picture that shows the shape of the track that has to be built. In the new version, every task has been elaborated in great detail. Each visual aid is also included as an appendix in the programme booklet. The level of detail in the revised programme guide allows the user to be objective and to carry out the tasks with precision.

6. IMPACT ASSESSMENT OF THE INTERVENTION PROGRAMME

6.1. METHODOLOGY

Hypotheses:

Our study investigated whether the intervention programme had an effect on preschoolers' computational thinking.

H1. It is hypothesised that there are differences in computational thinking abilities between children as early as preschool, depending on their age.

H2. A STEM-based programme using educational robots helps develop computational thinking skills in preschool children.

H3. Through a STEM-based programme using educational robots, preschool children can learn programming concepts that they were less familiar with in the pre-testing phase.

Participants

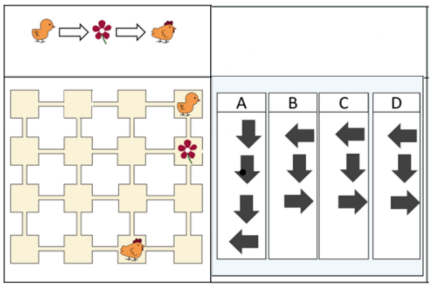
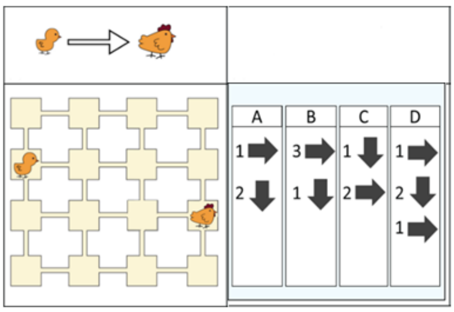
A total of 41 preschool children from three kindergarten groups took part in the study. All of the children were included in the experimental group. Given that the development of computational thinking skills is not a priority in preschool education, we did not use a control group comparison. The robot-based intervention programme was not compared with other methods of development; we assessed the children's development before and after the intervention in relation to themselves. In terms of age, 6-year-olds were the majority: 59% (23) of participants were 6-year-olds and 41% (16) were 5-year-olds. The kindergarten teachers were asked prior to the intervention not to select children under the age of 5 for the survey. The preschool groups included in the study were two rural kindergarten groups and one urban kindergarten group. The groups were heterogeneous in terms of the age of the participants.

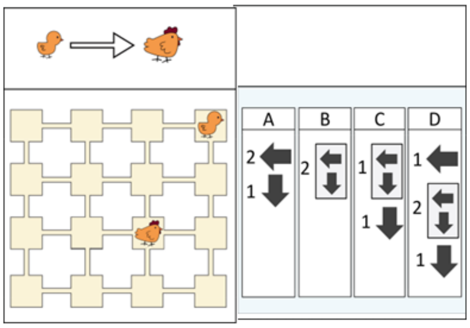
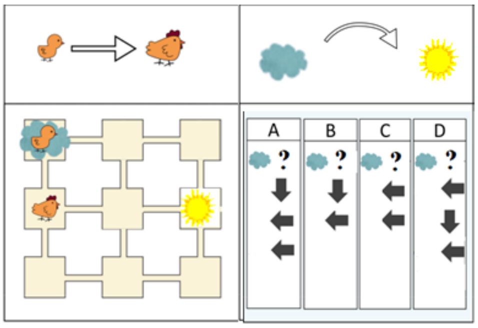
Assessment tools:

1. **AlgoPaint**: a test we developed to assess computational thinking in children aged 4 to 7 (Zsoldos-Marchis and Bálint-Svella, 2023).

2. **cCTt**: El-Hamamsy and colleagues (2022) validated the cCTt test, an instrument for the measurement of computational thinking in children between 7 and 9 years of age. We adapted some items from this test, creating a shortened version that we used in our research. The test book we designed contained a total of 6 tasks. The items covered the same computational thinking concepts as the items in the AlgoPaint test book: simple instructions, simple loops, nested loops and conditional structures. And for computational thinking skills, we measured the application of algorithms. At the beginning of the test book there is a guide with detailed instructions on what to do. In the test book, the tasks were multiple choice, meaning that the participants had to find and choose the correct answer from four options. The score for each task was therefore a 0 or a 1. Table 13 shows the concepts of computational thinking for which we developed the items, as well as examples of these items.

Table 13. Computational thinking concepts in the adapted version of the cCTt

Computational thinking concepts	*Description*	*Examples*
Simple instructions	A set of instructions to guide the chick to its mother on the square grid using the arrows.	
Simple loop	A structure that allows an instruction to be repeated several times.	

Nested loop	A structure that allows a sequence of instructions to be repeated multiple times	
Conditionals/ conditional structure	A structure that allows the performance of an instruction based on a condition	

The process of the intervention

The intervention programme was carried out in the school year 2022-2023. As a first step, we met with the preschool teachers, who were given a detailed explanation of the test books they were to use with their groups. The test books were presented one by one, with detailed instructions on how and when to apply and use them. We made separate folders of both tests for each child, which we delivered to the kindergartens in the autumn of 2022. Following the completion of the pre-testing phase in each group, another meeting was held with the teachers, one by one, during which the intervention programme was presented in detail. This involved a comprehensive review of all the activities and a demonstration of the visual aids. All teachers had the opportunity to test the robots, construct the track and carry out various programming tasks. We delivered the robots to each group and kept in contact with the teachers throughout the intervention. After the intervention, in May 2023, the test books for the post-test phase were distributed to all kindergartens in separate folders for each child. After the administration of the post-tests, we collected the test books from all three groups.

6.2. RESULTS

We first checked whether there were statistically significant differences between the groups, both before and after the intervention, in order to rule out the effect of different kindergarten teachers delivering the intervention in the groups. To investigate this, we used an ANOVA test. The results are summarised in Table 14.

Table 14. The difference in performance between the three groups

Test	*group 1.*		*group 2.*		*group 3.*		*F*	*p*
	mean	*SD*	*mean*	*SD*	*mean*	*SD*		
AlgoPaint pre-test	23.231	6.894	26.250	6.962	22.417	7.025	1.214	.308
AlgoPaint post-test	30.308	7.052	32.313	4.222	34.667	2.774	2.376	.107
cCTt pre-test	4.538	1.506	4.313	1.621	4.167	1.267	0.199	.820
cCTt post-test	5.846	0.376	5.813	0.403	5.917	0.289	0.284	.754

The table shows that there is no significant difference between the pre- and post-test scores of the three groups for either the AlgoPaint or the cCTt test. This result shows that the fact that the intervention was delivered by different teachers to different groups did not have an impact on the outcome.

In the following, we analyse the results on the basis of our hypotheses.

H1. *It is hypothesised that there are differences in computational thinking abilities between children as early as preschool, depending on their age.*

We compared the test scores of children of different ages using an independent samples t-test to see how children's performance varied with age. The results are summarised in Table 15.

Table 15. Pre- and post-test scores by age

Test	*5-year-olds*			*6-year-olds*			*t*	*p*
	mean	*SD*	*N*	*mean*	*SD*	*N*		
AlgoPaint pre-test	23.94	6.62	18	24.348	7.420	23	-0.181	.857
AlgoPaint post-test	31.11	6.21	18	33.348	4.052	23	-1.392	.722
cCTt pre-test	4.38	1.14	18	4.304	1.690	23	0.182	.857

cCTt post-test	5.83	0.38	18	5.870	0.344	23	-0.318	.752

The results show that the mean scores of 6-year-old children are higher on the AlgoPaint test for both the pre- and post-test phases; however, there is no statistically significant difference in the performance of 5- and 6-year-old children.

H2. *A STEM-based programme using educational robots helps develop computational thinking skills in preschool children.*

First, we analyse the results of the **AlgoPaint test**.

The total test score was divided by the number of items in order to obtain the mean score per item, which was then used in statistical processing. A comparison was made between the pre-test and post-test results for the experimental group. The mean of the post-test results is 0.67 higher than the mean of the pre-test results (see Table 16). The Shapiro-Wilk test shows a normal distribution, so when the means are compared using a paired t-test, the improvement is statistically significant. Also, the effect size is high, Cohen's d = 1.08.

Below is a comparison of the pre-test and post-test scores for each type of task.

For the *simple instructions* type tasks, the mean of the post-test scores is 0.74 higher than the mean of the pre-test scores. The Shapiro-Wilk test shows a normal distribution in this case as well, so a paired t-test was used to compare the data, showing significant differences between the pre- and post-test results. Also, the effect size is high, as Cohen's d = 0.983.

The difference between the pre- and post-test means for the *loop* tasks is 0.24. As the Shapiro-Wilk test shows a deviation from the normal distribution, the Wilcoxon test was used, which shows a statistically significant difference between the pre-test and the post-test scores. The effect size for this task type is low, as Cohen's d is 0.462.

For the *conditional structure* tasks, the mean of the post-test scores is 1.1 times higher than the average of the pre-test scores. As the Shapiro-Wilk test shows a deviation from the normal distribution in this case as well, the Wilcoxon test was used, which shows a statistically significant difference between the pre-test and the post-test. The effect size is high for the conditional structures.

Table 16. Comparison of pre-test and post-test scores of the AlgoPaint test

Algo-Paint	Pre-test		Post-test		Shapiro-Wilk normality test		Test	t / W (statis-tics)	p	Effect size
	Mean	SD	Mean	SD	W	p				
Average of the total test score per item	2.00	0.57	2.67	0.44	0.982	.753	Stu-dent	6.990	<.001	1.079
Simple instructions	2.04	0.68	2.78	0.39	0.983	.778	Stu-dent	6.373	<.001	0.983
Loop	2.34	0.59	2.58	0.72	0.917	.005	Wil-coxon	133.5	0.024	0.462
Conditional structure	1.34	1.19	2.44	0.83	0.933	.017	Wil-coxon	52	<.001	0.815

In the following, the results of the **adapted and revised cCTt test** are analysed. In this case as well, the total test score was divided by the number of items in order to obtain the mean score per item, which was then used in statistical processing. The mean of the post-test results is 0.25 higher than the mean of the pre-test results (see Table 17). As the Shapiro-Wilk test shows a deviation from the normal distribution, the Wilcoxon test was used, which shows a statistically significant difference between the pre-test and the post-test. Cohen's d in this case is 0.913, indicating a high effect size.

As in the case of the AlgoPaint test, we analysed the scores of the pre- and post-tests of the cCTt test for the different types of tasks. For all three task types, the Shapiro-Wilk test showed a non-normal distribution, so we used Wilcoxon. The results show that for all three task types, there is a significant difference between the pre- and post-test scores and that the effect size is high.

Table 17. Comparison of pre-test and post-test scores of the cCTt test

cCTt:	Pre-test		Post-test		Shapiro-Wilk normality test		Test	W (statis-tics)	p	Effect size
	Mean	SD	Mean	SD	W	p	Wil-			

							coxon			
Average of the total test score per item	0.71	0.24	0.96	0.11	0.932	.015	Wil-coxon	26.000	<.001	0.913
Simple instructions	0.85	0.23	0.97	0.11	0.680	<.001	Wil-coxon	10.000	.011	0.780
Loop	0.61	0.32	0.96	0.13	0.832	<.001	Wil-coxon	11.000	<.001	0.937
Conditional structure	0.50	0.50	0.90	0.29	0.702	<.001	Wil-coxon	10.000	<.001	0.895

H3. Through a STEM-based programme using educational robots, preschool children can learn programming concepts that they were less familiar with in the pre-testing phase.

In the following, we compare the difficulty of three programming concepts (simple structures, loops and conditionals) using a pooled sample analysis of variance.

We start by analysing the pre-test results collected using the *AlgoPaint test*. Since Mauchly's test of sphericity is not satisfied (χ^2 (2) = 10.470, p = .005), a Greenhouse-Geisser correction was used. The result shows that there is a statistically significant difference between the results of the 3 programming concepts (F(1.626) = 20.321, p < .001). Holm's post hoc analysis shows that the conditional structures are statistically significantly more difficult than simple instructions or loops (Table 18).

Table 18. AlgoPaint pre-test scores for different task types

	Mean	SD		Mean	SD	t	p
Simple instructions	2.04	0.68	Loops	2.34	0.59	-1.820	.072
Simple instructions	2.04	0.68	Conditional structure	1.34	1.19	4.381	<.001
Loops	2.34	0.59	Conditional structure	1.34	1.19	6.201	<.001

We now analyse the *post-test* results collected with the *AlgoPaint test.*

As Mauchly's test of sphericity is not satisfied in the case of the post-test (χ^2 (2) = 5.584, p = .005), a Greenhouse-Geisser correction was applied here as well. The results show a statistically significant difference in the scores of the programming concepts (F(1,769) = 4.593, p < 0.05).

Holm's post hoc analysis shows that the conditional structure (M = 2.440, SD = 0.835) is significantly more difficult than simple instructions, but there is no statistically significant difference between conditional structures and loops (Table 19). This shows that the intervention helped the children to understand how to use the conditional structure.

Table 19. AlgoPaint post-test scores for different task types

	Mean	SD		Mean	SD	t	p
Simple instructions	2.78	0.39	Loops	2.58	0.72	1.808	.149
Simple instructions	2.78	0.39	Conditional structure	2.44	0.83	3.011	.010
Loops	2.58	0.72	Conditional structure	2.44	0.83	1.203	.232

As before, for the *adapted cCTt test*, we compared the *pretest* scores for the three programming concepts using a pooled sample analysis of variance. As Mauchly's test of sphericity is not satisfied (χ^2 (2) = 8.170, p = .005), a Greenhouse-Geisser correction was applied. The results show a significant difference in the scores of the different programming concepts (F(1,688) = 15.928, p < 0.001). Holm's post hoc analysis shows (Table 20) that simple instructions are significantly less difficult than conditional structures and loops. However, there is no statistically significant difference between loops and conditional structures.

Table 20. cCT test pre-test scores for different task types

	Mean	SD		Mean	SD	t	p
Simple instructions	0.85	0.23	Loop	0.61	0.32	3.698	<.001
Simple instructions	0.85	0.23	Conditional structure	0.50	0.50	5.542	<.001
Loops	0.61	0.32	Conditional structure	0.50	0.50	1.844	.069

In the *post-test of the adapted cCTt* we used the same statistical procedures as in the pre-test, i.e. we compared the results obtained for the three programming concepts using a pooled sample analysis of variance. As Mauchly's test of sphericity is not satisfied (χ^2 (2) = 33.314, p = 0.001), a Greenhouse-Geisser correction was applied. The results do not show a significant difference in the scores of the different programming concepts ($F(1,278) = 2.091$, $p < 0.150$).

The results indicate that the intervention programme helped children to understand the conditional structures, which initially proved difficult, and that there were smaller differences in the scores obtained on different types of tasks in the case of the post-test.

7. CONCLUSIONS ANDPOSSIBLE FURTHER DEVELOPMENT

My thesis presents my research and my findings on the cognitive development of children in the preschool age group. More specifically, our study focused on the investigation of computational thinking skills and their development in preschool children.

1. *Questionnaire surveys:*

Since we planned to design development tasks related to the technological area of STEM activities, we first created a questionnaire about these activities, which was completed by 85 students enrolled in the Pedagogy of Primary and Preschool Education programme.

The responses to the questions indicated that students integrate a range of experience areas into their teaching practice, which aligns with the curricular expectations. Furthermore, it was observed that the most effective approach involved integrating mathematical and natural science activities, which is noteworthy given that both are part of STEM activities. A surprising finding was that more than half of the students were unfamiliar with the term "STEM" and therefore had no personal experience with these types of activities. Nevertheless, most of them believe that STEM activities encourage children to think creatively in the field of science. According to students, STEM activities develop a wide range of cognitive, emotional, social and physical skills and abilities, such as creative and exploratory thinking, teamwork, as well as independence, sensory exploration, fine and gross motor skills. Regarding the potential of applying STEM activities in preschool education, most respondents emphasised that such activities allow children to gain hands-on practical experience, enrich their knowledge in the technical field and in relation to the natural environment, and provide positive emotions and high motivation in different areas of science.

The second part of the questionnaire addressed the issue of technological skills and their use in the context of preschool education, as well as the possibilities of developing algorithmic and computational thinking skills. In addition to the 85 students, this part of the questionnaire was completed by 115 practicing preschool teachers. There was a consensus among students and practicing preschool teachers that technological skills embedded in preschool activities would develop positive attitudes in children towards this particular field. In the responses of both students and preschool teachers, the lack of adequate methodological training and the lack of appropriate didactic tools were identified as the main obstacles in the planning and carrying out of such activities. With regard to algorithmic/computational thinking, both students and preschool teachers considered it important to start developing these skills in kindergarten. However, they had little knowledge of how to develop these skills: educational robots were mentioned as an option by only a few of the respondents. None of the students had ever used such devices in their preschool teaching practice, and only four of the preschool teachers reported having used floor robots in their kindergarten activities. In light of the findings, it becomes evident that there is a necessity for the implementation of training programmes and workshops that facilitate the practical application of technological knowledge within the context of early childhood education.

2. *AlgoPaint test book:*

In order to evaluate the algorithmic and computational thinking abilities of preschool children, we developed a paper-and-pencil assessment tool that did not require the use of a computer or other technological devices. Following the pilot testing of the instrument with 56 preschool children (aged 5 and 6) by 11 kindergarten teachers, the test was also sent for evaluation to university professors in different countries who are experts in the field. The preschool teachers, the scores of the children and the opinions of experts all indicated that the test was an appropriate and suitable means of assessing the algorithmic and computational thinking skills of the preschool age group. Taking into account the suggestions and opinions of preschool teachers and university professors experienced in the development of algorithmic and computational thinking, we revised and improved the test book and thus created the AlgoPaint Computational Thinking Assessment Tool. This tool addresses a gap in the existing research and can be implemented in early childhood education settings in any country by simply translating the instructions.

3. *Development programme:*

The pretesting of the developed intervention plan, although carried out with a small group of children, proved to be an adequate impact study based on the children's scores. We corrected the errors that were identified during the pre-testing phase and we revised the tasks that proved to be difficult. The pre- and post-test data show that the AlgoPaint test was sensitive to the changes facilitated by the intervention, as the programme had a positive effect on the post-test scores. The findings indicate that computational thinking can be effectively introduced to preschool children through the implementation of targeted educational programmes. The results of our impact study show that educational floor robots can be an effective tool for developing computational thinking in preschool children.

Limitations of the study:

We have identified some limitations in the research, which are summarised below:

- due to the relatively small number of students and preschool teachers who participated in the questionnaire survey (85 students and 115 preschool teachers), our results cannot be generalised to the Romanian education system. Moreover, we realised that we had included relatively few questions about educational floor robots in the questionnaire.
- the absence of a control group introduced the possibility that we could not exclude and control natural development as a possible influencing factor in the design of the intervention. However, as the development of algorithmic and computational thinking is not part of the preschool programme, we are not specifically targeting the development of skills in this area and that the innovative nature of the intervention design and the tool used may explain the differences between the pre- and post-test results.
- regarding the survey instruments employed, it is not possible to consider the adapted cCTt-test as a validated comparative tool given that it was not administered in its entirety.
- only 41 preschoolers participated in the development, which is a relatively small sample.

An analysis of the limitations reveals possibilities for further improvements. In our case, the following directions arise:

- the implementation of the AlgoPaint test book validation procedure on a new sample, using the full cCT test as a validated assessment tool.
- the creation of an alternative version of the AlgoPaint assessment tool, containing exercises and items designed to inhibit the transfer effect.

REFERENCES

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., & Magdalean, 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/subbpsyped.2022.1.07*

18. Bálint-Svella, É. K., & Zsoldos-Marchiș, 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., Chioccariello, 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. Brenneman, 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., Martin 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: Vlados.

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 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., & Yıldırım, 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 Educaţiei Naționale. (2019). *Koragyermekkori Nevelés Curriculuma*.
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., Sárik, 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-Villagrá, 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., Hambrusch, 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