https://doi.org/10.53656/math2021-4-6-the

Educational Matters
Научно-методически статии

THE DEVELOPMENT OF METHODICAL APPROACHES TO THE IMPLEMENTATION OF THE PROPEDEUTIC COURSE OF INFORMATICS IN PRIMARY SCHOOL BY MEANS OF KODU GAME LAB

[1]Dr. Dmitry Igorevich Pavlov
[2], [4]Adel Victorovna Kaplan
[3] Kirill Viktorovich Butarev
[1], [2] State Educational Institution "School №2009" – Moscow (Russia)
[3] State Educational Institution "School №444" – Moscow (Russia)
[4] Federal State Educational Institution of Higher Education Moscow State Pedagogical University Faculty of Primary Education - Moscow (Russia)

Abstract. The article describes the problem of teaching programming at an early school age. The main objective of the article is to demonstrate the results of research on the problems of early learning. The authors analyzed teaching programming in Russian schools, its past, present and prospects. The local (subject) tasks and the global impact of early learning in programming on learning in primary school have been determined.

The authors, as the results of their research, highlight several problems that arise in early learning to program. And they establish the reasons, as well as possible ways to overcome these problems.

Keywords: computer science; elementary school; programming; propedeutics of programming

Introduction

Programming basics training appeared in the course "Informatics basics and computer technology" in 1985 as a mandatory element of education of schoolchildren. The author of this course academician Andrey Petrovich Ershov widely defined tasks of programming training, not reducing them to simple automation of calculation. Ershov in his paper "Computerization of schools and mathematical education" at the Sixth International Congress on Mathematical Education said "the union of three fundamental academic disciplines (language, mathematics and computer science) constitute basis of modern education" (Ershov 1995).

Noting the views of Andrey Ershov on comprehensive nature of computer science, we do not forget his most famous statement, understood most simplistically: “Programming is the second literacy”. Alexander Vladimirovich Goryachev in his report “About reasonability of the modular organization of informatics courses in primary and high school” at the conference “Informatization of Continuing Education – 2018” notes that “programming was a single tool for the development of computers and for the using computers in different sectors of economy in the eighties. The advent of more applications and the expansion of users programming tools contribute to the loss of relevance of programming as a second literacy” (Goryachev).

Despite the trend noted by Alexander Goryachev, teaching programming in school is experiencing a new growth. Now cumulative effect of learning programming is prefer then programming as a profession. Describing this effect, Andrey Ershov spoke about the operational style if thinking determining:

– “Ability to plan the structure of actions necessary to achieve the goal with a fixed set of tools.

– Ability to build information models to describe objects and systems.

– The ability to organize the search for information necessary for the computer solution of the problem.

– Discipline and structuring of language mean of communication.

– The skill of timely access to the computer when solving problems for different subject areas” (Ershov 1979).

In this way, we can say that the operational style of thinking is characterized by two important components – algorithmic, consisting in a formal description of the information process, and its processability divided into operations.

Since the late nineties the beginning of the two thousandth, the term “operational style of thinking” gave way to the notion of “competence”, the main feature of which, in relation to a person of the information society is the change in the assessment of educational quality, when, instead of the amount of knowledge accumulated by student during training, despite their role, the assessment undergoes a qualitative description of the training and first use the skill to put into practice the existing knowledge.

Competence and system-activity approached in teaching had an important impact on the structure and content of the school course of Informatics in the early twentieth century. Currently, the development of the course of computer science and the content line “algorithmization and programming” is influenced by new pedagogical trends. One of them is the development of Computational Thinking.

The term Computational Thinking was proposed by Seymour Paper, but more than ten years it did not attract attention, because the author’s interpretation of this term was specialized. In 2006, Janet Wing, Professor at the University of Southern California, published an article in which she noted: "Computational thinking is a way humans solve problems; it is not trying to get humans to think like 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" (Wing). Today, hundreds of scientific papers both in Russia and abroad are devoted to the interpretation of the concept of "computational thinking" and the ways of its formation. At the international conference on school Informatics ISSEP 2018 (Bosova) held in St. Petersburg, more than a third of the reports were devoted to the development of computational thinking style of preschoolers, schoolchildren and students.

Exploring the phenomenon of computational thinking, Henner Evgeny Karlovich underlines its continuity and fundamental. He notes that "First of all, even on formal grounds, neither "literacy" nor "culture" nor "competence" can compete with the computational style of thinking. Research on computational thinking has repeatedly emphasized that it should not be identified with algorithmic and/or mathematical thinking, computer literacy, or information competence. Algorithmic, logical, system and information thinking, intersecting with computational thinking, do not exhaust it". He also noted that in relation to computer science, the development of computational thinking style "is reflected in the change of programming paradigms and in the transition to the dominant object paradigm in our time, including the stages of object-oriented analysis and design" (Henner). This view has already spread in the scientific and pedagogical society and is the starting point for the development of new approaches to teaching programming in school (Pavlov).

The second trend, the impact of which on the school course of Informatics has already been noted by experts – the basic instrumental literacy. This term was formulated by the participants of the international project "Key Competences and New Literacy". The authors of the project determined that basic instrumental literacy "is based on the use of modern communication tools based on sign systems, implies the transformation in modern techno-logical conditions of habitual literacy "read + write + count" adjusted for the formats of interaction and methods of information transmission, including in the mode of "man – man" and "man – machine" (Frumin).

The basic instrumental literacy is a complex concept consists of:

– "the reader component is the ability to perceive and create information in various text and visual formats, including digital environment (in natural languages)";

– "the mathematical component is the ability to apply mathematical tools, reasoning and modeling in everyday life, including digital environment";

– “computational and algorithmic component is the ability to perceive and create information in formal languages, programming languages”.

Basic instrumental literacy can be considered in the context of the development of the earlier mentioned ideas of Andrey Ershov about the triple nature of literacy. It means that the development of approaches to programming is one of the main tasks at the present stage of development of informatics and education in general.

In the middle of the two thousandth the concept of “elementary informatics course” was introduced into the school course of informatics with its concretization: “elementary (propaedeutic) course of informatics is training in 2 – 6 classes, 3 – 6 classes or 5 – 6 classes, depending on the point of entry into the subject” (Beshenkov). Scientific and methodological substantiation of early education in informatics is given in works of Antipov Igor Nikolaevich, Ershov Andrey Petrovich, Zvenigorodsky Gennady Anatolyevich, Bosova Lyudmila Leonidovna, Beshenkov Sergey Alexandrovich, Goryachev Alexander Vladimirovich, Matveeva Natalia Vladimirovna, Semenov Alexey Lvovich, Pervin Yuri Abramovich and other russian specialists. Their position is conformable with the opinion of world experts (Bottino) (Ibashova). Professor of Comenius University in Bratislava, Ivan Kalas in his works regularly notes not only the importance of early education in informatics and programming, but also the status of informatics as a discipline in the training program of elementary school students. Professor Kalas insists on the mandatory of informatics courses.

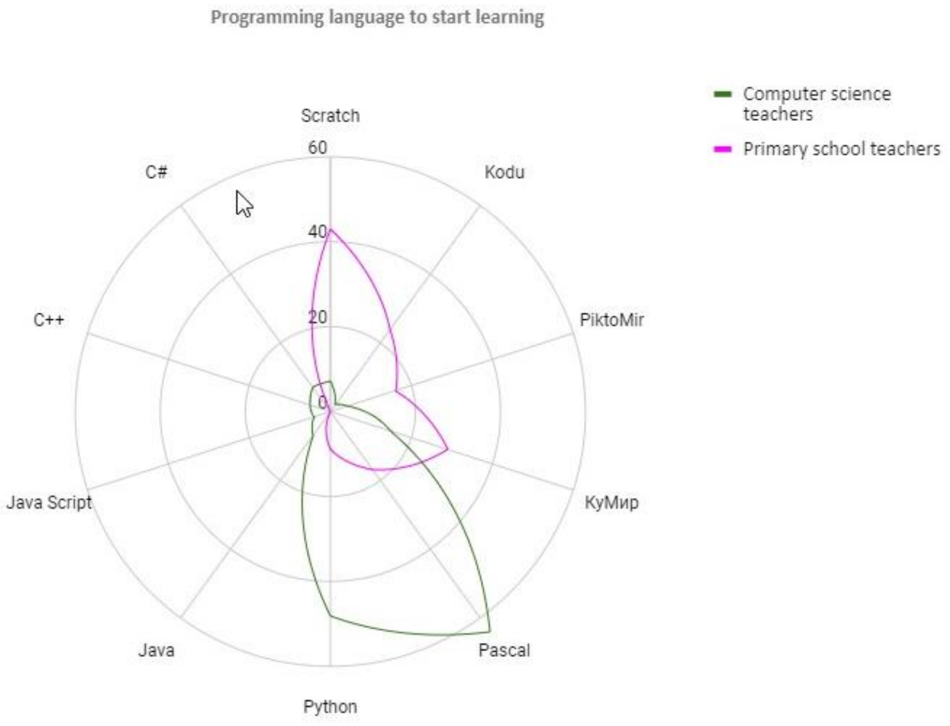
We can say that the problem of propaedeutic programming training at the level of elementary education is actual for Russia and for the whole world (Kabátová).

Methods

We used various methods to research the problem of propaedeutic programming training in elementary education. Passive (theoretical) research methods include research scientific and methodological literature, articles, monographs and dissertation. The ongoing experimental courses can be attributed to active (empirical) research methods.

Significant conceptual contradiction was found as the result of the analysis of scientific and methodological literature. On the one hand, experts in the context of scientific conferences and pages of scientific and methodological literature are about the optimal language of teaching programming (Gorodnaya) (Rodygin). On the other hand, other experts argue that the choice of language is a secondary problem and learning programming is primarily a work on the formation of a special style of thinking (Gladkov).

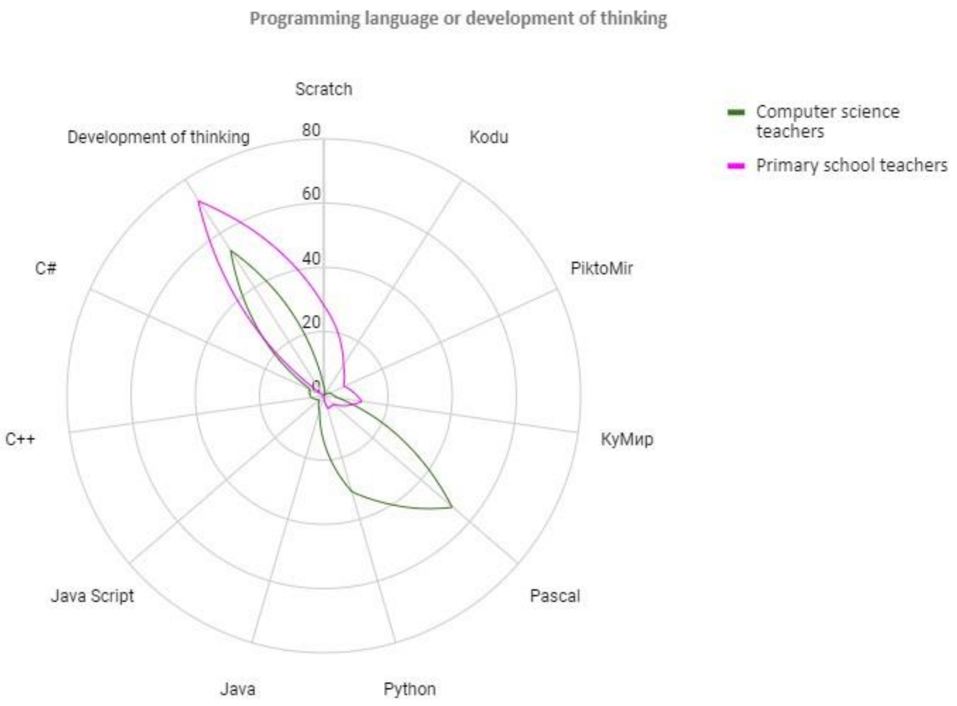
To study this phenomenon, we conducted a survey of teachers. A total of 301 teachers took part in the survey, including 163 informatics teachers and 138 elementary school teachers. The detailed results of the survey are shown on diagram 1.



The survey participants were not offered answers, they named one language at their discretion and based on their understanding of the issues. There are no Java, JavaScript, C++ and C# answers in elementary teachers' answers.

The results of the survey show a serious discrepancy in the choice of the first language for teaching programming in the understanding of elementary school teachers and informatics teachers.

After answering the question, teachers were asked to reflect on the question of what is primary, programming language or the formation of style of thinking, and then asked to give a second answer to the question. The results of the re-poll are shown on diagram 2.



The results show changes in the picture of teachers' opinions with the introduction of an alternative answer. Elementary school teachers demonstrating a greater orientation to the formation of the appropriate style of thinking, and the greatest inertia of thinking is demonstrated by informatics teachers, adherents of the use of Pascal.

In this paper, we have adopted the thesis: "At the propaedeutic programming teaching the first is forming the basis of the relevant thinking style, working out the elements of which can be conducted in different programming languages and using interactive environments".

We studied the leading world experience concerning the age of the beginning programming training. In particular, we studied didactic models of teaching Informatics adopted in Eastern Europe (Grozdev), as well as the results of experiments in learning programming in different age groups of students (Blaho) of different levels of initial training (Gujberova).

The features, traditions and problems of teaching programming in Russian schools (Pervin), as well as promising approaches were carefully studied

(Kaplan). On the basis of the studied materials, the thesis was adopted that "the point of entry into the programming course can be elementary school". This thesis is in contradiction with the level of scientific and methodological development of the topic. For individual languages and interactive environments for early learning programming created lessons planning and sets of didactic materials but teaching programming to elementary school students encounters methodological problems, to solve which there is no appropriate scientific and methodological support.

Based on the analysis of publications in scientific and pedagogical publications, as well as professional communities on the Internet, the following groups of problems can be identified:

– The self-worth of the computer in the child's mind – for many elementary school students, the use of the computer is purpose, not a means to purpose;

– Lack of algorithm-program communication – for many students, a "paper" algorithm is something like a mental exercise, or a mandatory activity before using a computer and has nothing to do with writing a program;

– Game, not learning – for many elementary school students, learning programming is a game, which often leads to blurring of boundaries and the importance of basic concepts such as "performer", "algorithm", "program", "condition", "loop".

To verify the reliability of the data obtained as a result of the generalization of the problems of early programming training, a series of experiments was organized, which consisted in:

– Conducting lessons on teaching programming;

– Conducting extracurricular activities on teaching programming.

The experiments were conducted in 2018 – 2020 in schools in Moscow and the Moscow region. In total during this time it was conducted:

		2018 – 2019	2019 – 2020
CLASSES	The full course of basics of programming in the amount of 34 hours using the Kodu environment	0	1
	The full course of basics of programming in the amount of 34 hours using the Scratch environment	0	1
	The full course of basics of programming in the amount of 34 hours using the code.org environment	0	2
	The full course of basics of programming in the amount of 34 hours using the codecombat.org environment	1	1
	The short course of the basics of programming in the amount of 17 hours using Kodu	2	3
	The short course of the basics of programming in the amount of 17 hours using Scratch	1	2
	The short course of the basics of programming in the amount of 17 hours using codecombat.org	2	2
EXTRACURRICULAR COURSES	The full extracurricular course of basics of programming in the amount of 68 hours using the Kodu environment	2	2
	The full extracurricular course of basics of programming in the amount of 68 hours using the Scratch environment	3	2
	The full extracurricular course of basics of programming in the amount of 68 hours using the code.org environment	1	2
	The short extracurricular course of basics of programming in the amount of 34 hours using the Kodu environment	1	1
	The short extracurricular course of basics of programming in the amount of 34 hours using the codecombat.org environment	1	1

In just two academic years, 34 courses were conducted by 6 teachers, the audience of which was (according to the set) 488 people and adjusted for attendance and dropout of students 407 people. Within the framework of the courses, teachers not only conducted classes on the chosen topic, but also paid attention to the diagnosis of selected problems, in order to establish the causes and relationships. The main element of diagnosis was individual conversations with students.

Results

Confirmations and additional materials were received on three previously identified groups of problems as a result of our research. And in addition, some new problems have been identified.

<u>The self-worth of the computer in the child's mind</u>

The question "Is the computer today?" is the most popular question when children entering the class. It is asked by 3 to 7 before the lesson. The year statistics on the number of questions is not reduced. Under this on question "What we will do with it?" children cannot give the answer on the first half of the year, on the second half of the year children's answers are "Playing" or "As in the last lesson".

In the minds of children 1 – 4 grades computer is the GOAL. At the same time, it is not the goal of training,it is only the GOAL. In 3 of 5 cases students do not even formulate specific wishes, it is just important for them to turn on the computer and press keys.

Our hypothesis that this effect applies to children who do not have a personal computer at home, or parents strongly restrict the student in using computer, was false. This behavior was equally characteristic of students with different levels of access to the computer at home.

To identify the reasons for this relationship with students, parents of students and elementary school teachers conducted individual interviews and group surveys, which resulted in the formulation of the reasons for the "self-worth" of the computer in the minds of children:

– Lack of preparation of children to use the computer, including informing them about what is a computer;

– Children have no experience of using a computer to solve personally significant tasks;

– Lack of experience of using a computer in the lessons (not in informatics lessons).

The lack of connection algorithm-program

Different groups of children took part in the experiment. There were children who had been studying informatics using a PC, children who have passed course of informatics without PC (Unplugged) and children not previously trained in informatics. The program was attended by elementary school students who have programming experience and students who do not have one.

The analysis of the results shows that at the initial stage of training, the problem with the connection "algorithm-program" is more often experienced by students who had no programming experience. Students who have had previous programming experience show much less dynamic in overcoming this problem. 50 percent of them came to the courses with the already formed problem of perception of connection "algorithm-program". At the end of the course, about 40 percent of students were unable to overcome the diagnosed problem. This value was achieved by students who had no previous programming experience, but the initial value in this group was slightly less than 70 percent! This leads to the conclusion that most of the problem is methodical in nature.

In General, the problem has some prerequisites that were identified during communication with teachers and individual conversations with children:

– Intensive algorithmic work in informatics course without PC;

– A quick introduction to the topic of programming without prior theoretical training;

– Big differences between tasks of algorithmic and programming orientation;

– Lack of connection "write an algorithm – realize it in programming environment" in education materials.

<u>Playing, not learning</u>

Game forms and mechanics often create difficulties in perceiving important elements of programming despite the new peak of interest in game technologies in education and the popularity of gamification technologies in early programming training. At the actualization stage of the lesson, at the question "What did we do in the last lesson?" a third of students answer "Played". Waiting to "run" the performer and get the "approval" of the program, students often do not look at the present of a specific task, trying to intuitively perform different sets of actions. The reasons for this perception were established as follows:

Sharp contrast between the excessive classicism of most lessons and the game mechanics in informatics lessons.

During the experiment, some problems were identified, causes' analysis of which was not carried out. Among them:

– The "click for click" problem;

– The problem "not the reader";

– The "Stupid robot" problem;

– Decomposition problem;

– Geometry problem;
– The problem of "equal questions".

Click for click

This problem is typical for pictographic programming environments such as "Pictomir", "Kodu Game Lab" and "Scratch". This problem in text environments has not been identified. General description of the problem: "students choose a block or icons with commands before they are familiar with the task and often cannot answer the question what they are doing". At the initial stage of training, such a problem in different forms was diagnosed in 70 percent of students. Data on the dynamics and method of correction is still insufficient.

Not the reader

General description of the problem: Students say, "Why does not work?" or "Please, help me, I do not understand what to do". When you try to find what the problem is, you find:
– Student did not try to find a solution;
– Student did not read the task.

Especially often this problem is diagnosed when solving problems in code.org, codecombat.com and "PictoMir" (up to 80 percent of students at the first stage of training). In Scratch environment this attribute varies from 70 percent (in the introduction to the topic through solving problems with geometric and mathematical content) to 50 percent (in the introduction to Scratch through animation). There are not enough data for diagnosis. There are observations: students often simply avoid the difficulty by trying to intuitively solve the problem without trying to understand the problem situation; the habit of increased guardianship and delegation of tasks to adults is true for some students. These observations have not been verified.

A special kind of problem "not the reader" is the "choice of numbers" in problems with geometric content. Students skip the task, choosing only numbers from it, and try to set the length, angle, number of repetitions of the cycle by the method of selection, without correlating them with the task at all, drawing per sample.

Stupid robot

General description of the problem: When the result of program does not give the desired result, the student refuses to take on the role of the programmer, whose actions depend on the order of the actions performed by the performer and blames the performer, the computer, bugs.

Decomposition

Decomposition is understood as a scientific method that uses the structure of the problem and allows replacing the solution of one large problem with the solution of a series of smaller problems.

In the context of propaedeutic programming training, the importance of decomposition increases with the development of the topic “loops”.

General description of the problem: Students try to use the loop, not parsing the task into stages. It generates commands like “Repeat 1” with nested. Sometimes such constructions are repeated several times in a row, the construction “Repeat 1” does not cause students doubt.

The difficulty is also caused by sequential tasks. Only a third of students of 2 – 3 classes can independently find a solution in 2 commands repeated 4 times when performing a drawing of a square with the help of a cycle. After parsing the optimal solution, half of the students cannot make a similar problem for the rectangle.

Geometry

Predicting the position of the performer causes difficulty at the initial stage in half of the students, but after 4 – 6 sessions, this problem disappears. What cannot be said about problems with geometric content.

Given that in grade 3 – 4, students do not have the proper knowledge, in particular, about the sum of the internal angels of the polygon, most training programs offer a variety of methods of work, in particular the methods of “selection” and “analogies”. In practice, this is not always possible, and students refuse to link the chains of tasks together and cannot perceive the work with polygons and turns to a certain angle.

Equal questions

Often, when performing a task, the student meets with difficulty and asks the teacher a question and, having received an answer to it, continues to work considering new data. Most of the students, even if they hear the question and answer, but do not perceive it and just a minute later another student can ask the same question. All students can ask the same questions. And if questions on one lesson arise 3 – 5 times, then the entire lesson is under threat. These “answers in the air” are not assimilated by students and can be repeated from lesson to lesson.

The causes of each of the problems require further study.

Summary

Based on the results of this work it is possible to formulate several new areas of research:

– Analysis of existing psychological-pedagogical technologies to overcome the self-worth of the computer in the minds of elementary school students and the formation of ideas about the computer as a tool;

– Analysis of existing pedagogical technologies, forms and methods of work, and their adaptation to solving the problem of breaking the connection between the algorithm and the program;

– Finding a balance between playing and learning programming using game technology;

– For the problems of “click for click”, “not the reader”, “stupid robot”, decomposition, geometry and equal questions, additional research is needed to identify the causes and patterns of manifestation of these problems when teaching programming to elementary school students.

REFERENCES

A. P. Ershov, G.A. Zvenigorodsky, Yu.A. Pervin, 1995. “School informatics-concepts, state, prospects: Preamble to the retrospective publication” in Informatics and Education, *Moscow*, **1**, 320.

A. V. Goryachev, 2018. “On the feasibility of modular organization of the course of computer science in primary and high school” in Informatization of Continuing Education – 2018 (materials of the *International Scientific Conference: in 2 volumes, edited by V.V. Grinshkun)*. Moscow, **2**, 450 – 453.

A.P. Ershov, G.A. Zvenigorodsky, Yu.A. Pervin, 1979. *“School informatics: concepts, conditions, prospects” in preprint no. 152 of the Computing Center of the Siberian Branch of the Academy of Sciences of the USSR*, Novosibirsk, 26.

J. Wing, 2006. *“Computational Thinking” in Communications of the ACM*, **49**(3), 33 – 35.

L. L. Bosova, A.V. Kaplan, 2018. *“International Conference on School Informatics ISSEP 2018” in Computer Science at School*, **6**, 2 – 6.

E. K. Henner, 2016. *“Computational thinking” in Education and science*, **2**(131), 18 – 31.

D. I. Pavlov, K.V. Butarev, E.V. Balashova, 2018. “On the Prospects for Using Gamification Technologies in Early Learning Object-Oriented Programming” in International Scientific Journal Modern Information Technologies and IT Education, **14**(4), 977 – 985.

I. D. Frumin, M.S. Dobryakova, K.A. Barannikov, I.M. Remorenko, 2018. *Universal competencies and new literacy: what to teach today for tomorrow's success. Preliminary conclusions of the international report on school education transformation trends*, Moscow, HSE, 28.

S. A. Beshenkov, E.A. Rakitina, N.V. Matveeva, L.V. Milokhina, 2008. *Continuous course in computer science, Moscow*, BINOM. Laboratory of Knowledge, 216.

R. M. Bottino et al. Coding, 2019. *Programming and the Changing Curriculum for Computing in Schools*, Linz, 45.

A. Ibashova, E. Orazbayeva, A. Kalzhanova, l. Buletova, A. Yesnazar, 2017. *"Formation Of Information Culture Of Primary School Students" in Astra Salvensis*, **10**, 229 – 241.

M. Kabátová, I. Kalaš, M. Tomcsányiová, 2016. *"Programming in Slovak Primary Schools" in Olympiads in Informatics*, **10**, 125 – 159.

L.V. Gorodnaya, A.S. Boyarsky, 2012. *"On the language of primary education in parallel programming" in Bulletin of NSU*. Series: Information Technology, **2**, 85 – 100.

E.F. Rodygin, 2011. *"Guidelines for teaching programming in school" in Bulletin of the Mari State University*, **7**, 20 – 22.

V.P. Gladkov, 2009. *"Methods of teaching programming" in Bulletin of PNIPU. Electrical engineering, information technology, control systems*, **3**, 94 – 101.

S. Grozdev, T. Terzieva, 2015. *"A didactic model for developmental training in computer science" in Journal of Modern Education Review*, **5** (5), 470 – 480.

A. Blaho, I. Kalas, M. Matusova, 1995. "*Experimental curriculum of informatics for 11 year old children" in IFIP World Conference on Computers in Education*, Springer, Boston, MA, 829 – 841.

M. Gujberova, I. Kalas, 2013. *"Designing productive gradations of tasks in primary programming education"in Proceedings of the 8th Workshop in Primary and Secondary Computing Education, Aarhus*, ACM, 108 – 117.

Yu. A. Pervin, 2005. *"Problems of early learning in computer science in a Russian school" in Issues of Education*, **6**, 11.

A.V. Kaplan, D.I. Pavlov, 2019. *"Development of methodological approaches to the implementation of the propaedeutic computer science course in elementary school by means of the Kodu Game Lab" in Computer science and education,* **8**, 14 – 23.

✉ **Dmitry Igorevich Pavlov, PhD in Education, Senior Lecturer**
ORCID iD: 0000-0002-0074-0899
Institute of Mathematics and Informatics
Moscow State Pedagogical University – MSPU
14, Krasnoprudnaya St.
Moscow, Russia
E-mail: di.pavlov@mpgu.su

✉ **Adel Victorovna Kaplan, Primary School Teacher**
ORCID iD: 0000-0002-8581-5086
School №2009
Federal State Educational Institution of Higher Education
Moscow State Pedagogical University
Faculty of Primary Education - Moscow (Russia)
16, Admiral Rudneva, bldg. 1
117042 Moscow, Russia
E-mail: adel.caplan@ya.ru

✉ **Kirill Viktorovich Butarev, Secondary School Teacher**
ORCID iD: 0000-0001-9387-1267
Graduate student of Moscow State Pedagogical University – MSPU
School №444
14, Nizhnyaya Pervomayskaya St.
105043 Moscow, Russia
E-mail: fallenmonk@mail.ru