Computer tools in education, 2026
№ 1: 115–128
http://cte.eltech.ru
doi:10.32603/2071-2340-2026-1-115-128

DESIGN AND APPLICATION OF WISE TASKS AS A TECHNOLOGY OF PRODUCTIVE LEARNING

Malyutin E. V.[1], ✉ egormalyutin1997@gmail.com

[1]Saint Petersburg State Electrotechnical University "LETI",
Professora Popova ul. 5, building 3, St. Petersburg, 197022, Russian Federation

Abstract

The article examines the pedagogical potential of digital educational resources such as Wise Tasks and their role in implementing productive learning technologies. The relevance of the study stems from the widespread practice of using digital tools primarily for organizing the learning process or knowledge assessment, which often leads to a passive role for students and limits the development of their research and creative activity. The paper analyzes the limitations of such approaches and substantiates the need to create educational environments in which students act not only as consumers of ready-made tasks and solutions, but also as active participants in their formulation and development. As a solution, the use of formalized means for representing subject area problems and digital tools that allow for the automatic verification of solutions to constructive tasks is proposed. This approach is implemented in resources like Wise Tasks, where the learner can experiment with different solution methods, test hypotheses, and formulate new problems. Using the example of developing an educational resource on graph theory, it is shown how such systems can support the "learning by teaching" technology, develop algorithmic and computational thinking, and engage students in the collaborative development of learning materials and software modules. The obtained results demonstrate that the use of such digital environments contributes to a deeper mastery of the discipline's content and the formation of research skills.

Keywords: *productive learning, informatics, mathematics, graph theory, pedagogical technologies, Wise Tasks problems, programming, team work, statement of problems.*

Citation: E. V. Malyutin, "Design and Application of Wise Tasks as a Technology of Productive Learning", *Computer tools in education*, no. 1, pp. 115–128, 2026 (in Russian); doi:10.32603/2071-2340-2026-1-115-128

1. INTRODUCTION

The problem of supporting educational research in the process of teaching mathematics has always been the subject of pedagogical research, for example, [1]. The technique presented in the works of Polya and Szegö [2–5] at the end of the last century is still relevant today. The problem of its mass use is a classroom system, in which it is difficult for a teacher to support an individual search for a solution to a problem, while this is precisely the benefit of organizing educational research. The appearance of a computer in the educational process was supposed to improve the situation with the solution of this problem, however, there was a social effect — the use of a computer for organizing search activities was supplanted by simpler forms of support for educational activities from the point of view of implementation — tests, presentations, etc., focused not on supporting the student's productive activity, but on a more effective design of the existing forms of classroom learning activities: instead of a blackboard — an interactive whiteboard, instead of drawing by the teacher on the blackboard — a presentation, instead of solving problems — tests. There was a substitution of learning goals, instead of supporting independent comprehension of the material through solving problems and justifying their actions, schoolchildren began to look for ready-made material on the network and collect presentations from them, which were announced as project activities. As a result we have a huge amount of school graduates who do not know how to build reasoning, compare and analyze facts, and solve complex tasks that do not have ready-made templates. The listed problems indicate that an urgent pedagogical task is to study the possibilities of digital technologies that will develop the student's thinking apparatus, without neglecting the rich information environment that has developed over the past decade.

This situation makes the study of digital technologies that can develop the student's cognitive apparatus, rather than merely facilitating the transfer of information, particularly relevant. In this context, the development of the modern information environment allows us to take a fresh look at the realization of the fundamental goals of education, primarily the intellectual development of the individual. As C. Hoyles [6] notes, digital technologies can transform mathematical activity when they support not only access to information but also new ways of working with it.

One promising direction for addressing this issue is resources like Wise Tasks — self-verifying constructive tasks based on a formalized description of a subject domain and automatic solution verification, without imposing a ready-made solution method on the student [7, 8]. Such systems create conditions for transitioning from reproductive task completion to research and project activities, and also allow us to reframe the question of involving students not only in solving but also in posing problems. It seems particularly important that such digital environments can serve as the foundation for technologies of productive learning and "learning by teaching" [6, 9–14], where the outcome of the activity is both an educational product and the development of the participants themselves in this activity.

These ideas also align with the socio-cultural tradition, tracing back to L. S. Vygotsky's theory of the zone of proximal development [15], which underpins much research on peer learning and tutoring. When students act not only as recipients of instructions but also as organizers, explainers, and designers of learning situations, their relationship with knowledge changes qualitatively.

In the article, constructive tasks are considered a special class of assignments that retain the exploratory nature of mathematical activity while allowing for formalized digital verification. Instead of choosing an answer from a list, the student constructs an object, proposes a solution, tests hypotheses, and refines the result. Resources like Wise Tasks are presented as an implementation of precisely this approach: they allow for the automatic verification of constructive task solutions without reducing the task to a predetermined human solution method. In such an environment, the student gets the opportunity to experiment with different strategies, explore task variations, and formulate new problems. Thus, digital technology becomes not a replacement for mathematical reasoning, but a means of supporting it.

The collaborative development of educational resources can itself be viewed as a pedagogical technology. The collective creation of a digital resource possesses all the characteristics of productive learning [16, 17]: its outcome is not only a specific educational or software product but also a change in the participants themselves, who master the interdisciplinary connections between mathematics, computer science, and programming, as well as teamwork skills. In the case described in the article, students participate in developing modules for the Wise Tasks system, work in teams, distribute responsibility for algorithms, test each other's work, and then contribute to populating the system with new tasks. This organization of activities not only supports the study of graph theory and programming but also creates conditions for genuine educational research, as students become involved in posing complex constructive problems, varying parameters, and testing hypotheses.

Consequently, the study of digital environments that support constructive tasks and allow the learner to act simultaneously as a teacher and a problem-setter represents a relevant pedagogical challenge. The article under consideration shows that environments like Wise Tasks can serve precisely as such a tool. Their use makes it possible to organize productive learning in mathematics and computer science, support research activities, develop algorithmic and computational thinking, and foster important social competencies. In this sense, the role of digital technology is not simply to enhance the efficiency of traditional educational practice, but to create conditions for a deeper assimilation of content, the intellectual freedom of the learner, and their participation in meaningful collaborative activity.

2. CONSTRUCTIVE TASKS

Problem solving is the most successful method of learning mathematics. The most famous example is the postgraduate book by Polya and Szegö in mathematical analysis [2]. Polya is the author of the most developed methodology for using tasks for organizing student research activities, which characterizes the productivity of an approach to learning. This technique is adhered in the mathematical circles (in the USSR, and now in Russia and other countries, due to emigration in the 90s of the last century, for example, [18].

Authors of problem books in mathematics for school usually distinguish several classes of mathematical problems, for example, construction problems, computation problems, transformation problems, proof problems, etc.

The appearance of computers in the educational process, initiated the transfer of problem books into electronic form. Several different trends have emerged here [19]. Let's highlight the following:

1. replacing classical problems with tasks convenient for verification, mainly with tests of various types;
2. using computer algebra systems to solve or check solutions when performing actions with formula representations of mathematical objects [20–22];
3. instrumental support of the performed actions, such as in dynamic geometry systems [23–25];
4. constructive problems with verification of partial solutions [8, 26–28].

The most common in Learning Support Systems (LMS) are test forms of assignments. This form of assignments can be useful for checking and self-checking simple knowledge associated with memorizing some facts [29]. With the correct design of tests, they can be used to test the understanding of the conceptual ideas of the material. At the same time, such tests are tied to the textual presentation of the material, terminology, designations, and reflect the author's view of the material. The main disadvantage of tests is that they are not (and should not be) a means of teaching, otherwise, instead of teaching the subject, students begin to learn how to respond to tests.

At first glance, the use of computer algebra systems seems to be the most natural way to use a computer to teach mathematics, but in practice, many pitfalls have emerged. Let's list some of them:

1. computer algebra systems are the implementation of algorithms for solving mathematical problems, they are not intended to teach these algorithms hemselves, which are usually invisible to the user (for example, Maple, Maxima, Mathematica);
2. using an interface to translate internal actions on mathematical objects into a form familiar to schoolchildren (for example, Universal Math Solver) is fraught with the fact that the tool turns from a problem book into a solution book, that is, it demonstrates the solution of any problem from a school course (in this case, algebra); it is difficult for a user of such a system to resist the temptation to look at how the system solves his problem; the fact that she will do it quickly and beautifully will be a motive to repeat this decision or look for another only for a minority;
3. the use of computer algebra systems in order to check the correctness of transformations and logical transitions in a freely presented solution (for example, STACK — computer aided assessment package for mathematics, which provides a question type for the Moodle quiz), will probably get rid of the above disadvantages, however In practice, it turned out that the creation of such systems requires large investments and at the moment the development of this direction has slowed down.

Instrumental support of educational activities in the study of mathematics turned out to be most in demand in schools. Dozens of dynamic geometry systems were born at the end of the last century and the beginning of this century [30], millions of teachers became users of such systems.

Let us list those properties of dynamic geometry that have determined the success of instrumental environments in school teaching:

1. working with tools is a fundamental basis for mastering important mathematical concepts [31, 32];
2. the teacher is not limited to the methodology embedded in the computer support tool and can use the instrumental environment so that it supports his established approach to teaching; some teachers use dynamic geometry as a tool for quickly constructing geometric drawings, others use it to introduce new concepts that are materialized by tools of the environment (for example, transformations of plane figures), and still others use it to organize experimental and research activities [25];
3. users of the system can find solutions to problems that are original (not those that the teacher expected), the problem arising in this case — how to convince the student of the correct solution (and the teacher quickly make sure that the student is on the right path) — is solved in a natural way due to visual verification of construction; the latter means that the construction constructed by the student will be correct not only for unique parameter values, but also for any parameters of objects that change automatically when moving, for example, points defining the constructed object.

The last of the paragraphs devoted to the analysis of the role of tools shows the importance of connecting tools with a verification tool for student-created constructions. This approach has been studied in [33, 34]. The development of this method led to the concept of self-checking tasks [35]. Let us present this concept using the example of two works: Wise Tasks Combinatorics, Wise Tasks Geometry [8, 28, 36–38]. These works are united by the possibility of a formalized presentation of mathematical problems, which allows us to consider the text of the problem not as its formulation, but as a formulation function, which translates from a strict language accessible for automatic processing into a verbal form familiar to the student.

In the first system [36], a special xml-language was created for the formal description of combinatorial problems, which allows one to describe the sets on which the problem and the conditions for selecting the required combinations are set. Next, the system calculates the answer based on enumerating the entire set of combinations and calculates those that satisfy the conditions of the problem. The student has an interface that allows you to represent the answer in arithmetic form using all arithmetic operations and combinatorial functions (factorial, number of combinations). The student's answer is presented in long numbers and compared with the one found by the system. Thus, the student's task is not to get the correct numerical answer (which he can theoretically do by writing an exhaustive algorithm similar to the one used in the system), but to find an algebraic representation of this object through the parameters specified by the condition. This allows us to consider the system as an intellectual partner of the student who does not impose his own solution, does not offer ready-made solutions, but allows in the process of finding a solution to test hypotheses — partial solutions and correct their own mistakes.

The question arise whether teachers will be able to create new tasks in the system, if for this it is necessary to master a special language for describing tasks. This problem was solved as follows: on the basis of the basic editor of the problem in the xml language, editors were created for individual classes of combinatorial problems, which allow generating new problems by choosing parameters. As an example, consider the editor of combinatorial problems on playing cards. In it, the creator of the problem can choose the size of the deck (36 or 52 cards), the number of cards that we randomly extract from it and the composition of the cards, in which both specific cards and their signs (suit, color, value) can be indicated.

Another problem can be a too long computation time, however, for educational tasks, you can always limit yourself to a set for which the computer will respond in real time (seconds).

Let's move on to the description of the Wise Tasks Geometry system [38].

Its difference is that the language for description already exists — it is the language of dynamic geometry (GeoGebra was chosen for the description). Over this system, an add-on was made that allows you to specify the relations between objects (belongs, perpendicular, parallel, equal, etc.). The teacher has the opportunity to set any requirements for the objects being created and to define a limited set of tools for solving the problem (note that the limitations of the tools have a mathematical basis and are used as a methodological media in other systems, for example, Euclidea).

The student creates a structure, and the system checks whether all the requirements of the task are met. Thus, in this system, a solution is not imposed on the student and there are no elements that are specially hidden from him (as, for example, answers are hidden in test problems).

What new pedagogical possibilities does the concept of self-checking problems provide?

First of all, it is an opportunity for the student to set problems independently. Do we need such an opportunity? If yes, why is it not introduced into the educational process of teaching mathematics?

The answer to the first question can be found in the works of Polya: the research approach to the problem involves its changes — simplifications, the solution of auxiliary or similar problems. Why aren't schoolchildren taught how to set problems? The statement of problems presupposes their subsequent solution. In group teaching, what should a teacher do if 30 students are presented with 30 different problems? You can, of course, offer students to exchange problems and solve them. But how can a teacher verify that all tasks are correct and meaningful? Thus, the reason is that students are not taught how to compose problems, not because this activity is not important, but because it was not technologically supported and even unrealizable under the existing conditions.

The emergence of digital technologies makes it possible to raise this problem as one of the significant goals of teaching mathematics at school and university. Moreover, the emergence of a computer in the human environment (noosphere) posed the problem of including a computer in productive human activity, in the educational process — in the relationship between teacher, student and the subject being studied. This direction has received the name — the development of computational thinking. Consequently, the role of task-setting skills becomes important not only for the intellectual development of the student, but also for the harmonious interaction of human and artificial intelligences.

3. "GRAPHS" SYSTEM: TECHNICAL ASPECTS

This paper presents process of creation and using of the "Graphs" system, which combines the properties of

1. self-checking tasks (that is, the system is of the "Wise Tasks" type);
2. the possibility of implementing pedagogical technologies based on the concept of learning through teaching and
3. organizing joint activities to create a product — an educational resource (team development of mathematical software products as a pedagogical technology has been practiced by us for more than five years and is presented in detail in the article [39].

Let's move on to an overview of the technical features of the "Graphs" system, after which we will consider its pedagogical capabilities [8].

The system is based on support of constructive problems on graphs. The problem is always posed as follows: "build a graph with given properties" (Fig. 1). This formulation allows you to get various variations of tasks by changing the user interface, for example:

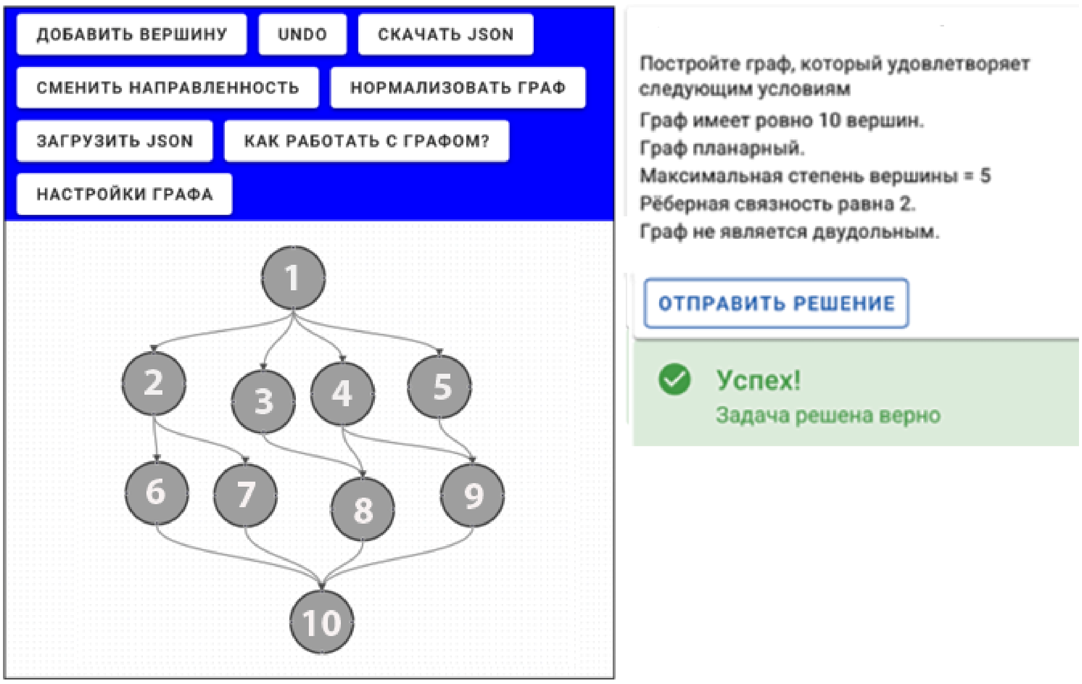

Figure 1. User interface of the "Graph" system with a demonstration of the work of the module that checks the isomorphism of graphs

1. a graph is given, construct a new graph linked by some property to the given one;
2. a graph is given, select its subgraph with the specified properties (for example, build a minimal spanning tree of the graph);
3. a graph is given, mark its vertices, in accordance with some algorithm (for example, the order of transferring edges to the tree of shortest paths of the graph according to Dijkstra's algorithm);
4. given a graph, build a new graph for which this graph will be a subgraph (for example, build a transitive closure graph of a binary relation).

The software foundation of the system consists of modules that represent different properties of graphs as well as algorithms operating on graphs.

As an illustration, several such properties are presented, which simultaneously correspond to the names of the system modules (Table 1). Within the system, three categories of properties are distinguished: binary properties (indicating whether a graph possesses a specified property), numerical characteristics, and properties related to particular subgraphs that are identified according to certain criteria [7].

Table 1.

Binary properties	Numerical characteristics	Marked subgraphs
The constructed graph is a transitive relation graph	Sum of degrees of vertices of an undirected graph	Edges of an undirected graph labeled with natural numbers form an Euler path (traversal in ascending order of labels)
The constructed graph is a Hasse diagram of a strict order relation	The number of vertices of an even degree in an undirected graph	Edges of an undirected graph labeled with natural numbers form an Euler cycle (traversal in ascending order of labels)
The constructed graph is a strict order relation graph	The number of vertices of an odd degree of an undirected graph	Edges of an undirected graph labeled with natural numbers form a Hamiltonian path (traversal in ascending order of labels)
The constructed graph is a graph of a relation of a strict linear order	Diameter of an undirected connected graph	The edges of an undirected graph labeled with natural numbers form a Hamiltonian cycle (traversal in ascending order of labels)
The constructed graph is an equivalence relation graph	Radius of an undirected connected graph	The marked subset of edges (together with their ends) is the spanning tree of an undirected graph
The constructed undirected graph is regular (the degrees of all vertices are equal)	Thickness of an undirected graph (the smallest number of planar subgraphs whose union gives the original graph)	The marked subset of the undirected graph vertices is the smallest dominant
The constructed undirected graph is Hamiltonian	Number of cliques (full subgraphs) in an undirected graph	The marked subset of edges of an undirected graph is the largest matching

4. "GRAPHS" SYSTEM: PEDAGOGICAL ASPECTS

The resource created on the basis of the proposed system "Wise Tasks Graphs" has three different pedagogical functions:

1. the resource was created as an open web resource for expanding the system with new modules, thus, adding new modules is a practice for students and schoolchildren in mastering algorithmic thinking when studying the topic "Graphs" as well as the actual material of the topic itself;
2. the system is being filled with tasks also through the web interface by combining various conditions imposed on the graph, for example, "build a nonplanar graph on 6 vertices that is not a complete graph";
3. the system can be considered as a problem book on graph theory; for this, in addition to adding new tasks, a task classifier is provided in the system.

The work with students developing modules in the described project is based on the approach presented in the work "Computer Algebra System as a Pedagogical Problem" [39]. Students are divided into teams of 1–5 people. A student who knows the language in which the system is written (in this case Java) is chosen as the team leader. Students are offered a choice of about 100 modules (see Table 10). The team takes for execution the number of modules equal to the number of people in the team so that each participant is responsible for the implementation of one algorithm. The team leader is responsible for the correct implementation of the algorithms, organizing testing of modules within the team.

The author of the shell (a 4th year student developing the shell as part of the final qualifying work) conducts weekly consultations with team leaders and those who wish on the Java language and the technical characteristics of the system, while taking into account the emerging problems for the correction and development of the system shell itself. He is also responsible for the final acceptance of the modules developed by the students.

After the end of the stage of creating the system, the stage of filling the system with tasks begins.

Since the system is open, any of its users who are interested in creating a problem book accompanying a course in graph theory can be engaged in filling it.

This opens up new pedagogical opportunities for working with students and schoolchildren in the study of mathematics (in this context of graph theory) through the compilation and solution of problems.

For students, this can be homework, in which it is proposed to compose the most difficult constructive problem on the topic under study and conduct research related to its solution, for example, increasing the dimension of the problem (the number of vertices and / or edges), putting forward and testing hypotheses.

In case of difficulties, the teacher can suggest topics. The possibility of organizing such a study even for unsolved problems is described in the article [40], where the Kelly-Ulam hypothesis of vertex reconstructability of a graph is considered as such a problem.

For school children, the form of "mathematical battle" can be taken as a basis, when teams compose and offer each other sets of problems, the solutions of which they know, and opponents must present such solutions in a fixed time. This way of organizing work with self-checking tasks was tested on the Wise Tasks Combinatorics system.

The topmost "layer" of work with the system will be the regular use of problem books created as part of the use of the resource by teachers. Note that only this "pedagogical layer" of educational activity is considered in the implementation of traditional methods of LMS filling.

Fig. 2 presents for comparison schemes of pedagogical aspects of traditional work with tasks and work using the resource of self-checking tasks and using the resource development process as a pedagogical technology.

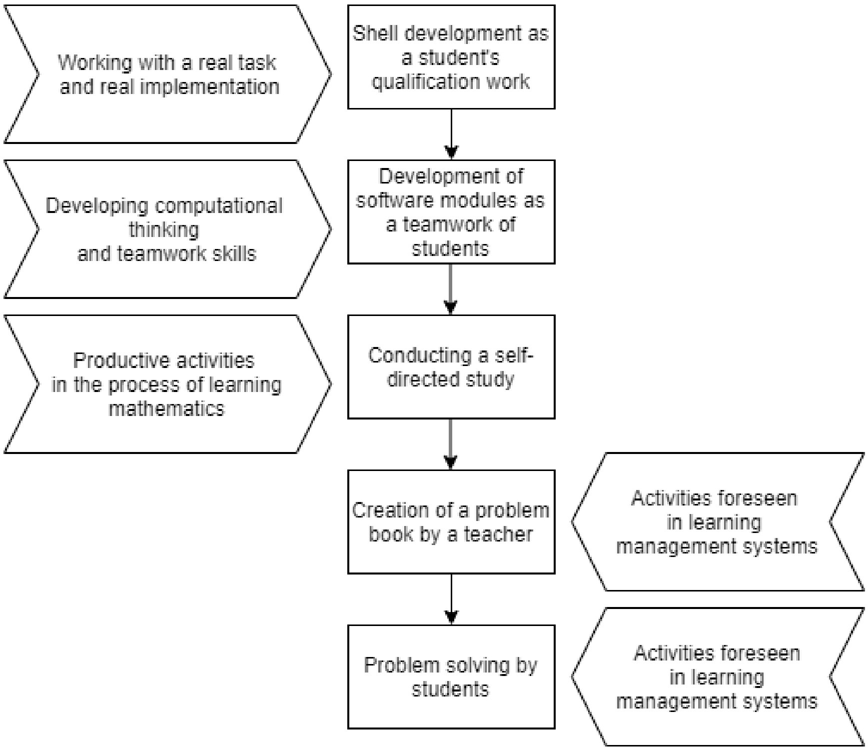

Figure 2. Diagram of the organization of activities to create a resource for self-checking tasks (middle part of the diagram). Left and right — pedagogical aspects of activities related to the Internet resource. On the right are the pedagogical aspects supported by the LMS, on the left are the pedagogical aspects associated with the introduction of new pedagogical technologies

5. DISCUSSION

Let us give an analysis of the proposed pedagogical technologies from two points of view:

1. from the point of view of changing the informational learning environment;
2. from the point of view of the formation of important social competencies (soft skills).

The first point of view is connected to a greater extent with the methodological aspect of teaching.

The problems of creating an information learning environment are discussed in detail in the monograph [41]. It highlights two essential features of the information environment for teaching mathematics:

1. availability of various forms of knowledge representation;
2. the growing role of instrumental knowledge representation.

The analysis carried out in the monograph has been refined in accordance with the changes that have occurred over the past two decades in the article [42].

The article emphasizes that the role of artifacts as tools enabling students to master the mechanisms of their own thinking — an idea clearly articulated in the works of Seymour Papert [43] — has not received the same level of thorough conceptual development in subsequent research. This occurred despite the fact that Constructivism became one of the most influential and widely discussed theories of learning. The authors therefore conclude that, in the preparation of mathematics teachers, greater emphasis should be placed on using information technologies not merely as instructional aids, but as tools that support the development of students' cognitive mechanisms.

Within this context, the proposed approach of collaboratively creating educational resources (artifacts in the sense described above) that possess an instrumental character demonstrates two possible ways of employing information technology as a means for developing students' thinking mechanisms.

1. increased attention to the algorithmic aspects of the school mathematics course, the transition from the study of formulas to the study of algorithms as an extension of the concept of a formula; thereby increasing links between mathematics and programming;
2. expanding the concept of a mathematical problem in such a way that it includes the formulation of the problem and thereby opens the way to a research approach to the study of mathematics; an increase in the number of resources such as Wise Tasks that allow you to combine the intellectual activity of students with intellectual technologies without sacrificing the development of the student's mental mechanisms.

The second point of view is determined by the general pedagogical goals of the learning process.

For the definition of "soft skills" we take the one given in Wikipedia: Soft skills, also referred to as core or transferable skills, are abilities that are valuable across all professions. Examples include critical thinking, problem-solving, public speaking, professional writing, teamwork, digital literacy, leadership, professional attitude, work ethic, career management, and intercultural competence. These differ from hard skills, which are specific to particular occupations or technical tasks. The term "skill" emphasizes a practical capacity to perform tasks. Broadly, it can refer to abilities ranging from simple actions, such as learning to kick a ball, to more complex capabilities, like developing creativity. In the context of professional development, however, "skill" specifically refers to the mastery of deliberate and controlled actions. It is difficult to form soft skills in an environment of lecturer-student interaction, when communication is subordinated to one task — the transfer of knowledge that the lecturer has extracted from some scientific field and transformed into a sequence of educational units available for students. The activity of students in this case consists in the activity of the intellect, which in the process of the lecturer's story must actualize their own knowledge, confront them with those offered by the lecturer, seek and get rid of contradictions, etc.

This is a very rich and little formalized area, which is often submitted to simplified form, being tied to pedagogical theory, reflecting only one aspect of the complex pedagogical process [44].

At the same time, soft skills are of a simpler nature, not related to the depth of the subject being studied, but only to the forms of interaction between students and the manifestation of their activity.

Thus, if the mastery (appropriation, meaning) of knowledge is the field of study of cognitive psychology, then soft skills belong to the field of pedagogy and are associated with the organizational forms of educational work. The use of resources based on self-checking tasks allows the implementation of pedagogical technologies in which soft skills and social skills are manifested to a greater extent. The joint development of a resource (joint both for students and for students with teachers) bears all the features of a productive approach to learning [45–48], when two meanings of the term "product" are considered as a whole: the concepts of a product as a physical the result of the activity and the product, as changes in the person himself, participating in this activity.

6. CONCLUSION

The article presents an analysis of pedagogical technologies related to the support of productive activities in the study of mathematics and computer science. Two pedagogical technologies were proposed and analyzed:

1. this approach involves a technology for organizing collaborative activities aimed at creating mathematical or computer science resources. In this context, the development of a software system is treated as a pedagogical task. Such activity exhibits all the characteristics of industrial intellectual work: the outcome is not only the creation of a software product, but the participants themselves also become a product of the process. Through engagement in this work, they acquire interdisciplinary knowledge bridging mathematics and programming, as well as essential collaborative skills necessary for effective joint activity.
2. technologies for using training tools to support work with a task such as Wise Tasks; tasks of this type are characterized by the fact that they carry out verification of solutions to problems without prior solution of these problems by a person; the use of resources built on the basis of Wise Tasks technology allows organizing support for research activities; the latter is ensured by the fact that the user of the system can set new tasks and solve them, using the system as a specialist correcting search activity.

References

1. V. A. Dalinger, “Educational and research activities of students in the study of mathematics,” *Almanac of Modern Science and Education*, no. 11 (42), part 1, pp. 36–39, 2010 (in Russian).
2. G. Pólya and G. Szegö, *Problems and Theorems in Analysis*, Berlin: Springer Science & Business Media, 1998.
3. G. Pólya, *Mathematics and Plausible Reasoning*, Princeton, NJ, USA: Princeton University Press, 1954.
4. G. Pólya, *Mathematical Discovery: On Understanding, Learning and Teaching Problem Solving*, New York, NY, USA: Wiley, 1981.
5. G. Pólya, *How to Solve It: A New Aspect of Mathematical Method*, Princeton, NJ, USA: Princeton University Press, 2004.
6. C. Hoyles, “Transforming the mathematical practices of learners and teachers through digital technology,” *Research in Mathematics Education*, vol. 20, no. 3, pp. 209–228, 2018; doi:10.1080/14794802.2018.1484799
7. D. G. Zaykov and S. N. Pozdniakov, “Wise Tasks Graphs System,” in *Proc. Int. Conf. Polynomial Computer Algebra*, Saint Petersburg, Russia, 2022, pp. 180–185.
8. E. Malyutin, “Wise Tasks: Algorithms for Self-Checking Problems and Intelligent Tutoring Systems,” in *Polynomial Computer Algebra 2025*, St. Petersburg, Russia: Euler International Mathematical Institute, 2025.
9. L. Fiorella and R. E. Mayer, “The relative benefits of learning by teaching and teaching expectancy,” *Contemporary Educational Psychology*, vol. 38, no. 4, pp. 281–288, 2013; doi:10.1016/j.cedpsych.2013.06.001
10. N. M. Webb, “Peer interaction and learning in small groups,” *International Journal of Educational Research*, vol. 13, pp. 21–39, 1989.
11. S. Stollhans, “Learning by teaching: developing transferable skills,” in *Employability for languages: a handbook*, E. Corradini, K. Borthwick, and A. Gallagher-Brett, Eds. Research-publishing.net, 2016, pp. 161–164; doi:10.14705/rpnet.2016.cbg2016.478

12. D. Duran, “Learning-by-teaching. Evidence and implications as a pedagogical mechanism,” *Innovations in Education and Teaching International*, vol. 54, no. 5, pp. 476–484, 2017; doi:10.1080/14703297.2016.1156011
13. G. Biswas, J. R. Segedy, and K. Bunchongchit, “From Design to Implementation to Practice a Learning by Teaching System: Betty’s Brain,” *Int. J. Artif. Intell. Educ.*, vol. 26, pp. 350–364, 2016; doi:10.1007/s40593-015-0057-9
14. E. Yarzebinski, A. Ogan, M. M. T. Rodrigo, and N. Matsuda, “Understanding Students’ Use of Code-Switching in a Learning by Teaching Technology,” in *Artificial Intelligence in Education*, pp. 504–513, 2015 2015; doi:10.1007/978-3-319-19773-9_50
15. L. S. Vygotsky and A. R. Luria, “Tool and symbol in child development,” in *The Vygotsky Reader*, R. Van der Veer and J. Valsiner, eds. Wiley-Blackwell, Oxford, UK: Blackwell, 1994, pp. 99–174.
16. C.-Y. Chou, T.-W. Chan, and C.-J. Lin, “Redefining the learning companion: the past, present, and future of educational agents,” *Computers and Education*, vol. 40, pp. 255–269, 2003.
17. E. Tolkacheva, S. Ivanov, and S. Pozdniakov, “Implementation of Horizontal Connections in the Course of Mathematics by Combining Pedagogical and Digital Technologies,” *Mathematics*, vol. 10, no. 13, p. 2352, 2022; doi:10.3390/math10132352
18. A. Burago, *Mathematical Circle Diaries, Year 2: Complete Curriculum for Grades 6 to 8*, (MSRI Mathematical Circles Library, vol. 20), Berkeley, CA, USA: MSRI, 2018.
19. V. Freiman, “Types of technology in mathematics education,” in *Encyclopedia of mathematics education*, pp. 869–879, 2020; doi:10.1007/978-3-030-15789-0_158
20. P. Leadbetter and P. Thomas, “A review of computer algebra and its educational implications in the teaching of mathematics,” *Education and Computing*, vol. 5, no. 4, pp. 243–259, 1989.
21. M. Artigue, “Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work,” *International Journal of Computers for Mathematical Learning*, vol. 7, no. 3, pp. 245–274, 2002; doi:10.1023/A:1022103903080
22. A. N. Arkhangelsky, V. N. Dubrovsky, M. Y. Lebedeva, et al., “Personality Extended by Digital Means,” *Herald of the Russian Foundation for Basic Research*, no. 113, pp. 38–51, 2022. (in Russian).
23. I. Lyublinskaya and V. Ryzhik, *Geometry investigations with GeoGebra*, St. Petersburg, Russia: SMIO-Press, 2020 (in Russian).
24. E. V. Shumara, “Computer Dialogue in the Education System,” in *Problems of the theory and practice of teaching mathematics: Proc. Int. Sci. Conf. “The 73rd Herzen Readings”*, St. Petersburg, Russia, 2020, pp. 7–11, (in Russian).
25. V. I. Ryzhik, “Geometry and computer,” *Computer Tools in Education*, no. 6, 2008. (in Russian).
26. A. S. Chukhnov, “Constructive Tasks as a Tool of Invasive and Non-invasive Assessment of Knowledge,” *Computer tools in education*, no. 3, pp. 96–104, 2019; doi:10.32603/2071-2340-2019-3-96-104
27. A. Chukhnov, A. Maytarrattanakhon, I. Posov, and S. Pozdniakov, “Constructive graph tasks in distant contests,” *Informatics in Education*, vol. 19, no. 3, pp. 343–359, 2020; doi:10.15388/infedu.2020.16
28. S. Abramovich, E. Malyutin, and S. Pozdniakov, “Mathematization Through Application and Common Sense: Motivating Intellectual Activities of Schoolchildren with Digital Tools,” *Digital*, vol. 5, no. 3, p. 41, 2025; doi:10.3390/digital5030041
29. E. V. Malyutin, “The use of digital technologies in teaching mathematics: from ‘black box’ systems to intelligent learning environments,” *Development of Education*, vol. 9, no. 1, 2026 (in Russian).
30. U. Kortenkamp et al., “Interoperable interactive geometry for Europe — first technological and educational results and future challenges of the InterGeo project,” in *Proc. 6th Conf. European Research in Mathematics Education (CERME 6)*, Lyon, France, 2009.
31. H. Poincaré, “On the Foundations of Geometry,” *The Monist*, vol. 9, no. 1, pp. 1–43, 1898; doi:10.1093/monist/9.1.1
32. H. Poincaré, *The Foundations of Science*, New York, NY, USA: Science Press, 1902–1908 (reprinted in 1921).
33. S. Pozdnyakov and S. Ivanov, “Computers in productive teaching of mathematics or how information technologies can support intellectual freedom of the learner,” in *The 10th Int. Congr. Math. Education, National Presentation: Russia, Selected Materials*, Copenhagen, Denmark, 2004, pp. 115–124.

34. D. I. Mantserov, "Environment Verifer-KD: Verification of solutions of problems in mathematics," *Computer Tools in Education*, no. 4, pp. 36–41, 2006 (in Russian).
35. *Wise Task Graph*. (2026) Accessed: Jan. 20, 2026. [Online]. Available: https://wisetask.ru/graph/library
36. M. S. Bogdanov, "Automation of checking the solution of problems according to the formal description of its conditions," *Computer Tools in Education*, no. 1, pp. 24–32, 2006 (in Russian).
37. M. Bogdanov, S. Pozdnyakov, and A. Pukhov, "Multiplicity of the knowledge representation forms as a base of using a computer for the studying of the discrete mathematics," *Pedagogika*, vol. 96, pp. 136–142, 2009.
38. O. V. Perchenok, S. N. Pozdnyakov, and I. A. Posov, "Automation of verification of solving geometric problems based on the description of their conditions in a domain-specific language," *Computer Tools in Education*, no. 1, pp. 31–38, 2012 (in Russian).
39. S. N. Pozdniakov, "Computer Algebra System as a Pedagogical Task," *Computer tools in education*, no. 2, pp. 25–41, 2017. (in Russian).
40. S. N. Pozdnyakov, A. G. Lyamov, and N. Y. Prokopenko, "Computer tool in teaching mathematics," *Computer Tools in Education*, no. 1, pp. 27–35, 2006 (in Russian).
41. M. I. Bashmakov, S. N. Pozdnyakov, and N. A. Reznik, *Learning Information Environment*, Saint Petersburg, Russia: Svet, 1997 (in Russian).
42. S. F. Adlaj and S. N. Pozdnyakov, "Digital Representations of Mathematical Objects in the Context of Various Forms of Representation of Mathematical Knowledge," *Computer tools in education*, no. 1, pp. 58–86, 2020 (in Russian); doi:10.32603/2071-2340-2020-1-58
43. S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, New York, NY, USA: Basic Books, Inc., 1980.
44. S. N. Pozdniakov, A. S. Chukhnov, and N. N. Pangina, "Analysis of the Understanding of the Material of Theoretical Informatics in Competitions and Olympiads in Informatics," *Computer tools in education*, no. 2, pp. 55–67, 2018; doi:10.32603/2071-2340-2018-2-55-67
45. M. Wertheimer, *Productive thinking*, New York and London: Harper & brothers, 1945.
46. M. I. Bashmakov, ed., *Theory and practice of productive teaching*, Moscow: Narodnoye obrazovaniye, 2000 (in Russian).
47. I. Böhm and J. Schneider, *Produktives Lernen—eine Bildungschance für Jugendliche in Europa*, Berlin, Germany / Milow, Germany: Schibri-Verlag, 1996.
48. S. Pozdniakov, "Computer in productive learning of mathematics," in *Proc. PME and Yandex Russian Conf.: Technology and Psychology for Mathematics Education*, Moscow: National Research University Higher School of Economics, 2019, pp. 77–92.

Received 27-02-2026, the final version — 12-03-2026.

Egor Malyutin, Assistant Lecturer at Department of Algorithmic Mathematics, ETU "LETI", ✉ egor-malyutin1997@gmail.com

Компьютерные инструменты в образовании, 2026
№ 1: 115–128
УДК: 37.091.3:004.588
http://cte.eltech.ru
doi:10.32603/2071-2340-2026-1-115-128

Проектирование и применение Wise Tasks как технология продуктивного обучения

Малютин Е. В.[1], ✉ egormalyutin1997@gmail.com

[1]Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Российская Федерация,197022, Санкт-Петербург, ул. Профессора Попова, 5, к. 3

Аннотация

В статье рассматриваются педагогические возможности цифровых образовательных ресурсов типа Wise Tasks и их роль в реализации технологий продуктивного обучения. Актуальность исследования связана с распространенной практикой использования цифровых средств преимущественно для организации учебного процесса или контроля знаний, что часто приводит к пассивной роли обучающихся и ограничивает развитие их исследовательской и творческой активности. В работе анализируются ограничения таких подходов и обосновывается необходимость создания образовательных сред, в которых учащиеся выступают не только потребителями готовых заданий и решений, но и активными участниками их постановки и разработки. В качестве решения предлагается использование формализованных средств представления задач предметной области и цифровых инструментов, позволяющих автоматически проверять решения конструктивных задач. Такой подход реализуется в ресурсах типа Wise Tasks, где обучающийся может экспериментировать с различными способами решения, проверять гипотезы и формулировать новые задачи. На примере разработки образовательного ресурса по теории графов показано, как подобные системы могут поддерживать технологию «обучение через обучение», развивать алгоритмическое и вычислительное мышление и вовлекать студентов в совместную разработку учебных материалов и программных модулей. Полученные результаты демонстрируют, что использование подобных цифровых сред способствует более глубокому освоению содержания дисциплины и формированию исследовательских навыков.

Ключевые слова: *продуктивное обучение, информатика, математика, теория графов, педагогические технологии, задачи типа Wise Tasks, программирование, командная работа, постановка задач.*

Цитирование: Малютин Е. В. Проектирование и применение Wise Tasks как технология продуктивного обучения // Компьютерные инструменты в образовании. 2026. № 1. С. 115–128. doi:10.32603/2071-2340-2026-1-115-128

Поступила в редакцию 27.02.2026, окончательный вариант — 12.03.2026.

Малютин Егор Владимирович, ассистент кафедры АМ, СПбГЭТУ «ЛЭТИ»,
✉ **egormalyutin1997@gmail.com**

