Taxonomy of Learning Objectives for the Development of Competencies of Computer Sience Teachers in a Developing Educational Environment*

Evgenia Baranova
ev_baranova@mail.ru

Irina Simonova
ir_1@mail.ru

Herzen State Pedagogical University of Russia,
St. Petersburg, Russian Federation

Abstract

The problem of the development of students' digital competencies in the process of achieving learning objectives of certain classes in the field of algoritmization and programming and methods of teaching computer science in a developing educational environment. The purpose of the study is the system of tasks developed by the authors and based on "revised" B. Bloom's taxonomy, designed to develop algorithmic competence as a basis of digital competencies of students, future computer science teachers.

As part of the study, the problems of modular education, identification and application of Computer Science classes of problems according to "revised" B. Bloom's taxonomy and its teaching method, development and application of electronic learning resources that enrich the educational environment in blended learning conditions were investigated.

The methods of the system analysis and competence approach, the Fisher method were applied to assess statistical validity of the results and to confirm the hypothesis on the efficiency of the use of electronic learning resources (ELR) to develop algorithmic competence of students.

The results of the study: it is proved that the development of students' digital competencies depends on the level of development of algorithmic competence manifested in students' readiness to develop algorithms and programs and to use them in teaching Computer Science in the conditions of digital economy, development of ELR, self-education in Computer Science.

Models of factual, procedural and conceptual knowledge forming the basis of the teaching content of the interrelated modules "Theory and Practice of Algorithmization and Programming " and "Theories and Methods of Computer Science Teaching at School" are described. Classes of problems for the development of algorithmic competence of students according to the "revised" Bloom's taxonomy which describes a hierarchical model of cognitive processes: remember, understand, apply, analyze, evaluate and create are identified and described. The classes of problems correspond to electronic learning resources (ELR) developed by the authors. The efficiency of application of identified classes of problems and ELR for development of algorithmic competence and digital competencies of students has been statistically confirmed.

Keywords: *Taxonomy of learning objectives, digital competencies, algorithmic competence of students, computer science teacher training, blended learning, revised Bloom's taxonomy, electronic learning resources*

1 Introduction

The development of digital competencies of education professionals, including computer science teachers, is key in the digital economy to prepare citizens who are ready to effectively solve problems in their professional activities, social and personal life.

Developed digital competence is characterized by the willingness confidently, creatively, critically, independently (unassisted) use digital technologies and information systems in both private and professional life.

According to the document "The Digital Competence Framework 2.0", developed by the European Commission, the term "digital competence" includes 21 components, combined in 5 groups: Information processing, Communication, Content creation, Safety and protection, Problem solving. The list of components includes such skills as search and selection, data extraction based on specified criteria, data quality assessment, digital content development, effective communication with the use of digital technologies, including in professional activities, formalization and solution of various classes of problems with the use of digital tools, etc.

With regard to the preparation of bachelors of pedagogical education, specializing in the field of information technology, the development of such components is based on algorithmic competence, i.e. readiness for the development of algorithms and programs, their use in professional activities in teaching Computer Science, the development of electronic learning resources, self-education in Computer Science [Baranova, et al., 2020].

The problem of developing algorithmic competence, algorithmic thinking of computer science teachers is investigated both in domestic and foreign sources. The difficulty of mastering the "algorithm" concept due to its abstract, ambiguity of the definition and ways of presenting algorithms is emphasized. Approaches to teachers training methods of developing algorithms, as a necessary stage in the process of creating programs, are considered [Waite et al., 2020].

Algorithmic competence is considered as an important component and means of developing computional thinking, which is one of the key skills of the 21st century [Tikva & Tambouris, 2021]. The term "computional thinking" is used to describe the ability of students to solve problems through digital technologies, the level of development of their algorithmic thinking, readiness to develop program code and characterizes the ability to abstract and analytical thinking, logical conclusions. The need for research in the development of students' "computional thinking" to develop digital competencies is emphasized in [Angeli & Giannakos, 2020].

The course created and implemented by the authors, designed to develop teachers' computational thinking through teaching the development of algorithms and programs using digital resources and specially organized pedagogical support is described in [Kong, et al., 2020]. The analysis of the experimental results showed the effectiveness of this approach for the development of teachers' digital competencies.

The focus of many works is on creating the conditions for effective programming teaching. So, in [Scherer et al., 2020], [Tsai, 2019] there is an approach to learning based on the wide use of visual programming environments as technological tools, and tasks of different levels from simple, reproductive to metacognition instructions, aimed at developing the readiness to create an algorithm for solving a problem and to analyze and evaluate the results.

The structure and content of learning bachelors and masters of pedagogical education in the disciplines of information technology, aimed at developing their professional competencies, have been created and tested by the authors for many years [Baranova Simonova, 2018]. The need to ensure continuity of the content of training, its consistency, interdisciplinarity, first of all, with mathematics, coherence, completeness, modularity, produceability was identified in order to replicate knowledge, for gradual, systemic development of students' readiness for the effective use of information technology tools in professional activities.

Improving the efficiency of learning should be carried out through the selection of content based on specified principles, the use of innovative methods such as e-learning and distance learning, blended learning, "flipped classroom", etc., the widespread use of electronic learning resources [Baranova, et al., 2020]. According to the authors, the key aspect of students' algorithmic competence development is a specially designed system of problems aimed at accumulating students' systemic knowledge of the subject area and developing a willingness to carry out cognitive processes that are adequate to the problems being solved. The study of basic concepts, principles, and algorithms of compilation theory contributes to the formation of students' systematic understanding of the sources of syntax errors and skills to develop syntactically correct programs [Kwon & Cheon, 2019].

The approach to constructing classes of problems developed by the authors is based on B. Bloom's taxonomy of educational objectives [Bloom et al., 1956] and the so-called Bloom's "revised" taxonomy, developed by his followers L. Anderson, D. Krathwohl and others. A two-dimensional structure is proposed [Anderson et al., 2001] to describe the learning objectives in this approach. It is presented in the form of a table which separates the processes of knowledge accumulation and cognitive processes, due to which each category of taxonomy can be more accurately and deeply characterized. The hierarchy of the knowledge category includes: factual knowledge, conceptual knowledge, procedural knowledge and metacognitive knowledge, as the highest level. The model of the development of cognitive processes [Krathwohl, 2002] is a hierarchy of cognitive activities, combined in the following categories: remember, understand, apply, analyze, evaluate and create.

2 Research Methods

The systematic and competence-based approach serves as a methodological framework of the research. The approach to constructing classes of problems is based on B.Bloom's taxonomy of educational objectives and Bloom's "revised" taxonomy, developed by his followers L. Anderson, D. Krathwohl and others. To solve the problem of the research a set of methods based on the analysis of Russian and foreign pedagogical and psychological theory and practice in the area of training future computer science teachers were used. Those were general scientific methods, including modeling, association, comparison and generalization, as well as experimental methods using diagnostic toolkit, expert evaluations and statistical processing of results of pedagogical experiment.

3 Results of Research and Discussion

An approach to constructing classes of problems for teaching students algoritmization and programming on the example of basic modules for preparing an information technology cycle: "Theory and Practice of Algorithmization and Programming" and "Theories and Methods of Computer Science Teaching at School" will be considered

3.1 Module "Theory and Practice of Algorithmization and Programming"

Factual or basic knowledge that students should acquire is knowledge about the system-forming terms of programming languages that form the conceptual core of the subject area. First of all, these are such basic terms as: algorithm, information model, programming paradigm, data type, data structure, identifier, basic control structure, class, object. The second component is detailed, clarifying understanding of the system-forming concepts.

Table 1: Model of presentation of factual knowledge for the module “Theory and Practice of Algorithmization and Programming”

Basic term	**Detailed terms**
Algorithm	algorithm properties, ways of expressing the algorithm
Information model	types of information models, ways of models representation
Programming paradigm	programming language, imperative procedural, functional programming logical programming object-oriented programming
Program	program structure, syntax errors, syntactically correct program, program lifecycle phases, program compilation
Data type	integer, real, boolean, string, pointers, static and dynamic data
Data structure	simple and complex structures (arrays, records, sets, lists, etc.)
Identifier	description and name of the identifier
Basic control structure	assignment operator, conditional operator, loop statement
Subroutine	procedure, function, module, formal and actual parameters
Class	class structure, class description, class as a set of objects
Object	object, as a representative of classes, object field, object method, object property, constructor, destructor, static and virtual methods, scopes
Compilation	program syntax, formal grammars, lexical analysis, syntax analysis, code generation
Database	relation, relation scheme, data type, domain, instance variable, tuple, primary key, foreign key, table, data query language.
Database management system	data storage and retrieval in the database, data processing, data security and integrity

Factual knowledge is formed in students in stages, during the entire period of studying the module. The initial, general understanding of the system-forming concepts is clarified, detailed, enriched by students mastering the purpose, characteristics, structure, specific properties, methods and areas of use of objects representing the concept.

The next level in the hierarchy of knowledge categories, according to [Anderson et al., 2001] is conceptual knowledge. This level involves the formation of students' knowledge of relation between the basic elements of the theory and practice of programming as a subject area. The conceptual level involves knowledge of the categories, principles, theories, models, and approaches to classification inherent in the subject area. With regard to programming, these are: the principles of structured programming, methods of top-down and modular programming, the concept of object-oriented programming, including encapsulation mechanisms, inheritance mechanisms and polymorphism, connections and relations and relations between the basic constructions of object-oriented programming languages, compilation principles, methods of lexical and syntactic analysis, relational database concept, etc.

The level of procedural knowledge involves mastery of algorithms, methods, techniques of the subject area, an idea of ways to choose a toolkit that is adequate to the problem being solved. With regard to programming, these are: basic search algorithms, sorting and processing information presented in the form of various data structures; methods for developing information models that describe the behavior of real and abstract objects, and modern methods for programs development, including software implementation, debugging, testing, presentation and interpretation of results; methods of database creation, tools for processing data in databases; methods of using computer programs in the educational process; methods of developing syntactically correct programs.

According to [Krathwohl, 2002], the model of the cognitive processes development is a description of the hierarchy of cognitive activities, combined in the following categories. To remember is to be able to retrieve relevant knowledge from long-term memory, recall terms, recognize objects, concepts. To understand is to construct meaning from definitions, illustrate, interpret, summarize statements, classify and match the entities of the subject area, conclude, predict the behavior of objects in the subject area under certain conditions. To apply is to carry out and implement procedures in a given situation.

To analyze is to break material into its constituent parts, determine how the parts relate to one another and to an overall structure, to distinguish components by specific properties, to organize sections of the subject area, to attribute the entities of the subject area.

To evaluate is to make judgements, to check the fulfillment of certain conditions, to test. To create is to put elements together to form a coherent whole or invent a new product, hypothesize, plan, design. According to the authors, such hierarchy of cognitive processes corresponds to the activities of specialists in the field of informatization of education, based on their readiness to develop software applications and other information technology tools. This activity involves invention of a new products based on fundamental, conceptual knowledge, experience in the analysis and evaluation of other people's programs, includes the stages of developing information models, algorithms and data structures that are adequate to the problem being solved, software implementation, analysis, evaluation, interpretation of the obtained results. The model proposed by B. Bloom's followers is the basis of the system of problem classes (Table 2), developed by the authors and used in the process of students training.

Table 2: Model of the system of problems for the module "Theory and Practice of Algorithmization and Programming" in the logic of the hierarchy of knowledge and cognitive processes

<table>
<tr><th>Categories of knowledge and cognitive processes</th><th>Factual knowledge</th><th>Conceptual knowledge</th><th>Procedural Knowledge</th></tr>
<tr><td>Remember</td><td>Class of problems 1</td><td></td><td></td></tr>
<tr><td>Understand</td><td></td><td>Class of problems 2</td><td></td></tr>
<tr><td>Apply</td><td></td><td colspan="2">Class of problems 3</td></tr>
<tr><td>Analyze</td><td></td><td></td><td rowspan="3">Class of problems 4</td></tr>
<tr><td>Evaluate</td><td></td><td></td></tr>
<tr><td>Create</td><td></td><td></td></tr>
</table>

Classes of problems focused on achieving learning objectives in accordance with levels of knowledge and categories of cognitive processes will be characterized. Classes of problems are generally described by the authors in [Baranova et al., 2019].

The problems of the first class are aimed at consolidating factual knowledge about basic control structures and data structures of programming languages, compilation stages, database components, stages of designing databases and information systems. Problems are formulated in the form of questions or tasks, for example, of the following types: "Give examples of control structures that provide repeated performing operations", "What is the structure of control structures such as conditional constructions?", "What are variable descriptions for?", "How are formal and actual parameters related in procedures and functions?", "What is the role of the lexical analyzer in the compilation process?", "Describe program lifecycle phases and their purpose", "Describe the basic concepts of a relational database", etc.

The problems of the second class are aimed at developing students' systematic understanding of programming as a science, readiness to build information models, as a basis for software implementation. By comparing the purpose and capabilities of programming constructs, students learn to select data structures, control structures that are adequate to the problem that needs to be solved. An important type of problems in this class is related to the development of the ability to predict the results of the execution of a specified operator, a group of operators, a procedure, a program as a whole. First of all, this is the ability to perceive (read, see, understand) the "program" not as a text, but as a structure, a hierarchy of interrelated entities, the interaction of which ensures the execution of the program and the receipt of results that are uniquely determined by the input data. Example tasks: "Describe the results of performing a specified procedure for different sets of actual parameters", "Find syntax errors in the program", "Describe the structure and relations between database tables to represent a certain subject area". Another type of problems, focused on the formation of students' conceptual knowledge of programming, involves program development that provide a visual representation of the studied control structures and algorithms, for example, loop structures, search algorithms, sorting, etc.

This class of problems develops students' readiness to perform basic data operations, including creating, changing values, deleting data. Data are represented by different structures: arrays, lists, sets, records, object classes, database tables.

The third-class problems are an extensive list of problems aimed at developing students' algorithmic competence. This competence is considered as systemic, conceptual knowledge of the subject area, procedural knowledge that ensures readiness to develop algorithms and programs, which can be applied in their professional activity. This is primarily the development of computational algorithms and programs based on the known mathematical apparatus: approximate calculations, calculation of series sum, array processing, computations of geometric figures' parameters, etc. When solving such problems students develop readiness to interpret mathematical models, to transform them into information models, to create data structures corresponding to the mathematical objects' nature, to «translate» from mathematical language to Computer Science language. Thus, the interdisciplinary links between informatics and mathematics are built.

A special type of this class of problems involves the development of algorithms and programs that simulate phenomena, processes, and behavior of objects from different fields. Solving such problems aims at verification of achieving a certain level of algorithmic competence at the practice level, students' readiness to solve problems in new situations based on known facts, models, and rules. This type, as stated by the authors in [Baranova et al., 2019], includes problems on visualization of known algorithms operation involving new interface elements, software implementation of simple models describing phenomena from various fields of knowledge, science and technology, development of game situation fragments, etc.

The fourth class of problems is focused on the development of the highest-level learning activities, such as analysis, evaluation, and creation. We should note that they are inherent in the professional activities in the field of algorithmization and programming, which involve analysis of subject domains, creation of a new product and evaluation of product quality as mandatory stages. At the stage of analysis, the focus is on the structuring of subject domains, identification of relationships between the components and attribution of selected entities at a certain level of abstraction in order to create an information model as a basis for software implementation. The class includes objectives on developing algorithms and programs for analyzing data that are presented in different structures, data generation according to a given scheme: algorithms for finding and sorting, forming arrays, lists of a given structure, creating classes of objects with a specified behavior, determining the values that characterize the data in the database table by means of the SQL language, creating linked data sets according to a defined scheme, etc.

The "create" category in the refined taxonomy of learning activities implies the ability to produce a new product[Anderson et al., 2001], [Krathwohl, 2002], to design a new whole from parts on the basis of proposed hypotheses according to the developed action plan. When teaching algorithmization and programming with such a new product, the result of students' educational activities is a computer program. The step-by-step process of creating software, even training ones, is inextricably linked to the evaluation of results: during the debugging process the syntax correctness of the program is checked, the program is tested on different sets of input data for verification of logic. A group external evaluation can be organized during the testing process. For example, all students in the group make a value judgment on the product quality:

- program interface – compliance with ergonomic requirements;
- the efficiency of the chosen algorithm – execution speed, the use of memory for data storage;
- the appropriateness of data structures used for the objective – representation of a set of one-type elements in the form of an array, a set of elements of different types in the form of a record, an unordered set of non-repeating elements in the form of a set, objects of the same structure and with the same behavior in the form of a single class, large datasets in the form of a database table, etc.

The problems of this class are rather complex for undergraduate students and involve the development of modular software (information systems). Such programs simulate the behavior of objects of subject domains from various areas of social life: production, scientific, social, and personal. Diversity of objectives is very important, as this helps students to understand the scope and variety of application of modern IT tools.

For example,

- creation of information systems: ATM, Library, Social Networks, Employment, and Dean's Office, simulating the relevant processes,
- development of computer programs: dynamic interactive models Properties of the Integers, Magic Square, etc.
- development of elements of game programming.

A number of tasks are related to students' future professional teaching activities. They include the development of electronic learning resources on various school subjects, test systems, information systems, and web resources for the organization and management of the educational process, etc.

L. Anderson and D. Krathwohl have introduced a fourth type which is called "metacognitive knowledge", the highest level of the knowledge hierarchy. According to the creators, this category implies knowledge of cognition and self-knowledge that are necessary for carrying out research activities. Such knowledge is primarily acquired through graduate qualification work (GQW). Mostly, by the end of their studies students are ready to formulate their preferences for topics and tools of their work, and while preparing GQW their understanding of research strategies and methodology are developed, as well as individual approaches to cognitive activity are formed.

The authors believe that in this respect the GQW subject, connected with the development of e-learning resources of various types in order to accompany the study of a school subject or information technology discipline in higher education, is very useful. Such training and research activities include the analysis of the subject area, development of an information model for e-learning resources, program execution, development of methodological guidelines for the use of e-learning resources, their testing in the learning process and evaluation based on results of such testing.

The considered approach to learning algorithmization and programming, which is based on a specially designed system of objectives, is implemented with extensive use of electronic learning resources developed personally by both the authors and their students while working on GQW under the author's guidance. The structure of all resources is similar. The content of e-learning resources is determined by specifics of the class of problems. Classes of resources are detailed by the authors in [Baranova et al., 2019]. These resources include:

- systematic description of programming principles and methods, algorithms and data structures that are studied, programming language constructs, basic data access operations;
- interactive demonstration examples: software applications that simulate algorithms, illustrate the purpose and features of the data structures being studied, professionally executed examples of information systems, etc.;
- sets of problems and tasks for students of different complexity aimed at accumulation of factual, conceptual, and procedural knowledge and developing readiness to conduct the identified cognitive activities in accordance with Table 2

The e-learning resources system is presented in the LMS Moodle and provides each student with the opportunity to build a comprehended individual educational route. For teachers, it is an effective instrument for managing both classroom-based and distance-learning educational activities.

Table 3: Model of conceptual knowledge presentation for module “Theories and Methods of Computer Science Teaching at School”

Concepts	Detailed concepts
Methodological system of Computer Science teaching at school	functions of the methodological system of Computer Science teaching at school (epistemological, humanistic, design-oriented, normative, and reflexive) and structure of the methodological teaching system (learning objectives, content selection principles, methods, means, and forms of teaching)
System of concepts constituting the content of Computer Science teaching at school	regulatory documents, the federal list of textbooks, stages of conceptualization, intension and extension of concepts, the generalization and abstraction of concepts, the types of concepts' definitions.
Methods and technologies for teaching Computer Science at school	classification of teaching methods, correlation and interconnection of methods, techniques, and technologies for teaching Computer Science at school, technological development of learning using ICT tools
Methods for evaluating knowledge and skills in Computer Science teaching at school	approaches and methods for evaluating achievement (criterion-oriented, standard-oriented, normative-oriented), criteria for knowledge assessment, structure of basic state examination and unified state examination as final knowledge assessment, knowledge quality assessment criteria (completeness, systematic character, depth, etc.)
Knowledge of the Computer Science teaching organization at school	classroom system, lesson typology, information security, safety
The knowledge that forms the content of the school Computer Science course	teaching units of sections: "information and how to present it", "basics of

Within the framework of theoretical education, the content of the school subject Computer Science is analyzed using a retrospective approach. The stages of establishment of the Computer Science content are considered in connection with the development of the science of information and information processes, computer as a universal tool of information processing, software. The content extracts the following concepts: "information", "algorithm", "model" and classes of problems whose solution requires understanding and application of these concepts. During the classes, students must identify topics in the content that have a stable conceptual core, reveal a set of standard problems and exercises (for example, numerical systems, information coding, basic algorithms, etc.), justify the relevance of these sections for the whole course, develop lesson notes, including technology maps and tasks that help to evaluate the students' knowledge and skills.

The introduction of research and creative methods has a positive impact on the development of students, but this requires considerably more teaching time and time to prepare the teacher for the lesson. They analyze and compare the texts of Computer Science school textbooks of different authors according to different criteria: principles of content selection, visibility and accessibility of educational material, development of universal educational activities, etc.

The algorithmic and programming line is certainly one of the most important for the development of a specific learners' (schoolchildren and students) thinking type, often called algorithmic. Some sources use the term "computer thinking" [link]. One of the main goals of the Computer Science teacher is to create the conditions for the successful acquisition of basic knowledge of this line by schoolchildren. It is the success of solving the problems corresponding to this line that serves as an indicator of professional ability in the field of Computer Science and information technology. The established methodology of teaching the basic concepts of this line has the longest history of development in our country. It corresponds to the primary, secondary and high school stages. According to our experience, the fully implemented methodology, starting with simple algorithm executors and ending with the production programming language at specialized schools, gives positive results in the development of digital competencies of students.

Teaching methods and techniques have a significant impact on the development of learners' digital skills and factual knowledge. Formally, Computer Science training uses active methods in the form of laboratory work and workshops. The analysis of texts of these works shows that reproductive-type problems, which do not contribute to creative development, predominate. Implementation of research and creative methods has a positive impact on students' development, but this takes much more learning time and teacher's time to prepare for a lesson.

Procedural knowledge in the methodology of Computer Science teaching ensures the implementation of the practical-oriented learning principle. This knowledge contributes to mastering of the basic methods of solving educational problems of Computer Science school subject by students, as well as of IT tools necessary for performing their tasks on a computer. Procedural knowledge enables students to master methods and technologies for implementing an individual approach to students when teaching Computer Science, ways of creating educational and methodological materials, including lesson notes and flow charts, sets of differentiated tests and other materials (keeping the electronic journal, updating information on a school website).

Methodology of teaching within the Theories and Methods of Computer Science Teaching at School module, aimed at acquisition of factual, conceptual and procedural knowledge by students, assumes the compliance with stages of development of cognitive processes defined by B. Bloom [Bloom et al., 1956] and developed by his followers [Krathwohl, 2002]: knowledge, comprehension, application, analysis, evaluation, creation. This approach is feasible to use in classrooms or in independent work.

Students need to be taught how to distinguish characteristics of each stage in order to be able to find an individual educational path, taking into account the hierarchy of learning activities, creation of problems' sets of certain cognitive complexity (D. Tollingerova), organization of students' project and research activities, including planning, evaluating, interpreting results of the research.

Proposed in [Krathwohl, 2002], the model is the basis for the system of problems (Table 4) developed by the authors and used in the student training process.

Table 4: Model of the system of problems for the module "Theories and Methods of Computer Science Teaching at School" according to the knowledge hierarchy and learning activities

<table>
<tr><th>Categories of knowledge and cognitive learning activities</th><th>Factual knowledge</th><th>Conceptual knowledge</th><th>Procedural knowledge</th></tr>
<tr><td>Remember</td><td rowspan="3">Class of problems 1</td><td rowspan="3">Class of problems 2</td><td></td></tr>
<tr><td>Understand</td><td rowspan="3">Class of problems 3</td></tr>
<tr><td>Apply</td></tr>
<tr><td>Analyze</td><td rowspan="3"></td><td></td></tr>
<tr><td>Evaluate</td><td></td><td rowspan="2">Class of problems 4</td></tr>
<tr><td>Create</td><td></td></tr>
</table>

We describe the classes of problems focused on the competence development of the future Computer Science teacher in accordance with the knowledge levels and learning activities categories. As noted above, the description of the learning goal includes the verb (action) and the noun (object). The verb describes the action that is related to the cognitive process. The object describes students' knowledge they must acquire or create.

The first class includes problems (tasks) aimed at mastering the content of the school subject Computer Science and the ability to explain the solution of problems using schemes, drawings, computer models, to remember and understand logical relationships of Computer Science key concepts:

acquisition of an understanding of content units, key concepts, teaching units and sample problems of the school subject at all levels of formal education: primary, secondary, high. For example: "List content lines (units) specified in the sample primary and high school curriculum", "Create a logical scheme of key concepts with regard to their continuity in the content units", "Create a graphic illustration of the school Computer Science course development against development stages of Computer Science as a science; use network services to make infographics";

solving problems that reveal the school subject Computer Science content units specified in secondary and high school textbooks and other educational learning materials recommended for teaching at school. For example, "Learn the content of the unit "Executor. Algorithm. Program" specified in the Computer Science textbook for secondary school. Solve the problems given in the text of the textbook and workshop. Compare the problems complexity level with the relevant problems in the training materials for the basic state examination. Describe the stages of solving sample problems on the topic. Compile instructional techniques to help students learn the topic of the lesson.

Class of problems 2 involves problems (tasks) facilitating the formation of knowledge related to basic principles of Computer Science teaching at school, the structure and functions of the methodological educational system: objectives, content, forms, methods, educational resources:

tasks facilitating memorization and understanding of key concepts which develop the structure and functions of the methodological educational system: objectives, content, forms, methods, educational resources. These may be a retelling of a lecture or a textbook fragment proposed, or searching for needed definitions in a printed dictionary and filling them into the table with respect to the given structure, or answering reproductive questions on the lecture material such as "What does the term "category mean?", "What components does the methodological educational system include? What problems do they solve? ", " List the functions of the methodological educational system? ", " At what stages of methodical system of Computer Science teaching development is goal-setting carried out?", " How do goals and objectives of Computer Science teaching correlate? "

tasks facilitating understanding of the key stages continuity and basic principles of Computer Science teaching at school.

For example, "List the 20th century main discoveries in Computer Science and specify which of them were reflected in the school subject content?", "Come up with a rationalization for the place of algorithmization and programming in school Computer Science?", "Express your judgment and exemplify what impact this domain had on the content development of the school subject Computer Science?"

This class of problems is refined and enriched in the course of studying the disciplines of the module and is applicable in the other classes of problems [Kostousov & Simonova, 2019].

Class of problems 3 involves problems (tasks) facilitating the formation of skills related to the analysis of regulations, educational learning materials, lessons conducted by Computer Science teachers as well as developing course schedules and creating notes for Computer Science lessons, including objectives, content, methods and educational resources; explaining problem solutions with diagrams, figures, computer models; For example, "What is the conceptual basis of the FSES?", "What subject area does Computer Science relate to?" "How does the list of competencies reflect it?", "List and justify the selection principles of content for basic and professional levels of Computer Science teaching at the high level of education?", " How do materials of the basic state examination and the unified national exam correlate?", "What are the educational standards developed for teachers to prepare for final exams?", "Develop a course schedule for Computer Science for one secondary school term. Create task sheets to support these schedules and arrange individual educational routes for students", "Write notes for lessons on the topic of Numeric Processing for high school professional level of the school Computer Science course", "Create a lesson scenario on one of the secondary school Computer Science course topics with respect to pedagogical programming techniques and implement it in LMS Moodle (Use the Lecture network service)".

Class of problems 4 involves problems (tasks) facilitating the formation of knowledge and skills related to analysis, adaptation and creation of sets of educational problems for Computer Science lessons with regard to differentiation and individualisation of learning as well as of exams, including tests with the help of IT. For example, "Analyze the main stages and techniques of creating a set of tasks with respect to the given cognitive complexity level (according to D. Tollingerova). Develop a set of tasks with respect to the given cognitive complexity level on the topic of Algorithmization for Year 9 school students", "Develop a set of differentiated tasks for secondary school students to strengthen their knowledge and skills on Algorithmization providing implementing tasks with the help of training executor while facilitating the future transfer of this knowledge to other areas of programming", "Develop test tasks to control the acquisition of educational material on one of the school Computer Science course topics and implement it with the help of IT resources", "An example from Kostousov with a link". This class of problems involves the tasks related to developing scenarios for electronic learning resources and their implementation in support of Computer Science teaching at school. For example, "Develop a scenario for a lesson on one of the secondary school Computer Science course topics with regard to the pedagogical techniques aimed to develop critical thinking and implement it in the interactive e-learning resource (ELR) using one of the available network services such as Learningapps", "Develop a scenario and implement multimedia educational resource of an informational nature to better explain theoretical material at Computer Science lessons", "Develop a scenario for an animation video aimed to explain problem-solving on the topic Executors of Algorithms while playing and implement it in the context of popular cartoon characters".

E-learning resources have been developed to support students' educational activities in the course of studying the module that implement individual educational routes during acquiring factual, conceptual and procedural knowledge in the logic of the hierarchy of cognitive activities [Baranova, et al., 2020]. The resources are available via LMS Moodle. Acquiring the systematized theoretical educational material on topics as well as presentations drawing students' attention to key concepts and topics and methodological materials containing questions and tasks for self-reflection in line with tests contribute to the formation of students' factual knowledge.

Solving problems related to the analysis of scholarly articles considering methods of Computer Science teaching, articles based on concepts of writers of Computer Science school textbooks and video clips with speeches of scientists and practitioners at thematic conferences contributes to the expansion of students' conceptual knowledge.

Procedural knowledge is strengthened by learning selected and systematized examples of Computer Science video lessons in various conditions of equipping with IT by the use of different methods of pedagogical techniques and certain effective methods aimed to teach schoolchildren to solve Computer Science problems.

3.2 Experiment

The experiment designed to assess students' readiness to solve the selected classes of problems aimed at the development of teachers' algorithmic competence, cognitive activities and digital competence has been conducted for 6 years (since 2015). 63 students of Years 2–4, trained in the study field of Teacher Education, specialization Computer Studies and Information Technology in Education, Herzen State Pedagogical University, Saint Petersburg, took part in the experiment.

Table 5: Results of the survey on experts' opinion about the correlation between the ability to solve problems of the selected classes and the development of students' digital competencies

Expert number	Information processing	Communication	Content creation	Safety and protection	Problem solving
1	4	2	3	1	2
2	3	2	3	1	2
3	4	2	3	1	2
4	4	2	4	1	3
5	3	1	2	1	3
6	4	3	2	1	3
7	2	3	2	1	2
8	2	1	3	1	4
9	4	2	2	1	2
10	4	2	2	1	4
Statistical characteristics and their values					
Mj	3.40	2.00	2.60	1.00	2.70
Dj	0.71	0.44	0.49	0.00	0.68
σj	0.84	0.67	0.70	0.00	0.82
Vj	0.25	0.33	0.27	0.00	0.30

Experts (teachers) working with the students participating in the experiment were surveyed in the course of the experiment. The experts were asked to analyze the classes of problems and assess their impact on the development of digital competencies. The table below shows the experts' rates x_{ij} (i from 1 to 10) of the correlation of all 5 groups of competencies that make up the digital competence in accordance with The Digital Competence Framework 2.0 and the selected classes of problems (j from 1 to 5); the experts used the integer point system (from 0 to 4). The columns contain all experts' assessments of the whole group of competencies and the rows have each expert's assessments of each group of competencies.

$M_j = \frac{1}{m} \times \sum_1^m x_{ij}$- inter-rater agreement on the j group of problems; the closer the M_j rate is to 4 (maximum possible point), the higher the experts assess the correlation of the relevant group of competencies with students' skills to solve the problems of the selected classes. The analysis of the data obtained showed a high correlation of students' skills to solve the problems of the selected classes with the development of the information processing competencies ($M_j = 3.4$), a lower correlation with digital content activities ($M_j = 2.6$) , problem-solving ($M_j = 2.7$) and communication competencies ($M_j = 2.0$). Students' skills to solve the problems of the selected classes have an even lower correlation with competencies related to information security ($M_j = 1.0$).

The table 5 demonstrates the values of variance and root-mean-square deviation σ_j.

The inter-rater agreement key indicator was calculated according to the coefficient of variation formula. The criterion was defined providing the closer the V_j is to zero, the higher is the level of the agreement between the experts. It can therefore be concluded that the experts have a complete agreement towards the 4th group of competencies ($V_j = 0$). The assessments of the other groups of competencies mark a range of views (V_j ranges from 0.25 to 0.33) indicating a certain degree of reliability corresponding to the value of the root-mean-square deviation σ_j.

The impact of skills to solve problems of the selected classes on the development of cognitive activities using B. Bloom's specific model was assessed in the experiment.

We note that prior to the experiment, both groups received initial training in Computer Science and Teaching Methods. At the beginning of Year 2, students had to solve Computer Science and Methodology problems of the relevant classes of problems and cognitive activities.

The next stages of experimental training were focused on the formation of skills to solve problems of the selected classes as well as on the assessment of the degree of cognitive activities development using B. Bloom's specific model. The results of tasks performed by the students at the end of Years 3 and 4 are presented in Table 6. Columns with % indicate the percentage of completing tasks of the relevant classes by students in each year.

Table 6: The results of the students' cognitive activities formed as of the end of Years 2, 3 and 4

	Beginning of Year 2		End of Year 2		End of Year 3		End of Year 4	
	Number of people	%	Number of people	%	Number of people	%	Number of people	%
Remember	45	71	52	82	55	87	58	92
Understand	38	63	47	75	49	77	60	95
Apply	35	55	41	65	48	76	57	90
Analyze	30	48	34	54	43	68	52	82
Evaluate	27	43	32	51	39	62	44	70
Create	20	32	26	41	38	60	42	67

The results show that solving problems related to analysis, data evaluation, creation of a software product raises difficulties for students in the initial stage which is also observed during subsequent training. However, mastery of these cognitive activities by students along with the rest demonstrates an upward trend as well.

4 Conclusions

The described approach to training bachelors of pedagogical education through the system of tasks in the logic of hierarchy of cognitive activities allows ensuring continuity in the study of special subjects and is aimed at the formation of digital competencies of future Computer Science teachers willing to effectively use and develop the information education environment.

References

[Angeli & Giannakos, 2020] Angeli Ch., Giannakos M. (2020) Computational thinking education: Issues and challenges. Computers in Human Behavior, Vol. 105, April 2020, 106185.

[Anderson et al., 2001] Anderson, L.W. (Ed.), Krathwohl, D.R. (Ed.), Airasian,P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., Wittrock, M.C. (2001). A taxonomy forlearning, teaching, and assessing: A revision of Bloom's Taxonomy of Educational Objectives (Complete edition). New York: Longman.

[Baranova Simonova, 2018] Baranova E.V., Simonova I.V. (2018). The development of professional competencies in the bachelor-level pedagogical training through digital learning in the field of informatics. Izvestia: Herzen University Journal of Humanities Sciences, No. 190, pp. 116–124 (in Russian).

[Baranova et al., 2019] Baranova E.V., Simonova I.V., Bocharov M. I., Zabolotnaia V. V. (2019). The development of students' algorithmic competence by means of electronic learning resources/Proceedings of the 16th International conference on Cognition and Exploratory Learning in the Digital Age (CELDA 2019), Cagliari, Italy, pp. 77–85, November 7–9, 2019. ISBN: 978-989-8533-93- 7.

[Baranova, et al., 2020] Baranova E.V., Simonova I.V., Pavlova T.B.(2020) Development of teachers digital competence through algoritmization and programming/Ceur Workshop Proceedings. Proceedings of the XV International Conference (NESinMIS-2020), 2020, pp. 40–51.

[Bloom et al., 1956] Bloom B. S.; Engelhart M. D.; Furst E. J.; Hill W. H.; Krathwohl D. R. (1956) Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: David McKay Company

[Kong, et al., 2020] Kong S.-Ch., Ming L., Daner S. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. Computers & Education, Vol. 151, July 2020, 103872

[Kostousov & Simonova, 2019] Kostousov S. A., Simonova I. V.(2019). Visual modeling for exploratory problem solving on computer science lessons/Proceedings of the 16th International conference on Cognition and Exploratory Learning in the Digital Age (CELDA 2019), Cagliari, Italy, November 7–9, 2019, pp. 265–273. ISBN: 978-989-8533-93-7

[Krathwohl, 2002] Krathwohl, D. R. (2002) A Revision of Bloom's Taxonomy. Theory into Practice. Vol. 41, No. 4, Autumn, 2002. Ohio State University.

[Kwon & Cheon, 2019] Kwon K., Cheon J. (2019) Exploring Problem Decomposition and Program Development through Block-Based Programs. International Journal of Computer Science Education in Schools, 3(1), No 1.

[Tikva & Tambouris, 2021] Tikva Ch., Tambouris Ef. Mapping (2021) computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. Computers Education, Vol. 162, March 2021, 104083.

[Scherer et al., 2020] Scherer R, Siddiq F., Sánchez Viveros B. (2020) A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. Computers in Human Behavior, Vol. 109, August 2020, 106349.

[Tsai, 2019] Tsai Ch. (2019) Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. Computers in Human Behavior, Vol. 95, June 2019, pp. 224–232].

[Waite et al., 2020] Waite J., Curzon P., Marsh W., Sentence S. (2020). Difficulties with design: The challenges of teaching design in K-5 programming Computers Education, Vol. 150, June 2020, 103838].