Informatics: Logic, Algorithms, and Working with Information

Aleksandr Kleymenov

Fractal Academy
Issue 4 (Autumn 2026), Article 12
Series: “Developing Mental Mechanisms Through School Subjects”
Author: Aleksandr Kleymenov
ORCID: 0009-0004-9404-2400
DOI: [to be assigned]
Publication date: 1.11.2026

© 2026 Aleksandr Kleymenov. This work is licensed under the terms of Creative Commons Attribution-ShareAlike 4.0 International License.
Title (English): «Informatics: Logic, Algorithms, and Working with Information»
URL: https://fractal-academy.com/articles/article-4-12/
(CC BY-SA 4.0)

Annotation

In this article, I show how informatics as a school subject can become the foundation for the development of mental mechanisms in the digital age - provided that we stop reducing it to teaching work with office programs and memorizing the syntax of programming languages. Drawing on the CCFP model presented in previous articles of the module, and on current research of 2024-2025 in the field of informatics education, I substantiate that informatics possesses a unique potential for the development of the construction of cause-and-effect relationships, critical thinking, information search and analysis, forecasting and strategic thinking, as well as spatial imagination and long-term memory. I propose specific examples of lessons and tasks in which algorithmization, programming, work with data and digital hygiene serve not as an end in themselves, but as a means of developing thinking. I pay special attention to the diagnostics of progress and the connection of subject-based learning with the school's infrastructure - an IT laboratory, a robotics workshop, a media studio. I also substantiate that the restructuring of informatics teaching is not merely a methodological question, but a strategic task: it is precisely through informatics that the cognitive apparatus capable of algorithmic, computational and systems thinking is formed - the type of thinking that is becoming necessary for understanding and creating the technologies that shape our future.

Keywords: informatics, mental mechanisms, algorithmic thinking, computational thinking, critical thinking, programming, work with information, diagnostics, infrastructure.

Code That Begins with a Question

I will begin with a statement that may seem contentious, but which I consider fundamental. Informatics at school is not a subject about how to use a computer, but a subject about how logic is structured, how algorithms work, and how we can manage information in a world where there is more and more of it. It is a subject through which a person learns to think systemically - not as a set of disconnected facts, but as a structured, algorithmically organized reality. Of course, knowledge of interfaces and the ability to work with programs is important. But this is a tool, not a goal. The goal is the ability to break complex tasks into simple subtasks; the ability to critically evaluate information in the digital environment; the ability to find, analyze and structure data; the ability to forecast results and plan strategies; the ability to see abstract structures and operate with them [1; 5].

Here it is appropriate to recall Ada Lovelace, who in the 19th century worked with Babbage's machine. She saw in it not merely a calculator, but an instrument that can work with symbols, and not only with numbers. She wrote the first program - and in this gesture there was more than simply a set of commands. She asked the question: what if a machine could not only count, but also create? This question still stands before us. And informatics at school should not simply teach how to write code. It should teach how to ask such questions. Because code is an answer. And a question is a beginning. And if a child has learned a command but has not understood why it is needed, he has learned a word but has not understood the meaning. He knows what a loop is, but does not know why it is needed. He knows what an array is, but does not know how it helps to solve tasks. And this is the greatest defeat of informatics - when it becomes a set of symbols behind which nothing stands. When it ceases to be a language and becomes a cipher. Which must be deciphered. But no one knows the key. Because the key is not a command. The key is understanding. Which comes not from memorization. But from a question. From a task. From a mistake. From the surprise. That you experienced when the program worked. And then you wanted to understand. Why? And how? And what stands behind it? And this desire to understand - it is the beginning of informatics. Not commands. But surprise. Which

makes you a researcher. And not an executor. And this is the most important thing. Not the answer. But the question. That you asked. To the machine. And which it answered you. Not with words. But with a result. Which you saw. And understood. And remembered. Forever. Because it became yours. Discovered by you. Understood by you. Experienced by you. And this is precisely education. Not a sum of knowledge. But a sum of surprises. That you have experienced. And which have made you who you are. And who you will become. In the next program. In the next algorithm. In the next life.

As I have already shown in the first article of this series [5], the principle of the dual purpose of a lesson means that every informatics lesson must simultaneously solve two tasks: the subject task - the mastery of informatics content, and the meta-subject task - the development of mental mechanisms. These tasks do not contradict each other - on the contrary, they mutually reinforce each other. It is important to emphasize: informatics creates the conditions of possibility for the development of mental mechanisms, but is not their sufficient cause. In itself, memorizing syntax or working with office programs will not develop systems thinking. Development occurs only through specially organized pedagogical activity, where the teacher acts as a facilitator and architect of the cognitive apparatus [5; 6]. Research confirms: the subject of informatics allows one not only to equip a person with skills, but also to form a more systemic thinking, that is, to teach one to think critically, to conduct logically substantiated proofs and reasoning, and also to carry out analysis and synthesize new knowledge [8]. In the words of V.S. Lednev, every basic component of general education is included in the content of education in two ways - in the form of a special school subject and in the form of inclusions in all other subjects [8]. Informatics is precisely such a subject: it is not only an independent discipline, but also a universal instrument that permeates all the others. Contemporary research also emphasizes that in the last decade there has been a growing shift toward the teaching of computer science and computational thinking, starting from kindergarten [9]. However, as Marina Umaschi Bers notes, most of these efforts are dictated by the demand for a technically qualified workforce [9]. Such an instrumental approach, in my view, misses the main point. The teaching of informatics should be a humanistic enterprise that promotes self-directed learning, critical thinking and relational processes [9]. Bers's study showed that four practices are key to promoting humanistic education through programming instruction: critical thinking, effective communication, personal responsibility and social responsibility [9].

But I want to go further and show how each of the eleven mental mechanisms can be developed through informatics teaching - and how this changes the very approach to the subject. Informatics is not merely a technical subject. It is the universal language of algorithms, without which modern thinking is impossible. That is precisely why informatics, in my view, is a key subject for the formation of the communicative-cognitive framework of the personality in the 21st century [1]. Moreover, the restructuring of informatics teaching is not merely a methodological question. It is part of that historical necessity of changing the pedagogical paradigm about which I wrote in articles [3] and [4]. The industrial school reduced informatics to a set of technical skills to be memorized. But post-industrial society needs people capable of thinking, creating, adapting - and informatics here is a key instrument. As I showed in article [2], subject-centrism is one of the main reasons for the vitality of the old model. Overcoming this subject-centrism begins with a rethinking of the role of informatics in education.

Logic That Can Be Seen

The construction of cause-and-effect relationships develops through informatics in the most direct way. An algorithm is a sequence of steps where each step logically follows from the previous one and determines the next. Understanding an algorithm is understanding a cause-and-effect chain. The ability to break a complex task into a sequence of simple steps. The ability to see how a change in one step influences the entire algorithm. The ability to build a logical chain from input data to the result. The teacher must consciously focus attention on the logic of algorithms, and not only on syntax. Do not memorize the command, but explain why this command must come after that one. Do not write a program according to a model, but build a logical sequence of actions. As researchers note, programming instruction allows one to teach one to think critically, to conduct logically substantiated proofs and reasoning [8]. Informatics in this sense is an ideal trainer for cause-and-effect thinking.

It is like a recipe. You can learn the sequence of actions. But if you do not understand why each ingredient is needed, you will not be able to prepare the dish if something is not at hand. You will not be able to improvise. And a program is improvisation. Because every task is unique. And every time one must think. Not repeat. But create. And the algorithm gives you the language for this creation. A language in which you speak with the machine. And the machine

understands you. If you speak logically. If you speak consistently. If you do not skip steps. And then the program works. And you understand that you have created something that works. Not because you repeated. But because you built. With your own mind. With your own logic. By your own path. Which began with a question. And ended with a result. Which became yours. Because you built it. As a builder builds a house. From bricks. From thoughts. From logic. And this is the most important thing. Not the command. But the path. That leads to it. And that remains with you. Even when the command is forgotten. Because the path is a skill. And a skill is not forgotten. As the ability to walk is not forgotten. Or to speak. Or to think. Which you acquired. And which is now yours. Forever. And this is precisely education. Not a sum of knowledge. But a sum of skills. Which make you who you are. And who you will become. In the next task. In the next algorithm. In the next life.

Debugging as a Dialogue with Error

Critical thinking is another mechanism that informatics develops in a unique way. Informatics provides unique opportunities for the development of critical thinking. Debugging a program requires finding errors. Evaluating information from the internet requires checking reliability. Analyzing data requires interpretation and evaluation. The ability to find and correct errors in algorithms. The ability to assess the reliability of digital information. The ability to analyze data and draw substantiated conclusions. The use of problem-oriented learning. A 2025 study showed that the problem-based learning model using Scratch had a statistically significant effect on the critical thinking skills of seventh-graders [10]. The mean post-test score of the experimental group rose from 62.4 to 83.6, with a large effect size (Cohen's d) of 1.24 [10]. The greatest growth was recorded on the indicators of inference and explanation [10]. Research also confirms that informatics allows one to form the skills of systems and critical thinking [13]. The second-generation Federal State Educational Standard emphasizes that the formation of subject results alone is insufficient - personal and meta-subject results are necessary, which cannot be formed without the activity component of the learning process [13].

It is like a doctor who makes a diagnosis. He does not simply look at symptoms. He seeks the cause. He tests hypotheses. He makes mistakes. And seeks again. And in the end finds it. Or does not find it. But he does not stop. Because every mistake is a step toward the truth. So too in informatics. A student who finds an error does not simply correct the code. He understands why the error arose. And how to avoid it in the future. And this is the most important thing. Not the correction. But the understanding. Which makes you not an executor. But a researcher. Who knows how to see errors. And correct them. And go further. To a new code. To a new task. To a new life. Which always begins with the first step. That you take. Now. Here. In this program. In this life. In this world. Which awaits your step. And your thought. And your decision. Which will change everything. Or will not change anything. But you will still make it. Because you are a computer scientist. And a computer scientist cannot but think. As he cannot but see. As he cannot but live. And this is the most important thing. Not the code. But the life. Which makes you who you are. And who you will become. In the next step. In the next thought. In the next life.

Data That Speak

Information search and analysis is a mechanism that informatics requires and develops. The modern world is a world of data. Informatics is the subject that teaches how to work with this data: to find, structure, analyze, interpret. This mechanism belongs to the supporting level: without it the core will not receive material for work, but information search in itself does not yet create a solution. The ability to formulate information queries. The ability to evaluate sources of information. The ability to analyze data and present it in an intelligible form. Work with databases, search engines, data visualization tools. Analysis of real data, and not only ideal examples from the textbook. Research emphasizes that informatics allows one not only to equip a person with skills, but also to form a more systemic thinking [8]. And systems thinking is impossible without the ability to work with information at all levels - from search to interpretation.

Here it is appropriate to recall Florence Nightingale, who in the 19th century used statistics to show that most soldiers died not from wounds, but from disease. She visualized the data in the form of diagrams - and this changed medicine. She did not simply collect data. She made it speak. And informatics teaches us this. Not simply to collect data. But to understand what it means. And how to show it so that others also understand. And then data becomes not a set of numbers. But a story. Which can be told. And which can be understood. And which can change the world. If you know how to tell it. And if you know how to understand it. And informatics gives you this language. The language

of data. In which one can speak with the world. And be heard. And this is the most important thing. Not data. But the story. That stands behind them. And that can be told. If one learns to. And if one wants to.

A Machine That Learns to Think

Forecasting and strategic thinking is a mechanism that informatics requires and develops. Programming is constant forecasting. Before writing code, one must foresee how it will work. Before choosing an algorithm, one must foresee its efficiency. Before running a program, one must foresee possible errors. The ability to foresee the results of an algorithm's operation. The ability to choose the optimal solution strategy. The ability to evaluate the effectiveness of different approaches. Work with algorithms of different complexity, analysis of their effectiveness, comparison of different strategies for solving one task. Projects requiring planning and forecasting. A 2024 study shows that computational thinking includes a problem-solving process that includes designing solutions that take advantage of computers [14]. And designing a solution is always forecasting and strategic planning. Researchers are also developing models that combine six critical types of thinking: logical thinking, systems thinking, sustainable thinking, strategic thinking [13]. Informatics is an ideal environment for developing all these types of thinking simultaneously.

It is like a chess player who thinks several moves ahead. He does not simply see the current position. He sees future positions. And chooses the one that will lead him to victory. But if he does not see the future, he plays at random. And loses. So too in informatics. A student who does not know how to forecast writes code at random. And often makes mistakes. But a student who knows how to forecast writes code consciously. And often finds the right path. Even if it is not the shortest. But it leads to the goal. And this is the most important thing. Not speed. But direction. Not the answer. But the path. That leads to it.

Structures in the Mind

Spatial imagination and long-term memory are mechanisms that informatics also develops. Informatics develops spatial imagination through work with graphics, 3D modeling, and data visualization. And long-term memory through the necessity of memorizing data structures, algorithms, and syntax. The ability to see data structures and algorithms in space. The ability to retain complex logical constructions in memory. Work with visual programming environments (Scratch, flowcharts), 3D modeling, creation of games and animations. Research shows that programming instruction allows one to develop logical, abstract and computational thinking [8]. Abstract thinking is closely connected with spatial imagination - the ability to operate with images and structures in the mind [1]. It is like an architect who looks at a drawing and sees a building. He does not see lines. He sees space. He sees how people will walk. How the light will fall. How the echo will sound. And he can change the drawing so that the building becomes better. But a student who does not know how to see space sees only lines. And cannot change the drawing. Because he does not understand what stands behind it. And behind it stands the world. Which can be seen. If one learns to look. And if one knows where to look.

A Lesson That Begins with a Task

Now I will turn to specifics. What does an informatics lesson built on the principle of dual purpose look like? I propose several examples - from simple to complex. The traditional approach: explain the syntax of commands, give the assignment to write a program according to a model. The approach that develops thinking: give a real task and propose that students independently develop an algorithm for solving it. An example of a task for the seventh-eighth grade: you need to write a program that sorts a list of numbers in ascending order. But before writing the code, you need to develop an algorithm. Analysis of the task: what does it mean to sort a list? What input data do you have? What should the result be? Division into subtasks: into which simpler steps can this task be divided? Construction of the algorithm: write the algorithm in pseudocode or in the form of a flowchart. Explain why the steps go in precisely this order. Forecasting: what will happen if you swap the steps? What will happen if there are identical numbers in the list? Implementation: now write the program in the chosen programming language. Research confirms: the development of effective algorithms for solving tasks is an important part of the school informatics course [8]. And programming instruction is an approach to the development of logical, abstract and computational thinking in schoolchildren [8]. Which mental mechanisms are developed? Cause-and-effect relationships - through the construction of a logical sequence. Forecasting - through forecasting the consequences of changes. Critical thinking - through analysis and verification of the algorithm [1]. It is like a child assembling a construction set. He does not

simply connect parts. He sees in his mind what should result. He imagines how the model will look. And if something does not fit, he changes the parts. And imagines again. And so on until what he has conceived is achieved. But if he does not know how to imagine, he simply connects the parts at random. And often makes mistakes. And gives up. Because he does not see the result. And the result is not what turns out. It is what you see in your mind. Before it turns out. And informatics teaches this. To see before it turns out. To imagine before you build. And then the program becomes not a set of commands. But a solution. Which you saw. And which you can now show to others. And this is the most important thing. Not the result. But the solution. Which makes you a creator. And not an executor. And this is precisely education. Not a sum of knowledge. But a sum of solutions. Which you have created. And which are now yours. Forever. And in the next task. And in the next algorithm. And in the next life. Which always begins with the first question. That you ask. Now. Here. To the machine. Which does not simply answer. It executes. And you are surprised. And you want to understand. And this is the most important thing. Not the answer. But the surprise. Which makes you a human being. Who knows how to see. And understand. And live. In a world. Which is not code. But which can be understood. If one learns to look. And if one knows where to look. And informatics teaches this. Every day. In every lesson. With every child. With every program. Which is not simply written. It is thought through. And in this is its strength. And in this is its beauty. And in this is its meaning. Which does not end. As informatics does not end. As thought does not end. As life does not end.

Walls in Which Code Lives

The development of mental mechanisms requires not only new teaching methods, but also appropriate infrastructure. For informatics, this means creating spaces where algorithmic and computational thinking can develop through practical, research and creative activity. By creating this infrastructure, we form what Andy Clark and Kim Sterelny call a cognitive niche. Infrastructure is not merely premises, but the material part of the environment that provides scaffolding for the development of mental mechanisms. Without this niche, the mechanisms remain abstract concepts. An IT laboratory: a space with modern computers where students can program, work with data, and create projects. Research shows that the integration of project-oriented learning, problem solving and computational thinking into informatics curricula positively influences students' critical thinking and computational skills [13]. A robotics workshop: a space with robotics construction sets where algorithms become tangible. Research shows that physical computing based on Arduino positively influences students' logical thinking [15]. A media studio: a space for creating digital content - video, podcasts, animations, games. Develops not only technical skills, but also eloquence, creativity, and teamwork [1]. A cyber range: a space for teaching digital security, where students can practice protection against cyberthreats in a safe environment.

The creation of this infrastructure is not an expense, but a strategic investment. Every child who finds through these spaces their point of entry - whether it be programming, robotics or the creation of digital content - receives a powerful impulse for the development of all the other mental mechanisms [1]. It is like a city. In a city there are libraries, theaters, workshops. And every child can find what is closer to him. And having found it, can grow. Not because he is forced to. But because the environment allows it. And this is the most important thing. An environment that allows. And not an environment that demands. Because one can demand much. But to allow - that is an art. And school should be an art. And not a factory.

Roots and Branches

Informatics does not exist in isolation. It is connected with all other subjects - and these connections can be used for the development of mental mechanisms. Informatics and mathematics: algebraic and computational thinking are closely connected. A 2025 study covering six countries showed that algebraic thinking and computational thinking are key competencies in modern education, contributing to the development of problem-solving skills and logical thinking [12]. Informatics and physics: modeling of physical processes, processing of experimental data - direct application of computational thinking [1]. Informatics and biology: bioinformatics, processing of genomic data - an area where informatics becomes an indispensable instrument [1]. Informatics and geography: GIS technologies, processing of spatial data - the use of informatics for understanding the world [6]. Informatics and art: generative art, computer graphics, creation of games - a bridge between technology and creativity [1]. Informatics and logic as a new subject: formal logic, theory of algorithms - the foundation on which all informatics is built [1].

The Universal Language of Thinking

I want to end this article with what is perhaps the most important. Informatics is not merely a school subject. It is the universal language of thinking in the digital age. It teaches us to see the world as a system of algorithms, data structures and information flows. It forms the cognitive apparatus capable of logical, algorithmic, computational and systems thinking. That is precisely why informatics, in my view, is a key subject for the formation of the communicative-cognitive framework of the personality [1]. When we teach a child informatics, we teach him not simply to use a computer or write code. We teach him to think - to break the complex into the simple, to see patterns, to build logical chains, to foresee results. We teach him to understand how machines work. And how his own mind works. Because programming is not only about the computer. It is about how you think. How you solve tasks. How you find errors. How you create the new. And this is the most important thing. Not the code. But the thinking. That stands behind it. And that can be developed. If one learns to. And if one wants to.

That is precisely why the restructuring of informatics teaching is not merely a methodological question. It is a strategic task. As I showed in the article on epistemological recursion [7], without the development of mental mechanisms, without the formation of the communicative-cognitive framework of the personality, humanity will not be able to solve the difficult problems of the 21st century. Informatics is the instrument through which the capacity for algorithmic, computational and systems thinking is formed. And if we want our children to become not merely users, but creators of technologies - we must give them this instrument in full measure. Not for the sake of grades. Not for the sake of exams. For the sake of thinking in the digital age. For the sake of life. For the sake of enabling them to see the world not as a set of separate programs, but as a system of algorithms. And so that they can not merely live in this world, but understand it. And, perhaps, change it. For the better. With the help of informatics. Which does not merely count. It thinks. Like Ada Lovelace, who saw in the machine more than a calculator. Like Turing, who asked the question whether a machine can think. Like a computer scientist who sees the truth where there is as yet only code. And this is the most important thing. Not the code. But the truth. That stands behind it. And that can be seen. If one learns to look. And if one knows where to look. And informatics teaches this. Every day. In every lesson. With every child. With every program. Which is not simply written. It is thought through. And in this is its strength. And in this is its beauty. And in this is its meaning. Which does not end. As informatics does not end. As thought does not end. As life does not end.

In the next article I will turn to the subject that, in my view, is the bridge between informatics and the physical world, between algorithms and real actions - to physical education. I will show how this subject can develop not only physical qualities, but also strategic thinking, non-verbal communication, teamwork and emotional intelligence. This will be the next step. The next door. The next house. And I invite the reader to walk this path together with me. Not as an observer. But as a participant. Because school is not what others do. It is what we do. All together. Teachers. Students. Parents. Society. And if we want to change school, we must begin with ourselves. With every lesson. With every program. With every thought. And then, perhaps, in twenty years we will see a different school. Different children. A different world. A world in which thought is not afraid of itself. A world in which questions are valued no less than answers. A world in which education is not preparation for an exam. But preparation for life. And this will not be a utopia. It will be a reality. Which we have created ourselves. With our own hands. With our own thoughts. With our own lessons. And this is the greatest thing we can do. Not for ourselves. But for them. For the children. For the future. For life. Which continues. In every algorithm. In every program. In every thought. Which seeks. And finds. And seeks again. Because in this is the meaning. Not in answers. But in questions. Not in periods. But in ellipses. Not in the end. But in the continuation. Which is always possible. As long as there is code. As long as there is an algorithm. As long as there is thought. As long as there is life.

References to Primary Sources

1. Kleymenov A. Basic Mental Mechanisms: The Structure of the Communicative-Cognitive Framework of the Personality // Fractal Academy. - 2026. - Issue 1. - Art. 5. https://fractal-academy.com/articles/article-1-5/

2. Kleymenov A. Diagnostics and Development of Mental Mechanisms: From Theory to Pedagogical Practice // Fractal Academy. - 2026. - Issue 1. - Art. 6. https://fractal-academy.com/articles/article-1-6/

3. Kleymenov A. The Historical Necessity of Changes in the Pedagogical Model: From the Industrial Conveyor to Post-Industrial Thinking // Fractal Academy. - 2026. - Issue 1. - Art. 3. https://fractal-academy.com/articles/article-1-3/

4. Kleymenov A. The Economic and Evolutionary Disadvantage of Old Pedagogical Models // Fractal Academy. - 2026. - Issue 1. - Art. 4. https://fractal-academy.com/articles/article-1-4/

5. Kleymenov A. Introduction: How Subjects Become Instruments for the Development of Thinking // Fractal Academy. - 2026. - Issue 4. - Art. 1. https://fractal-academy.com/articles/article-4-1/

6. Kleymenov A. Foreign Language: Communication, Dialogue and Intercultural Understanding // Fractal Academy. - 2026. - Issue 4. - Art. 11. https://fractal-academy.com/articles/article-4-11/

7. Kleymenov A. Epistemological Recursion in Cognitive Science: Mental Mechanisms as an Instrument for Overcoming the "Hard Problem" of Consciousness // Fractal Academy. - 2026. - Issue 3. - Art. 1. https://fractal-academy.com/articles/article-3-1/

8. Kagan E.M. Teaching Programming as an Approach to the Development of Logical, Abstract and Computational Thinking in Schoolchildren // RUDN Journal of Informatization in Education. - 2017.

9. Bers M.U. Computer science education as a humanistic endeavor: a model for designing technology-rich formative experiences // Education and Information Technologies. - 2025.

10. Afifudin M.A., Rusijono R., Susamo L.H., Rokhman A.Y. The Effect of Problem-Based Learning Model Assisted by Scratch in Grade VII Informatics Learning on Critical Thinking Skill // EDU-TIK: Jurnal Pendidikan Teknologi Informasi. - 2025.

11. Jeong E.S., Heo M., Jin Y.H., Kim Y.S. Effect of a Flow Chart Learning on Logical Thinking Ability and Performance Achievement in Middle School Computer Programming Class // The Journal of Korean Association of Computer Education. - 2009. - Vol. 12, No. 6. - P. 11-19.

12. Sarmasági P., Rumbus A., Bilbao J., et al. Bridging Algebraic and Computational Thinking: Impacts on Student Development in K-12 Education // Informatics in Education. - 2025. - Vol. 24, No. 2. - P. 343-376.

13. Wardani F.P., Widiyastuti S., Ferisca H.G. The Role of Informatics Teaching in Developing Critical Thinking and Computational Skills in High Schools // Indonesian Journal of Economic and Management Studies. - 2024. - Vol. 2, No. 4.

14. Grizioti M., Kynigos C. Integrating Computational Thinking and Data Science: The Case of Modding Classification Games // Informatics in Education. - 2024. - Vol. 23, No. 1. - P. 101-124.

15. Kim Y., Hong K. The Effects of Physical Computing Based Software Applications Using Arduino on Logical Thinking of Elementary School Students // The Journal of Korean Association of Computer Education. - 2016.

16. Gardeli A., Vosinakis S. Fostering computational thinking in young students through student generated challenges in tangible mobile augmented reality games // Discover Education. - 2025.

Conflict of Interest
The author declares no conflict of interest related to the publication of this article.

Funding
The research was conducted without external funding.

Data Availability Statement
The data discussed in the article are contained in the text of the publication. Additional materials can be requested from the author.

Author Contributions
Aleksandr Kleymenov - concept, research, text writing, final editing.

Disclaimer
The opinions, points of view, and data contained in Fractal Academy publications belong exclusively to the author(s) and do not necessarily reflect the position of the publisher or the editorial board. Fractal Academy and the editorial board bear no responsibility for any damage arising from the use of ideas, methods, instructions, or materials mentioned in the publications. Responsibility for compliance with copyright, accuracy of citation, and reliability of facts lies with the author(s) of the publication.