The importance of computational thinking training for primary school teachers

Nataliia V. Morze[1 [0000−0003−3477−9254]], Olha V. Barna[2 [0000−0002−2954−9692]], and Mariia A. Boiko[1 [0000−0003−0293−5670]]

[1] Borys Grinchenko Kyiv University,
18/2 Bulvarno-Kudriavska Str, Kyiv, 04053, Ukraine
{n.morze, m.gladun}@kubg.edu.ua
https://eportfolio.kubg.edu.ua/teacher/152
https://eportfolio.kubg.edu.ua/teacher/2466

[2] Ternopil Volodymyr Hnatiuk National Pedagogical University,
2 Maxyma Kryvonosa Str, Ternopil, 46027, Ukraine
barna_ov@fizmat.tnpu.edu.ua
https://tnpu.edu.ua/faculty/fizmat/barna-o-v-.php

Abstract. The importance of modern schools in developing students' problem-solving skills, including through digital tools, is described in the article, which includes the development of basic coding skills and digital literacy, as well as the ability to solve problems and make decisions based on planning and analysis of situations. Computational thinking (CT) is built on the foundation of these abilities. The authors contend that the employment of specialized digital tools promotes the development of computational thinking and that purposeful creation of computational thinking improves teachers' and students' digital competence. The concept of CT, as well as existing definitions and components, are examined in this article. A list of courses from various countries' curriculum on which CT is studied in primary school is provided. The importance of CT as a fundamental talent for everyone is underlined, and it should be developed through the integration of several disciplines to solve problems. The link between the CT components and the key competencies that should be instilled in kids as part of the primary education curriculum. The findings of a survey of more than 60 primary school teachers from various regions of Ukraine about their understanding of the concept of computational thinking and their experience in the development of skills related to all components of computational thinking in primary school students are presented. A review of open educational resources from around the world is offered to assist teachers in enhancing students' CT skills. The strategy for putting the CT development concept into practice for future teachers and primary school teachers is offered, taking into account the ability to employ specific digital tools.

Keywords: computational thinking · decomposition · abstraction · patterns · algorithms · primary school · standard of primary education · digital educational resources · digital competence of primary school teachers

1 Introduction

Today, during the crisis caused by COVID-19, the low level of digital skills of citizens and the lack of wide access to the free use of digital technologies further exacerbate inequality in society. That is why the Digital Education Action Plan (2021–2027) envisages strengthening cooperation at European level: learning from the COVID-19 crisis, when technologies are used on a scale not previously seen in education and training, and creating a digital age-appropriate education system. The new Action Plan has two strategic priorities: helping to develop a highly effective digital education ecosystem and enhancing digital skills and competences in the field of digital transformation [19]. At the same time, according to the [2] in the list of top-10 skills is dominated by problem-solving skills: Analytical thinking and innovation (1), Complex problem solving (3), Critical thinking and analysis (4), Creativity, originality and initiative (5), Reasoning, Problem Solving and Ideation (10). Therefore, it is important to develop basic coding and digital literacy skills, the ability to solve problems and make decisions based on planning and analysis of situations, which is the basis of computational thinking (CT). It is important to develop such skills from an early school age. But such activities can be carried out by teachers who have the most developed ability to perform operations that make up the structure of CT. Therefore, the purpose of this article is to substantiate the need to teach CT to primary school teachers.

2 Theoretical foundations of the study

CT is a component of human thinking, which provides its activities in solving problems of everyday life, and its importance is constantly growing. Various definitions were used to explain the concept of CT. Today there is no single interpretation of this concept. In Ukraine, the English word Computational, in addition to the use in the mathematical sense (production of calculations), is now used in parallel in a broader sense, related to the term ”Computing” – a collective designation of computer science, information technology and information systems, computer and software engineering [18]. The term “Computational Thinking” is tied to the English-language basis and has been used several times in the Ukrainian scientific literature [5, 14, 15]. The widespread use of the term *Computational Thinking* began with the publication in 2006 of the work of the same name by Jeanette M. Wing, who described the definition of CT as follows: computers. “Computers are dull and boring; humans are clever and imaginative. We humans make computers exciting. Equipped with computing devices, we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations” [20]. In particular, Jeannette Wing formulated the following definition: “is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” [23]. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) define computational thinking Computational thinking (CT) as a problem-solving process that includes (but is not limited to) the following characteristics [10]:

– Formulating problems in a way that enables us to use a computer and other tools to help solve them.
– Logically organizing and analyzing data
– Representing data through abstractions such as models and simulations
– Automating solutions through algorithmic thinking (a series of ordered steps)
– Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources
– Generalizing and transferring this problem solving process to a wide variety of problems

In our study, we draw on the approach of Andrew Csizmadia, Paul Curzon, Mark Dorling, Simon Humphreys, Thomas Ng, Cynthia Selby and John Woollard, who believe that CT is a mental activity aimed at solving problems, better understanding of situations, expression of qualities through the systematic use of abstraction, decomposition, creation of algorithms, generalization and evaluation in the process of producing automated solutions that can be implemented using a digital or human computer (computing) device [6]. Components of CT are decomposition, pattern detection, generalization and abstraction and development of algorithms (table 1) [16].

In order to develop students' ability to think using CT, many countries have introduced or plan to introduce a special subject of CT into primary and secondary education programs, including programming subjects as compulsory or optional, of which CT is an integral part [7]. ICT and programming are part of educational programs in the UK, New Zealand, South Korea, USA, Estonia, Cyprus, Australia, Poland, either as a compulsory or optional subject [17]. In Greece, a one-year experiment was conducted in which students learn programming by developing games [8]. Spanish scholars describe the experience of integrating CT in two sections of a Spanish high school course. Students work in small groups and encode three small and one three-dimensional digital history of Spanish culture in Scratch. The results showed that students who took a lesson with an integrated computer theme had the same degree of improvement in their knowledge of Spanish culture as their peers who did not take lessons in integrated CT and significantly improved their knowledge of CT [23]. Some Danish primary schools are participating in pilot studies where students in grades 1–9 work with Scratch and Lego MindStorms in STEM subjects (science, technology, engineering and mathematics) [4]. The Netherlands and Japan have also introduced programming as part of primary or secondary education [9, 21]. The National Research Council (NRC) [1] emphasized the importance of familiarizing students with the concepts of CT in the early school years and helped them understand when and how to apply these basic skills. The Association of Computer Science Teachers (CSTA) and the International Society for Technology in Education (ISTE) presented the basics of CT for K-12 schools in 2011 with

Table 1. Components of computational thinking.

Abstraction	Abstraction is the process of making an artefact more understandable through reducing the unnecessary detail. The skill in abstraction is in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important [6, p. 7].
Algorithmic thinking	Algorithmic thinking is a way of getting to a solution through a clear definition of the steps [6, p. 7].
Automation	Automation is a labour saving process in which a computer is instructed to execute a set of repetitive tasks quickly and efficiently compared to the processing power of a human [12, p. 33].
Decomposition	Decomposition is a way of thinking about artefacts in terms of their component parts. The parts can then be understood, solved, developed and evaluated separately. This makes complex problems easier to solve, novel situations better understood and large systems easier to design [6, p. 8].
Debugging	Debugging is the systematic application of analysis and evaluation using skills such as testing, tracing, and logical thinking to predict and verify outcomes [6, p. 9].
Generalization	Generalization is associated with identifying patterns, similarities and connections, and exploiting those features. It is a way of quickly solving new problems based on previous solutions to problems, and building on prior experience. Algorithms that solve some specific problems can be adapted to solve a whole class of similar problems [6, p. 8].

basic concepts and possibilities of CT, including data collection, data analysis, data presentation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization and modeling [3]. "Thinking by computational method" is a fundamental skill of everyone, not just computer scientists. It can be used to support problem-solving in all disciplines, including the humanities, mathematics and science. This indicates the importance of integrating computational ideas into other disciplines. It should be noted that there are several types of interdisciplinary approach, depending on the links between disciplines: interdisciplinary, multidisciplinary, crossdisciplinary, transdisciplinary [13]. In our opinion, teaching CT in primary school corresponds to a transdisciplinary approach, as it forms the ability to: solve problems and design systems referring to the basic concepts of computer science; create and use different levels of abstraction to better understand and solve problems; think with algorithms and with the ability to apply mathematical concepts for more effective development; understand the implications of scale not only for reasons of efficiency but also for economic and social issues [22]. Transdisciplinary goes beyond individual disciplines, focuses on a specific problem and acquires relevant knowledge that is related to all disciplines, between them and beyond, in order to understand the modern world under the imperative of unity of knowledge (figure 1).

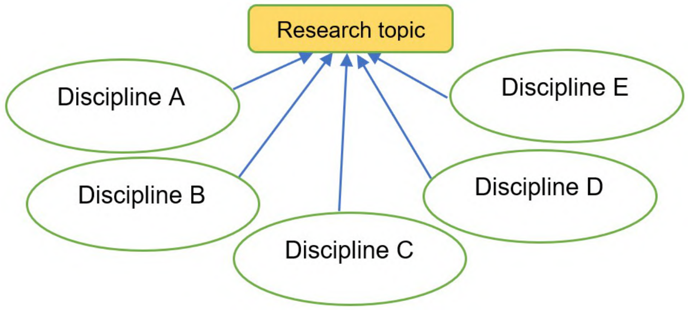

Fig. 1. Transdisciplinary approach.

3 Experimental study

In Ukraine, CT as a separate subject is not studied in primary school, and the development of relevant skills is built through an interdisciplinary approach and integration. According to the Concept of the New Ukrainian School, which has been implemented since 2017, the Standard of Primary Education provides for the formation of 10 equally important and interrelated key competencies that children acquire when studying different subjects at all stages of education. Their combination forms the elements of CT (table 2).

Among the cross-cutting skills that are declared in NUS and those that are implemented through CT, are the skills: critical thinking; ability to logically justify the position; show initiative; ability to solve problems, assess risks and make decisions. The computer science program for primary school clearly identifies topics that cover the described component of digital competence – CT: Teams and performers (2nd grade), Algorithms and performers (3rd grade), Algorithms with branching and repetition (4th grade) (table 3).

However, teachers try to focus on the formation of algorithmic thinking, which is only one component of computing, which leads to the need to focus on other elements. Specially created tasks with the use of electronic educational resources will allow to intuitively involve students not only in the development of algorithms, but also in the processes of decomposition, pattern detection, generalization and abstraction. In addition to these topics, CT can be formed in the study of other topics, with the following requirements for student achievement: chooses and uses the necessary tools of the graphic editor environment to create an image based on a sample and your own design; complements the missing data in simple diagrams, charts; seeks information in texts with false statements and proves the truth; is able to find the necessary information in the text and highlight part of the text; distinguishes and gives examples of devices for input and output of information (3–5) and more. Among the general results of primary school education in language, literature, mathematics and natural sciences can be distinguished components of CT (table 4).

Table 2. The relationship of components of CT with the key competencies of NUS.

NUS key competence	Content	Components
Communication using the state (and native in case of difference) languages	Ability to express and interpret concepts, thoughts, feelings, facts and views orally and in writing	Abstraction, Decomposition, Generalization
Communication using foreign languages	Ability to properly understand a foreign language ...	Abstraction, Generalization
Mathematical competence	Culture of logical and algorithmic thinking. Ability to apply mathematical (numerical and geometric) methods to solve applied problems in various fields. Ability to understand and use simple mathematical models. Ability to build such models to solve problems	Abstraction, Algorithmic thinking, Automation, Decomposition, Debugging, Generalization
Basic competencies in natural sciences and technologies	... Ability to apply the scientific method, observe, analyze, formulate hypotheses, collect data, conduct experiments, analyze results	Debugging, Algorithmic thinking, Automation, Decomposition, Generalization
Ability to learn throughout life	... effective management of resources and information flows, the ability to define learning goals and ways to achieve them ...	Abstraction, Automation, Decomposition, Debugging, Generalization
Initiative and entrepreneurship	Ability to generate new ideas and initiatives and implement them	Algorithmic thinking, Automation, Decomposition, Debugging
Social and civic competence	... Ability to work with others for results, to prevent and resolve conflicts, to reach compromises ...	Decomposition, Debugging
Awareness and self-expression in the culture	The ability to understand works of art, to form own artistic tastes, to express ideas, experiences and feelings through art ...	Abstraction, Decomposition, Generalization
Environmental literacy and healthy living	Ability to use natural resources wisely and rationally	Algorithmic thinking

Table 3. Description of the components of CT, skills, abilities and requirements for the student.

Components	Skills and abilities of the student	State requirements for student achievement
Abstraction Automation Decomposition	can formulate problems in such a way that it becomes possible to solve the problem with the help of a computer or other tools; analyze possible solutions	has an idea of the team, performers; sequence of actions; algorithms and performers of algorithms; composes and executes algorithms for performers to a certain situation in programming environments for children; understands the recording of algorithms in the form of blocks; determines the correct order of commands to the performer in a familiar algorithm; distinguishes false phrases; names opposing statements; is able to look for errors in the sequence of commands; combines items into a group on certain grounds, comes up with a name for the group; removes extra items from the group by attributes, recognizes items by these attributes and selects from the group
Generalization	can systematically collect data through experiments, interviews, surveys or literature studies	
Abstraction Decomposition	can evaluate the found information; understand and compare the found data	
Abstraction Decomposition	can display information in words, pictures, graphs, tables; choose the most efficient data representation	
Decomposition	can divide tasks into smaller tasks, a long list of tasks in subcategories	
Abstraction	can reduce complexity; compare two different concepts and connect them	
Algorithmic thinking Automation Generalization	can generate solutions using algorithmic considerations; automate decisions using algorithmic thinking; write a computer program; generalize the problem-solving process so that it can be applied to other problems	
Automation	can solve a problem or achieve a certain goal by compiling a series of algorithms; use computer resources to obtain a final solution	
Abstraction Decomposition Generalization	can make instructions, simulate the process of solving a problem or perform an experiment based on a specific model; summarize conclusions on problem solving and apply solutions to other problems	
Decomposition Generalization	can create a plan / schedule and assign tasks to team members during the project; allocate resources so that it is possible to use them simultaneously to achieve a common goal	

Table 4. Description of the components of CT in different fields of education.

Educational branch	General learning outcomes of the students	Components
linguistic and literacy	Highlights information	Abstraction
	Analyzes and interprets information and text	Decomposition
	Converts information	Algorithmic thinking
mathematical	Recognizes everyday situations that are solved by mathematical methods; evaluates the data of the problem situation, necessary and sufficient for its solution; analyzes the objects of the surrounding world and situations that arise in life	Abstraction
	Researches, analyzes, evaluates data and relationships between them to solve the problem of mathematical content	Debugging
	Predicts the outcome of a problem	Generalization
	Perceives and transforms information (heard, seen, read), builds an auxiliary model of the problem situation; develops strategies for solving problem situations; evaluates different ways to solve a problem situation, chooses a rational way to solve it; has computing skills, applies them in educational and practical situations	Algorithmic thinking
	Simulates the process of solving a problem situation and implements it	Generalization Algorithmic thinking
	Recognizes geometric shapes by their essential features; uses algebraic concepts to solve a problem situation; explores problems	Decomposition
natural	Identifies and formulates research problems; defines the purpose of the study, puts forward a hypothesis; groups and classifies objects	Decomposition
	Plans research	Algorithmic thinking
	Analyzes and substantiates research results, formulates conclusions	Generalization
	Converts information from one form to another; identifies relationships in nature and takes them into account in its activities; identifies the problem by correlating new facts with previous experience; critically evaluates the problem	Abstraction
	Generates new ideas to solve the problem	Debugging

During the research, we interviewed more than 60 respondents who participated in the education of schoolchildren in NUS programs. Among the respondents, 77% were computer science teachers at NUS, more than 21% were primary school teachers (figure 2). The survey was conducted in all regions of Ukraine through a survey form, which is posted in groups of primary school teachers on social networks.

The sample covered the vast majority of teachers with more than 10 years of experience, which indicates that teachers have sufficient experience of practical work in school (figure 2).

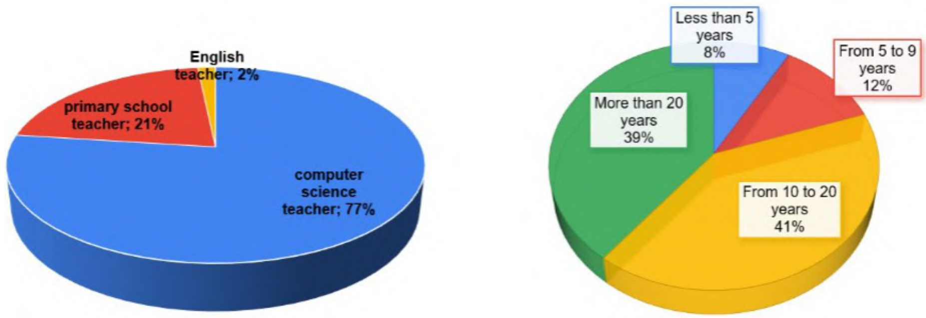

Fig. 2. Contingent of respondents and Work experience of respondents.

Most respondents have a false impression of the essence of the concept of "computational thinking", namely as a method of forming students' computing skills (figure 3).

More than 88% of respondents said that they offer students tasks to identify and formulate a real problem. At the same time, most often it happens in computer science lessons, almost half less – in mathematics lessons, and even less – in the integrated lesson "I explore the world", which combines natural, social, civic, health education (figure 4).

Almost 41% of teachers indicate that they offer students problem questions every day; most often respondents say that their students perform their own experiments several times a week, in 30% of respondents students never or very rarely work with data in different experiments, in 41% of cases students once a month, or even less often record the process and result of the experiment, 50% offer tasks for the formation of assumptions in solving a specific problem (figure 5).

When assessing their own level of CT, the vast majority of respondents rated them at a high or close to it level (figure 6).

The most difficult for teachers is the process of teaching students to divide the task into components (decomposition), plan their activities during experiments (algorithmic thinking), ask problem questions (abstraction, decomposition), use elements of modeling in solving problems (abstraction, generalization), formu-

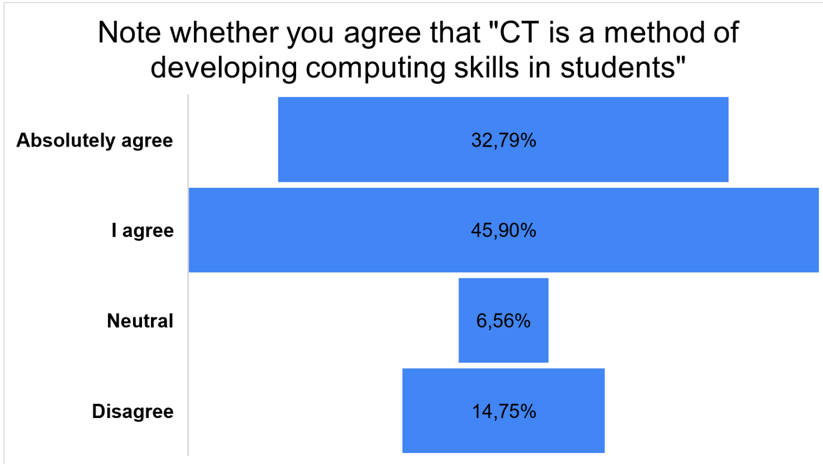

Fig. 3. The choice of respondents to determine the CT.

late assumptions when solving a specific problem (Debugging). The results of the teacher survey are extrapolated to the formation of students' end-to-end skills, some of which are components of CT. The evaluation took place in 2019 as part of a comprehensive study of the effectiveness of NUS implementation (link). According to the results of the study of skills of primary school students, the lowest results were found in the development of critical and systematic thinking skills – 42% of students have low and medium levels, only 7% – high level (figure 7).

Students' ability to think critically and systematically was assessed by performing two competency tasks. The first of them involved the division of statements into true, false and doubtful, the second – a systematic presentation of reliable information and data. The ability to think critically and systematically is developed in students the worst: 7% of groups of students found all the false statements in the task and the ability to structure information and its systematic presentation; 51% of groups of students coped with the tasks quite successfully, at the same time made several mistakes; 10% of student groups did not show signs of critical and systematic thinking; The ability of students to solve problems was determined on the basis of observations of the problem in terms of content and form of presentation of the results of their work and the effectiveness of solving two problems. We analyzed the syllabi of academic disciplines and educational programs in the specialty 013 "Primary Education" of five universities that are in the top 10 (link) among pedagogical universities of Ukraine on the corresponding sites (https://npu.edu.ua, http://tnpu.edu.ua, http://pdpu.edu.ua/b, http://uipa.edu.ua/ua/). None of the programs involves the study of CT as a separate subject. In the syllabuses of courses in academic methods of teaching mathematics, computer science and other disciplines in primary school there are no topics that directly address the issues of CT or its components. Fragmentarily, the syllabi of the courses provide for the formation of future teachers' skills

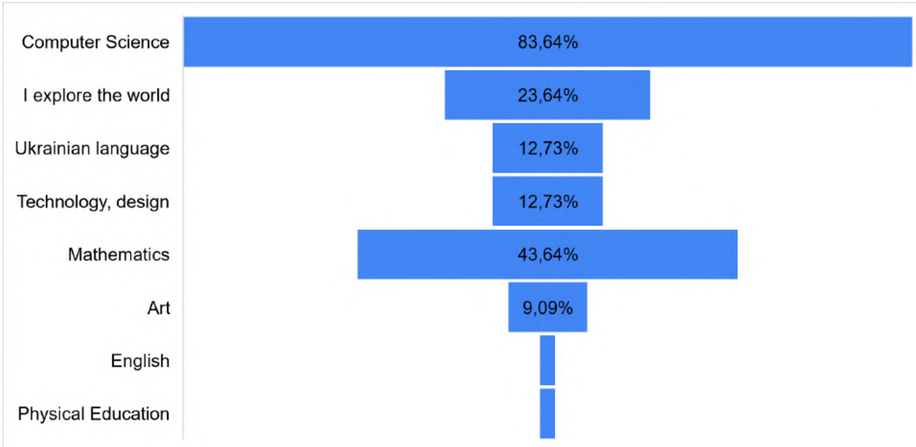

Fig. 4. The choice of respondents to solve real problems with the students during the lessons.

in the formation of compulsory learning outcomes, in accordance with the CURRENT Standard of Primary Education and the Concept of the New Ukrainian School. Thus, the formal component of teacher training for the formation of CT in students is incomplete. A Google search query for the phrase “computational thinking + training” provided access to information on 9 trainings conducted as part of the project of the EU program Erasmus + №586098-EPP-1-2017-1-UA-EPA “Modernization of Pedagogical Higher Education by Innovative Teaching Instruments” (MoPED), International Seminar ”Introduction to Computational Thinking”, organized by the Institute for Digitalisation of Education of the National Academy of Educational Sciences of Ukraine and several local events for educators. This indicates that in the segment of non-formal education, measures to train teachers in CT cover a critically small circle of educators. The fact that teachers need such training and tools for its formation of the components of CT in the classroom is also evidenced by the results of our survey (figure 8).

There are no professional communities and specialized sites on CT in the Ukrainian segment of the Internet. For example, as in the UK (https://community.computingatschool.org.uk/door), the Computational Thinking initiative from Wolfram Research (https://www.computationalthinking.org), the virtual educational environment (Computational thinking in education), community of researchers (https://digitalpromise.org). In this regard, there is a need to create a platform for educators to support informal education on CT.

Components of such a platform can be, for example, a digital library:

- Sites with information about CT
  - Computational thinking https://www.computationalthinking.org
  - K–12 Computer Science Framework https://k12cs.org/computational-thinking
  - Computational thinking https://dystosvita.gnomio.com

https://doi.org/10.31812/educdim.4466

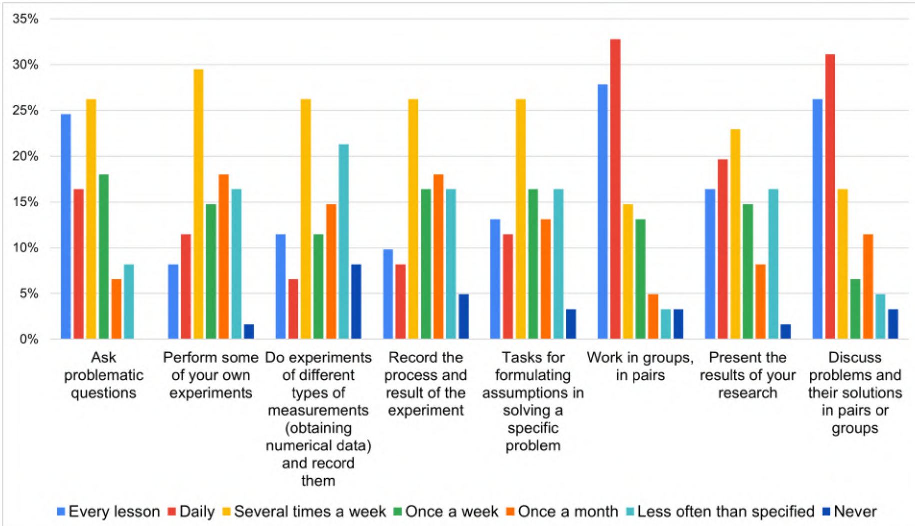

Fig. 5. Respondents' choice of lessons activities.

  - ISTE Computational thinking in education https://www.iste.org/areas-of-focus/computational-thinking
- Resources for teachers with ready-made developments
  - Google for Education. Exploring Computational Thinking resources https://edu.google.com/resources/programs/exploring-computational-thinking
  - Computing at School https://community.computingatschool.org.uk/resources/landing
  - Barefoot Computing primary classroom resources https://www.bare-footcomputing.org/primary-computing-resources
  - Computer Science without a computer https://csunplugged.org/en
  - Programamos https://programamos.es
  - Raspberry Pi https://projects.raspberrypi.org/en
- E-resources with exercises
  - Blockly https://blockly.games
  - Compus https://compus.deusto.es
  - Code https://code.org
  - Bebras https://www.bebras.org
  - Coder Dojo https://coderdojo.com/resources
  - Code Club https://www.codeclubworld.org/projects
- Resources for creating your own exercises with CT
  - Kodetu http://kodetu.org
  - MakeWord https://makeworld.eu
  - LearningApps https://learningapps.org
  - Puzzle designer http://pazlyonline.com/konstruktor.html, https://www.jigsawplanet.com
  - Rebus designer http://rebus1.com/ua/index.php

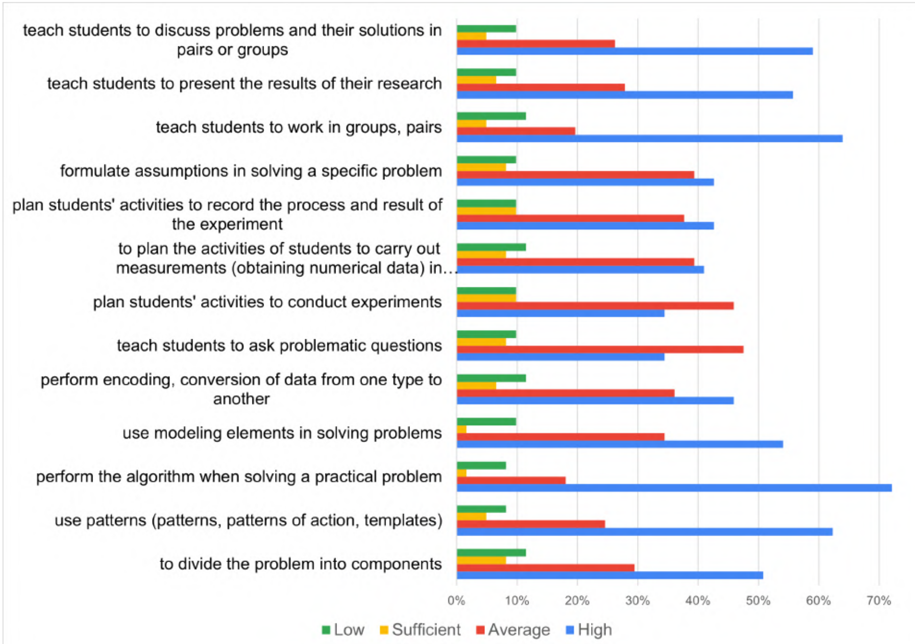

Fig. 6. Respondents' choice of skills they possess.

  - Tinkercad https://www.tinkercad.com
  - Studystack https://www.studystack.com
- Integrated coding environments
  - Scratch https://scratch.mit.edu
  - Alice https://www.alice.org
  - Greenfoot https://www.greenfoot.org/door
  - Agentsheets https://agentsheets.com
- Robotics and circuitry
  - Lego WeDo, Mindstorms
  - Arduino LilyPad
  - BBC micro:bit
  - Bee-Bot
  - Makeblock
  - Makeymakey

In the Ukrainian educational space, not only such open educational resources should be created to help teachers, but first of all it is expedient to develop a concept of teaching teachers and future teachers CT and provide ways and means to develop appropriate skills in primary school students. As an initial step we can consider the introduction of the subject "Computational Thinking" due to the selective component, and over time in substantiating and experimenting with the content of learning and ways of integration with the basics of science

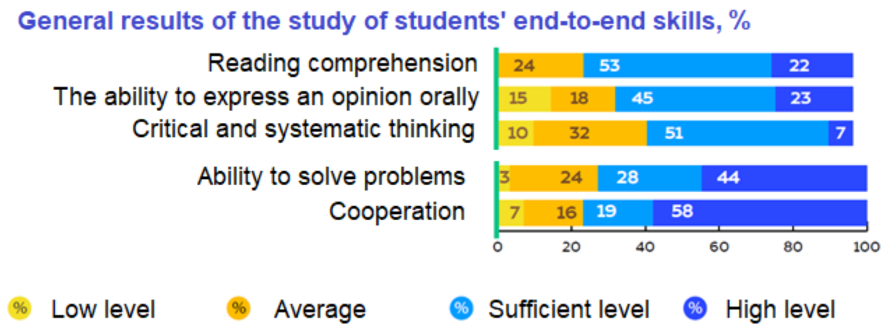

Fig. 7. General results of the study of students' skills.

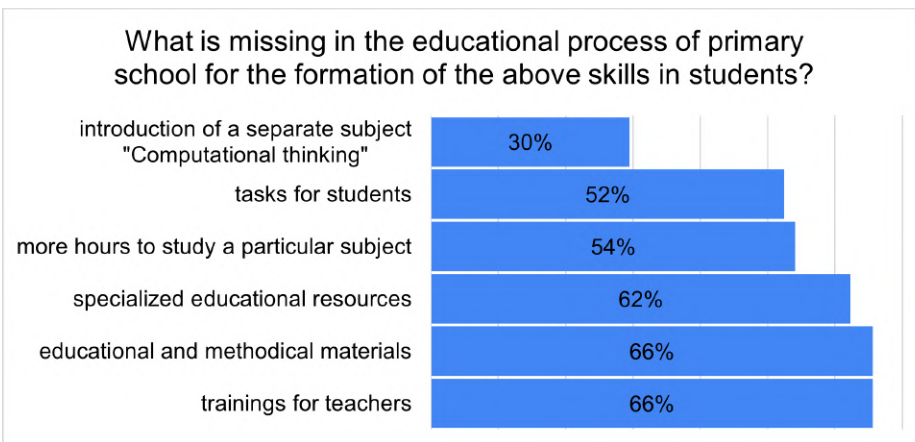

Fig. 8. Respondents' answers to the needs for the formaming CT.

through solving certain integrated competency problems - a separate subject, especially in primary school. In the Ukrainian educational space, not only such open educational resources should be created to help teachers, but first of all it is expedient to develop a concept of teaching teachers and future teachers CT, provide ways and means to develop appropriate skills in primary school students, increase the digital competence of students and teachers [11]. As an initial step we can consider the introduction of the subject "Computational Thinking" due to the selective component, and over time in substantiating and experimenting with the content of learning and ways of integration with the basics of science through solving certain integrated competency problems – a separate subject, especially in primary school.

4 Conclusions

Preparing teachers to teach CT is an important task of all components of teacher education: formal, informal and informal. In Ukraine, despite the widespread inclusion of components of CT in state educational standards, in particular in the Standard of Primary Education, the issue of development of resources for such training is insufficiently resolved. The analysis of the survey showed teachers' misunderstanding of the concept of CT, their unwillingness to form CT in students and interest in learning about approaches that could help with the implementation of CT, unwillingness to use special digital resources. In particular, teachers' requests, learning outcomes of students according to the concept of NUS, educational university programs and resources to support non-formal and informal education allows to build a model of the concept of development of CT in primary school (figure 9).

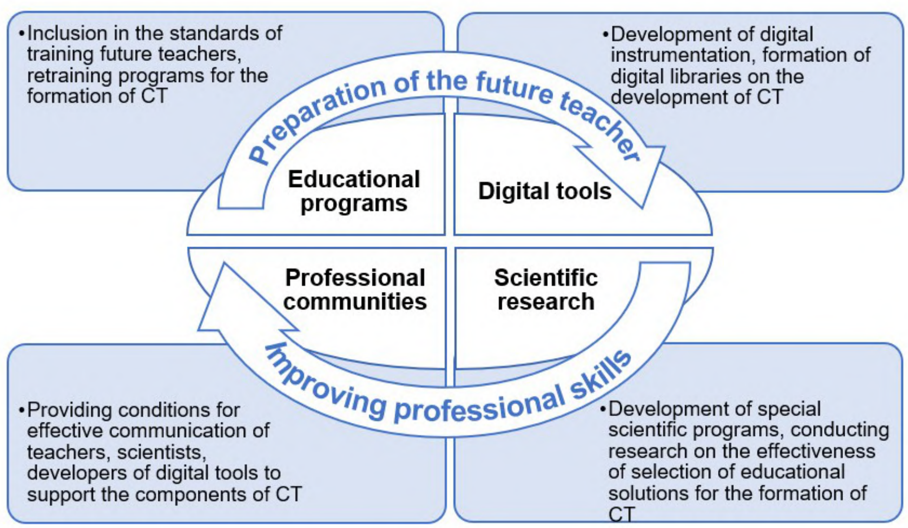

Fig. 9. Conceptual model of teacher preparation for teaching CT.

To implement this concept, you need to provide the following steps: comprehensive integration (Integrate CT across all levels of compulsory education); systematic rollout (Adopt a holistis approach for introducting CT into compulsory education); consolidated understanding (Develop a shared understanding of CN and the relationship with 21st century skills); support policy (foster broad engagement and optimize impact). It is expedient in the system of advanced training of primary school teachers to provide trainings on the development of CT of students based on the use of digital tools and a corresponding elective course in educational programs of future primary school teachers, which would help increase their digital competence. Such programs should include the following sections and an appropriate system of tasks that meet the standard of primary education and are based on competency-based learning and integration of knowledge based on a transdisciplinary approach: decomposition, identification of patterns in various subject areas, generalization and abstraction, development of algorithms and coding. This approach is one of the main in the implementation of STEAM education, including the use of digital technologies. These digital resources for the formation of CT should be included in training programs. Prospects for further research include research and description of specialized digital resources for the formation of CT students, preparation of training programs for teachers, development of a database of tasks with CT.

References

1. Digital education action plan (2021-2027), https://education.ec.europa.eu/focus-topics/digital-education/about/digital-education-action-plan
2. The Future of Jobs Report 2020 (Oct 2020), https://www.weforum.org/reports/the-future-of-jobs-report-2020/
3. Allsopp, B.B.: A playful programming products vs. programming concepts matrix. In: Pivec, M., Gründler, J. (eds.) Proceedings of the 11th European Conference on Game-Based Learning. vol. 1. Academic Conferences and Publishing International, Reading, UK (2017), https://vbn.aau.dk/en/publications/a-playful-programming-products-vs-programming-concepts-matrix
4. Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K.: Developing computational thinking in compulsory education – Implications for policy and practice. Tech. Rep. EUR 28295 EN, European Commission, Joint Research Centre, Luxembourg (2016). https://doi.org/10.2791/792158, https://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf
5. Boiko, M.A.: Development and implementation of electronic learning resources in the process of teaching computer science to elementary school students. The thesis for the degree of Candidate of Pedagogical Science, in specialty 13.00.10 – Information and Communication Technologies in Education, State Institution „Taras Shevchenko National University of Luhansk", Starobilsk (2019), http://dspace.luguniv.edu.ua/xmlui/handle/123456789/4054
6. Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., Woollard, J.: Computational thinking - a guide for teachers. Project report (2015), https://eprints.soton.ac.uk/424545/
7. Duncan, C., Bell, T.: A pilot computer science and programming course for primary school students. In: Proceedings of the Workshop in Primary and Secondary Computing Education. p. 39–48. WiPSCE '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2818314.2818328
8. Fletcher, G.H.L., Lu, J.J.: Human Computing Skills: Rethinking the K-12 Experience. Commun. ACM **52**(2), 23–25 (feb 2009). https://doi.org/10.1145/1461928.1461938, https://cacm.acm.org/magazines/2009/2/19318-human-computing-skills-rethinking-the-k-12-experience/fulltext
9. Fokides, E.: Students learning to program by developing games: Results of a year-long project in primary school settings. Journal of Information Technology Education: Research **16**, 475–505 (2017), https://doi.org/10.28945/3893

10. International Society for Technology in Education (ISTE), Computer Science Teachers Association (CSTA): Operational Definition of Computational Thinking for K–12 Education (2011), https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf
11. Kuzminska, O., Mazorchuk, M., Morze, N., Pavlenko, V., Prokhorov, A.: Digital competency of the students and teachers in ukraine: Measurement, analysis, development prospects. In: Ermolayev, V., Suárez-Figueroa, M.C., Yakovyna, V., Kharchenko, V.S., Kobets, V., Kravtsov, H., Peschanenko, V.S., Prytula, Y., Nikitchenko, M.S., Spivakovsky, A. (eds.) Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kyiv, Ukraine, May 14-17, 2018. CEUR Workshop Proceedings, vol. 2104, pp. 366–379. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2104/paper_169.pdf
12. Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L.: Computational thinking for youth in practice. ACM Inroads **2**(1), 32–37 (feb 2011). https://doi.org/10.1145/1929887.1929902
13. Lu, J.J., Fletcher, G.H.L.: Thinking about computational thinking. In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education. p. 260–264. SIGCSE '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1508865.1508959
14. Morze, N.V., Kuzminska, O.H.: System of information support of formation of masters' ICT competence scientific component. Information Technologies and Learning Tools **44**(6), 42–56 (Dec 2014). https://doi.org/10.33407/itlt.v44i6.1146, https://journal.iitta.gov.ua/index.php/itlt/article/view/1146
15. Pasichnyk, O.V.: Computational thinking in the computer science lessons. Kompiuter u shkoli ta simi (7), 13–18 (2014), http://nbuv.gov.ua/UJRN/komp_2014_7_5
16. Pérez-Marín, D., Hijón-Neira, R., Martín-Lope, M.: A methodology proposal based on metaphors to teach programming to children. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje **13**(1), 46–53 (2018). https://doi.org/10.1109/RITA.2018.2809944
17. Selby, C., Woollard, J.: Refining an understanding of computational thinking. Working paper (December 2014), https://eprints.soton.ac.uk/372410/
18. Sukhomlin, V.A.: International educational standards in the field of information technology. Applied informatics (1(37)), 33–54 (2012), http://www.appliedinformatics.ru/general/upload/articles/PI_137_2012_light_33-renamed.pdf
19. Tsukamoto, H., Takemura, Y., Nagumo, H., Ikeda, I., Monden, A., Matsumoto, K.: Programming education for primary school children using a textual programming language. In: 2015 IEEE Frontiers in Education Conference, FIE 2015, El Paso, TX, USA, October 21-24, 2015. pp. 1–7. IEEE Computer Society (2015). https://doi.org/10.1109/FIE.2015.7344187
20. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (mar 2006). https://doi.org/10.1145/1118178.1118215
21. Wing, J.M.: Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **366**(1881), 3717–3725 (2008). https://doi.org/10.1098/rsta.2008.0118, https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2008.0118
22. Yadav, A., Stephenson, C., Hong, H.: Computational thinking for teacher education. Commun. ACM **60**(4), 55–62 (mar 2017). https://doi.org/10.1145/2994591

23. Zha, S., Morrow, D.A.L., Curtis, J., Mitchell, S.: Learning culture and computational thinking in a spanish course: A development model. Journal of Educational Computing Research **59**(5), 844–869 (2021). https://doi.org/10.1177/0735633120978530