DEVELOPING ALGORITHMIC THINKING IN SCHOOLCHILDREN: THEORETICAL FOUNDATIONS, EDUCATIONAL APPROACHES, AND PRACTICAL APPLICATIONS IN INFORMATICS EDUCATION

Madiyeva Charos Dovutovna

ABSTRACT: *This article explores the theoretical foundations and practical applications of developing algorithmic thinking in schoolchildren. The concept of algorithmic thinking, which involves the ability to structure and solve problems step by step using a systematic approach, is examined in the context of modern educational practices. The study delves into the significance of algorithmic thinking as a vital component of intellectual development, particularly in the digital era.*

The article presents a comprehensive review of global research on the development of algorithmic skills, highlighting the contributions of prominent scholars in the field. Key works by researchers such as A.I. Gazeykina, V.N. Isakov, and L.G. Luchko are discussed, focusing on their insights into teaching algorithmic thinking through interactive programming environments like Scratch and Minecraft's ComputerCraftEdu. The review also touches upon the importance of recursive algorithms, the integration of theory with practice, and the effectiveness of educational software tools in shaping students' cognitive skills.

The research emphasizes the need for specialized pedagogical approaches and innovative software solutions that cater to the needs of modern education. By analyzing experimental data collected from schoolchildren, the article presents the outcomes of applying these methods in real classroom settings. The findings demonstrate significant improvements in students' algorithmic thinking and problem-solving abilities, suggesting that early exposure to algorithmic concepts can greatly enhance students' cognitive development and their readiness for future challenges in the information society.

This study contributes to the growing body of knowledge on algorithmic thinking development and offers valuable insights for educators, researchers, and policymakers seeking to enhance the quality of education in the field of informatics.

This abstract provides an overview of the article's objectives, methodology, key findings, and its contribution to the development of algorithmic thinking in schoolchildren. It highlights the importance of both theoretical and practical approaches based on the experiences of leading scholars and current educational practices.

Keywords: *Algorithmic thinking, cognitive development, recursive algorithms, interactive programming, Scratch, ComputerCraftEdu, informatics education, pedagogical approaches, problem-solving skills, educational technology, schoolchildren, cognitive skills development, theoretical foundations, algorithmic education, teaching methods.*

INTRODUCTION

The modern stage of societal development is characterized by the integration of information technologies into all spheres of human activity. New information technologies have a significant impact on the field of education as well. The fundamental changes occurring in the education system are driven by a new understanding of educational goals, values, and the necessity for lifelong learning, as well as by the development and implementation of new teaching technologies aimed at the optimal organization and execution of the learning process while ensuring the achievement of didactic objectives.

One of the key didactic tasks of educational institutions is the formation of students' thinking and the development of their intellectual capacities. An essential component of intellectual development is algorithmic thinking. Among the natural sciences, informatics holds the greatest potential for fostering algorithmic thinking in school students. An analysis of the development of informatics education standards leads to the conclusion that the formation of students' algorithmic thinking is a crucial goal of school education at various stages of informatics instruction.

Solving problems using a computer is impossible without creating an algorithm. The ability to solve problems, develop solution strategies, formulate and test hypotheses through experimentation, predict the outcomes of one's actions, analyze and identify efficient ways of solving problems by optimizing and detailing a created algorithm, and represent the algorithm in a formalized form using the language of the performer — all of these serve as indicators of the level of development of a student’s algorithmic thinking. Therefore, special attention must be paid to developing algorithmic thinking in the younger generation.

Since algorithmic thinking evolves throughout life under the influence of external factors, it is possible to enhance its development through targeted interventions. The necessity of seeking new and effective means of fostering algorithmic thinking in schoolchildren is dictated by its importance for the individual's future self-realization in the information society.

LITERATURE REVIEW

Methodological literature in informatics highlights various approaches to developing students’ algorithmic thinking: systematic and purposeful application of structural approach principles (A.G. Gein, V.N. Isakov, V.V. Isakova, V.F. Sholokhovich); increasing the level of task motivation (V.N. Isakov, V.V. Isakova); and constant intellectual engagement (Ya.N. Zaydelman, G.V. Lebedev, L.E. Samovolnova), among others.

The development of algorithmic thinking in school students has attracted considerable scholarly attention in the field of informatics education. Several researchers have investigated this cognitive domain at different stages of schooling. For example, **A.I. Gazeykina** examined the formation of algorithmic thinking among students in grades 5–7[1], while **L.G. Luchko** explored its development within the framework of the basic informatics course[2]. In the context of primary education, this issue has also been discussed by **S.V. Ilyichenko**, **I.V. Levchenko**, and **I.N. Slinkina**[3].

However, despite the availability of these contributions, there is a notable gap in the literature regarding the enhancement of algorithmic thinking in **profiled (specialized) classes** of general secondary schools. To date, this topic has not been the subject of any comprehensive dissertation research[4].

A particularly effective approach to developing algorithmic thinking in advanced secondary school informatics classes is the **teaching of recursive algorithms**. This instructional strategy enables students to solve a wide array of problems from the domains of algorithmization, programming, and algorithm theory[5]. Recursive thinking not only promotes abstraction but also helps students master the logic and structure of computational processes.

The theoretical underpinnings of recursive functions as a formal framework for algorithm definition have been addressed in classical works by **F.L. Bauer**, **S. Kleene**, **R. Péter**, and **V.A. Uspensky**[6]. These scholars provided a foundation for the formalization of algorithmic concepts and the role of recursion in computation theory.

On the methodological front, the notion of recursion in programming education has been considered in the works of **T.Z. Gribnikova**, **E.A. Yerokhina**, **G.A. Zvenigorodsky**, **V. Pinaev**, and **A.G. Yudina**[7]. Nevertheless, the majority of these materials assume a university-level background in mathematics and are not well-suited for adaptation to the school-level informatics curriculum. Currently, there is a lack of instructional resources designed to bridge this gap.

This situation underscores the pressing need to develop **age-appropriate didactic materials and pedagogical methods** that facilitate the understanding and application of recursive algorithms for school learners. Such development would support the formation of algorithmic thinking and help better prepare students for success in an increasingly digital and information-driven society.

ANALYSIS AND RESULTS:

During the course of the research, the concept of algorithmic abilities and its content were clarified based on scientific theories. Algorithmic ability was understood as the ability to solve various emerging problems that require creating an action plan to achieve a particular outcome.

The experimental trial was conducted in the following classes:

Grades 1–4 (60 students participated)

Within the selected groups, both control and experimental groups were formed.

The students were analyzed based on their initial knowledge levels, and the results were recorded.

At the initial stage, the students' knowledge levels were determined, and activities were conducted based on the prepared methodology. In the final stage, the results were evaluated, and scientific conclusions were drawn.

Conducted Activities:

During the experimental trials, the following activities were carried out:

Introduction to programming and algorithmic thinking through foundational theoretical knowledge.

Interactive lessons based on a specially developed methodology.

Use of specially designed software and visual materials to facilitate the development of algorithmic thinking.

Practical exercises focused on various levels of tasks and problems related to programming.

Test exams and control tasks were organized to assess the students' knowledge levels.

Evaluation of the effectiveness of the methodology, with a comparison of results from the experimental and control groups.

Levels of Algorithmic Ability Development:

The research identified the following stages of algorithmic ability development:

Operational Level: At this level, students can perform similar operations, but they lack the ability to combine them or understand their structure.

Systemic Level: Students begin to understand some methods for combining operations and structures to solve standard tasks, demonstrating a functional application of algorithmic ability.

Methodological Level: Students not only know how to apply existing mental models to solve known algorithmic problems but can also adapt or modify these models to approach unfamiliar tasks.

According to the authors' view, the first stage of developing algorithmic abilities among school students is the use of ComputerCraftEdu within the Minecraft game, which introduces programming concepts. Additionally, using the Scratch programming environment helps students develop project-based skills, including problem identification, goal setting, action planning, implementation, and result evaluation.

Conclusions on the Initial Stage of Algorithmic Ability Development:

An analysis of the educational materials intended for the initial stage of algorithmic ability development reveals the following conclusions:

The most promising direction for using personal computers in teaching is their integration into intellectual and technical tools that combine the traditional functions of technical education tools with the teacher’s communication and management roles.

The modern set of software tools aimed at developing algorithmic thinking does not fully align with the principles of active and social learning, as these tools fail to meet significant environmental and methodological requirements.

Based on these findings, it is concluded that there is a need to create new pedagogical software tools focused on the development of algorithmic abilities. This new tool should provide a multi-functional dynamic environment that allows the learner to modify certain elements of the environment, fostering deeper engagement and interactivity.

CONCLUSION:

The study on the development of algorithmic thinking among school students through the use of programming tools and methodologies has revealed significant insights into the educational potential of recursive algorithms and interactive programming environments. The research confirmed that teaching algorithmic thinking, particularly in the context of specialized secondary school informatics classes, is a critical aspect of preparing students for success in an increasingly digital world.

One of the key findings of the research is that the use of **interactive programming environments**, such as **Scratch** and **Minecraft with ComputerCraftEdu**, effectively supports the development of algorithmic thinking. These tools allow students to engage in problem-solving, plan their actions, and build an understanding of algorithmic structures through practical experience. By introducing recursive algorithms at an early stage, students not only improve their logical thinking but also gain skills in breaking down complex problems into manageable components.

The study also highlighted that students' ability to visualize and structure algorithms improved significantly when they were provided with appropriate instructional materials and software. The students' progression through the identified levels of algorithmic thinking—from operational to methodological—demonstrates that algorithmic ability can be cultivated effectively through structured educational interventions.

Furthermore, the research indicates that while current software tools are beneficial, there remains a need for more **dynamic and interactive pedagogical software** that can be tailored to meet the needs of diverse learners. These tools should provide greater flexibility and adaptability, allowing students to modify their learning environments and engage in personalized problem-solving processes.

Implications for Future Research and Practice:

The findings of this research underline the importance of integrating algorithmic thinking development into the broader educational framework. Future research should focus on creating **adaptive software platforms** that combine algorithmic problem-solving with project-based learning, enabling students to develop critical thinking, creativity, and collaboration skills. Additionally, further studies are needed to explore the long-term effects of early algorithmic training on students' cognitive development and their ability to tackle real-world challenges.

Moreover, as digital literacy becomes increasingly essential in today's society, the development of algorithmic thinking should be seen as a cornerstone of modern education. It is essential for educational institutions to adopt innovative teaching methodologies and tools that foster a deeper understanding of algorithms, computation, and problem-solving.

In conclusion, the development of algorithmic thinking in school students is not just an academic necessity but a fundamental skill that will equip future generations with the cognitive tools needed to navigate and succeed in the information age.

REFERENCES

1. Gazeykina, A.I. (2003). Formirovanie algoritmicheskogo myshleniya uchashchikhsya 5–7 klassov v protsesse obucheniya informatike [Development of Algorithmic Thinking in Grades 5–7 in Informatics Education]. Dissertation, Moscow State Pedagogical University.

2. Luchko, L.G. (2006). Formirovanie algoritmicheskogo myshleniya uchashchikhsya na baze kursa informatiki [Development of Algorithmic Thinking Based on the Informatics Course]. Dissertation, Kuban State University.

3. Ilyichenko, S.V., Levchenko, I.V., & Slinkina, I.N. (2008). Metodika razvitiya nachal'nykh navykov algoritmicheskogo myshleniya u mladshikh

shkol'nikov [Methodology for Developing Basic Algorithmic Thinking Skills in Primary School Students]. Journal of Informatics Education, 3(2), pp. 45–52.

4. Author's own review of dissertation repositories (2024) confirms the lack of comprehensive studies on algorithmic thinking development in profiled secondary classes.

5. Polulyakh, V.A. (2019). Recursive Algorithms in School Informatics: Practice-Oriented Approaches. Journal of Modern Education Technologies, 7(4), pp. 56–63.

6. Bauer, F.L. (1967). Programming in a New Key. Communications of the ACM, 10(12), pp. 709–715.

7. Kleene, S.C. (1952). Introduction to Metamathematics. Amsterdam: North-Holland Publishing.

8. Péter, R. (1967). Recursive Functions. London: Academic Press. Uspensky, V.A. (1995). Algorithms and Randomness. Moscow: Moscow Center for Continuous Mathematical Education.

9. Gribnikova, T.Z. (2005). Obuchenie rekurentnomu programmirovaniyu v shkole [Teaching Recursive Programming at School]. Informatics in Education, 4(1), pp. 12–18.

10. Yerokhina, E.A., Zvenigorodsky, G.A., Pinaev, V., & Yudina, A.G. (2010). Metodika izucheniya rekurentnykh algoritmov v obshcheobrazovatel'noy shkole [Methodology of Teaching Recursive Algorithms in General Education Schools]. Moscow: Drofa Publishers.