Problems of Teaching Pupils of Non-Specialized Classes to Program and Ways to Overcome Them: Local Study

Yuliya Rudenko[†], Marina Drushlyak[††], Nataliia Osmuk[††], Olha Shvets[†††],

marydru@fizmatsspu.sumy.ua
Oleksandr Kolyshkin[††] and Olena Semenikhina[††]
[†]Sumy College of Economics and Trade, Sumy, Ukraine
[††]Makarenko Sumy State Pedagogical University, Sumy, Ukraine
[†††]Sumy National Agrarian University, Sumy, Ukraine

Summary
The development and spread of IT-technologies has raised interest in teaching programming pupils. The article deals with problems related to programming and ways to overcome them. The importance of programming skills is emphasized, as this process promotes the formation of algorithmic thinking of pupils.
The authors determined the level of pupils' interest to programing learning depending on the age. The analysis has showed that the natural interest of younger pupils in programming is decreasing over the years and in the most productive period of its study is minimized. It is revealed that senior school pupils are characterized by low level of interest in the study of programming; lack of motivation; the presence of psychological blocks on their own abilities in the context of programming; law level of computer science understanding.
To overcome these problems, we conducted the second stage of the experiment, which was based on a change in the approach to programing learning, which involved pupils of non-specialized classes of senior school (experimental group). During the study of programming, special attention was paid to the motivational and psychological component, as well as the use of game technologies and teamwork of pupils.
The results of the pedagogical experiment on studying the effectiveness of teaching programming for pupils of non-specialized classes are presented. Improvement of the results provided the use of social and cognitive motives; application of verbal and non-verbal, external and internal means; communicative attacks; stimulation and psychological setting; game techniques, independent work and reflection, teamwork. The positive effect of the implemented methods is shown by the results verified by the methods of mathematical statistics in the experimental and control groups of pupils.
Key words:
programming, teaching to program, pupils of non-specialized classes, algorithmic thinking, approaches and learning technologies.

1. Introduction

Education in the field of information technology (IT industry) today involves the presence of programming skills, which, among other things, means not programming language knowledge (there are already several thousand of them), but how to see the algorithmic way of solving the task and to write the successive steps of such a solution. Solving various, including mathematical problems, visualization and formalization of the steps of the solution, verbal description of the phased decision of the problem contribute to the development of algorithmic thinking [1]. At the same time, we now have a situation where interest in the programmer's profession in Ukraine starts to fall, as evidenced by the statistics of admission to the relevant specialty (in particular, in 2018, the number of applications for the IT industry has decreased more than three times . At the same time, the social demand of Ukrainian society for the development of its own economy is growing at the expense of the IT industry, and therefore we see the importance of identifying the reasons for the emergence of such a contradiction.

Since at the entrance the university, the demand for a programmer's profession is diminishing, it is natural to assume that the decline in interest in this profession is due to schooling. On the other hand, the greatest influence on the choice of the programmer's profession is at the Informatics classes. Moreover, the first skills of algorithmization and programming are formed at the Informatics classes. Therefore, the reasons why young people are less interested in programming and IT professions, we have been looking at the problems of informatics learning at school when they receive secondary education.

Let us note that Ukrainian secondary education has two levels: basic school (5-9th grades, 10-15 years old), senior school (10-11th grades, 15-17 years old). The study of Informatics begins in 2nd grade, but the study of programming begins in 6th grade.

The senior school assumes profiling (focusing on the future profession) on particular fields of knowledge. When choosing a particular profile, the curriculum envisages an increase in the time for studying profile subjects. Therefore, when choosing a profile related to the IT industry, the Informatics course in 10-11th grades increases twice from 70 to 140 hours. More detailed information on the

Manuscript received December 5, 2021
Manuscript revised December 20, 2021
https://doi.org/10.22937/IJCSNS.2022.22.1.16

distribution of hours for the study of Informatics in general and Programming in particular, are given in Table 1.

Table 1: Distribution of hours for the study of Informatics and Programming

Grade	Number of hours per week	Informatics, basic level	Programming, basic level	**Programming, profile level**
5th	1	35	-	-
6th	1	35	7	14
7th	1	35	9	18
8th	2	70	28	20
9th	2	70	10	56
10th	1	35	6	28
11th	1	35	8	-
Total		315	68	136

The analysis of the number of 9-10th grade pupils (2018-2020, 431 persons), which has already been determined with the study profile, shows that the proportion of pupils who choose physical and mathematical and technological profile in the senior school decreases from 55% to 38%. In the same time, the number of pupils in humanitarian classes increases from 45% to 62%. Due to these circumstances, we investigated the feature of Informatics and Programming study in non-specialized classes (10th grade) of senior school and 5-9th grades of basic school.

Thus, the purpose of the article is to analyze the problems associated with teaching the pupils of non-specialized classes of senior school to program, and a description of the way to overcome them.

2. Analysis of recent findings

The study of scientific works on the training of programming and closed topic has revealed a number of findings that cover the conceptual framework of the programming study. Scientiscs note problems of educational gaps in the “school-university” chain during the study of programming languages. M. Zhaldak [2] notes the contradiction between the amount of necessary knowledge of programming and the limited time to learn them in schools and offers its solutions through the involvement of active teaching methods and changing the order of learning programming languages.

Let us focus the analysis of findings on the problem of using non-traditional approaches in teaching programming. In this direction it should be noted the results of I. Mintii [3], who offers to use the theory of inventive solutions – the technology of creativity in learning programming. O. Spivakovskyi [4] proposes activity approach, active learning methods, which cause the increase of learning motivation, activation of pupils’ cognitive activity, development of ability for independent learning, development of teamwork skills, and correction of the children self-esteem. I. Dedinskii [5] focus attention on a methodological project approach, as an effective solution to the problem of formation of youth algorithmic thinking. However, S. Ishcheriakov [6] stresses on the effectiveness and efficiency of the problem approach of programming study, which is based on the principle "one problem – one solution". V. Bazurin [7] recommendes to use a visual approach based on the use of a visual programming language with simple syntax (HTML+CSS, JavaScript, Python). The author notes that it is advisable to use different environments for bigginers and pupils that already have programming skills. S. Papadakis [8], J. da Silva & I. Silveira [9] propose a game approach for programming learning and teaching to achieve a positive effect. According to Y. Yan, H. Nakano, K. Hara, T. Kazuma & A. He [10] independent work of pupils is effective way to master programming skills.

Consequently, the problem of content, quality and level of pupils’ preparation for the ability to program is the focus of researchers. However, despite the considerable experience gained in this sphere, the state's awareness of the importance of quality training of young people in the field of programming, the issues of identifying and overcoming the problems that lead to a low level of interest (or unwillingness) of pupils to learn to program remain relevant.

3. Research Methodology

The study involved three stages. The first stage implies the literature analisys, finding open questions, conducting a survey on interest in programming and analysis of results. The second one is for clarification of the purpose of the study and theoretical justification of ways to achieve it. The third stage is applied for the organization of pedagogical experiment, definition of indicators of

efficiency of experimental training, the quantitative analysis of empirical data and a qualitative estimation of results of pedagogical experiment.

3.1. Research Data Collection

The following data collection methods were used.

- Documantain review (Imformatics curriculum for secondary education) was used to determine the number of hours devoted to learning the basics of programming and to list the relevant knowledge and skills of students.
- Questionnaire with open-ended question for the determining the level of pupils' interest to learn to program depending on the age. The questionnaire contained the one question: "How do you feel about learning to program at Informatics lessons? Why?"
- Conducting of test to determine the level of academic achievements. We offered pupils to solve two tasks of the following type.

1. Work of the traffic light is programmed as follows: from the beginning of every hour, the red light is on for 3 minutes, and then the green one is on for 2 minutes. Time is entered from the keyboard. What signal will the traffic light be in this case? (up to 5 points)

2. A citizen opened an account for UAH 10 000 with a monthly rate of 2% at the bank on March 1. How many months will the amount of deposit exceed 10 500 UAH? (up to 7 points).

3.2. Research Data Analysis Tools

As we conducted a qualitative research at the first stage, the questions of the questionnaire were open. Pupuls of 10-11th grades took part in the survey. The survey was not conducted among students of specialized physics and mathematics classes. Empirical data were accumulated until the answers to the questionnaire questions were repeated.

Learning results of pupils of the experimental and control groups at the third stage of the study were divided into four levels:

Low – 1 – corresponds to the evaluation of educational achievements within the range of 1 to 4.

Average – 2 – corresponds to the evaluation of educational achievements within the range of 5 to 7.

Sufficient – 3 – corresponds to the evaluation of educational achievements within the range of 8 to 10.

High – 4 – corresponds to the evaluation of educational achievements within the range of 11 to 12.

The results of the pedagogical experiment (the third stage of the study) were analyzed using statistical tests. Pearson's test was used to test the hypothesis of normal data distribution. Student's test was used to compare the mean values in the samples [11].

3.3. Research Participants

The base of the study was the educational institutions of Sumy region (Ukraine). The total number of respondents was 469 people: 54 Informatics teachers, 200 basic school pupils, 215 senior school pupils. The selection criterion of the participants was the involving the pupils from non-specialized classes. Pupils' participation was voluntary and they were provided with all necessary information about the experiment procedures, purposes and outcomes.

Participants in the pedagogical experiment (the third stage of the study) were divided into two groups – experimental and control. The participants of the control group studied programming using traditional teaching methods. Non-traditional methods (the motivation to study of programing; the stimulation and psychological aspiration; the use of game technologies; the use of teamwork; independent work and reflection) were involved in the process of teaching Informatics of participants of the experimental group.

3.4. Research ethics

All participants of the research gave their voluntary informed consent in the participating in the study. They were informed about the right to withdraw at any time, the process in which they were engaged, including why their participation is necessary, how it will be used and how and to whom it will be reported and agreed without any duress. All research ethics practices were followed according to Helsinki guidelines to avoid negatively affecting participating human subjects. This includes, but is not limited to, the requirement of anonymity for all participating in questionnaire survey individuals for greater objectivity. Seeking to ensure confidentiality and de-identification, the research participants in tests were encoded. This allows the authors to be confident in the reliability of the data. The data collection was been performed according to educational research ethics based on AERA Code of Ethics and BERA Ethical Guidelines for Educational Research.

3.5. Research Procedure

The authors selected pedagogical experiment as an empirical method of scientific research, which allows in practice determining the ways to overcome the problem of teaching pupils to program.

The study lasted two years (2018-2020). At the first stage (2018), factors that reduce the effectiveness of pupil programming learning were identified by conducting questionnaires of pupils. At the second stage (2018), changes in approaches to programming learning were identified and justified. The third stage (2019, 2020) provided for the organization and conduct of a pedagogical experiment.

4. Research Results

The solution of the research aim at the ascertaining stage provides determining the level of pupils' interest and the willingness to learn to program depending on the age (questionnaire). The survey was not carried out among pupils of specialized physical and mathematical classes. The sharpness of contradictions in the interest to learn programming languages was found. Answers to this question of the questionnaire distributed as follows (Table 2).

Table affirms the decline of interest in programming from 60% (5th-6th grades) to 23% of the pupils (9th-11th grades and only 5% learn to program it on their own, 18% express interest and want to learn). A comparison of the questionnaire results also shows that senior school pupils have low self-esteem regarding the hope to learn to program. Low self-esteem is manifested in psychological barriers ("I will never understand it" – 26%, "Programming is too complicated and boring" – 23%, " I do not need, I am humanitarian" – 28%). The analysis of answers to other questions of the questionnaire has showed similar results. As a result, in 9-10th grades we have only 17% of those who want to study programming.

Table 2: Distribution of pupils' answers

Answer		5th-6th grades		9th-10th grades	
I want to study	It is very interesting. I study on my own.	120/60%	154/77%	11/5%	50/23%
	I like to study.	34/17%		39/18%	
I don't want to study	Programming is too complicated and boring.		26/ 13%	48/23%	165/77%
	I do not need, I am "Humanitarian"			61/28%	
	I do not want to study.	26/13%			
	I will never understand it.			56/26%	
I don't know		20/10%	20/ 10%	-	-

The content analysis of answers reflecting attitude of pupils to the programming in 5th, 6th grades and 9th, 10th grades allowed us to make such a conclusion – the natural interest of younger pupils in programming is decreasing over the years. It is revealed that senior school pupils are characterized by:

– low level of interest in the study of programming;
– lack of motivation;
– the presence of psychological blocks on their own abilities in the context of programming.

We assume that this situation is caused by pedagogical approaches and technologies used in senior school in the study of programming.

Traditional approaches to teaching programming are as follows. First, the language alphabet and types of variables are described. Then the teacher describes the solution of a typical problem according to the "Do as I" principle. Then the analogues problems are offered for an independent solution. Teacher, as a rule, does not use technologies of activating cognitive activity, special methods, etc.

To overcome these problems, we conducted the second stage of the experiment, which was based on a change in the approach to study of programing. In the second stage of the experiment during the study of programming of pupils of non-specialized classes of senior school (experimental group) special attention was paid to:

– the motivation to study of programing;
– the stimulation and psychological aspiration;
– the use of game technologies;
– the use of teamwork;
– independent work and reflection.

At the beginning of the experiment it was found that the level of pupils' knowledge on programming was statistically identical.

To test the effectiveness of the proposed strategies, we conducted control tests of acadamic achievements and offered pupils to solve two tasks.

At the end of the experiment, the average score of educational achievements of the control and experimental groups were also studied. They showed a statistical difference at the significance level of 0.05: in the experimental group, the average score was higher, which is explained by chosen learning strategies (Table 3, Fig. 1 and Fig. 2).

Table 3: Evaluation of the average score of control and experimental groups (at the significance level of 0.05 according to Student's test)

	Respondents	At the beginning	At the end	Note
CG	104	6,80	7,6	Tstat=0,4 < 1,97 = Tcr at the beginning of the experiment. Tstat=|-2,5|>1,97=Tcr at the end of the experiment.
EG	111	6,66	8,4	

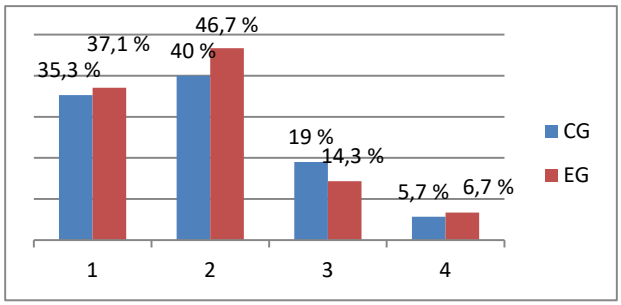

Fig. 1. Level of pupils' educational achievements at the beginning of the experiment

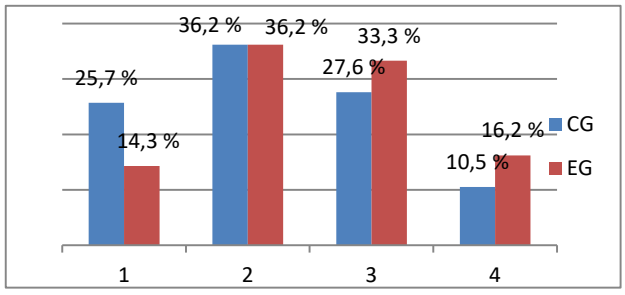

Fig. 2. Level of pupils' educational achievements at the end of the experiment

The analysis of the post questionnaire "How do you feel about learning to a program at Informatics lessons?" found that the interest to program of pupils of the 10th grade increased from 23% to 67%, and the number of those who began to study programming independently increased from 5% to 24%.

5. Discussion

The actuality of this problem is noted not only in Ukraine. Analysis of the latest periodicals covering the current state and problems of the IT-industry confirms such a conclusion [12, 13]. In particular, Ginni Rometty, the head of company IBM, in the letter to Donald Trump, the former President of the United States, notes the insufficient quality of teaching American pupils programming and suggests supporting special six-year curriculum of learning programming in the high school [14]. To increase interest in programming in schools in France, a number of activities are carried out; in particular, 2014 was declared the year of the Code [15]. The basics of programming as a subject is introduced in primary schools in Estonia, Austria, Finland, Poland and other countries [13].

However, the increase in the number of hours and disputes over the choice of programming language, which is offered to study at school, do not solve the problems of formation of skills to program, which are quite acute in Ukrainian schools. Therefore, the authors turned to the involvement of non-traditional methods in teaching pupils of non-specialized classes to program. Among such methods we have identified the following.

Motivation – the use of social (understanding of social importance, needs, focus on the future) and cognitive (interests, emotions, ideals, desire) motives.

Because motivation contributes to the willingness to learn, encourages to act with maximum energy, than techniques became appropriate (training based on emotional suggestion, which involves the integrated use of all verbal and non-verbal, external and internal means [16], communicative attacks, proofs and persuasions that are actively used during the first classes [4, 5]. By these techniques, we tried to interest with the training material all conventional categories of pupils (the "Humanitarians", "Mathematicians", "Sportsmen" and others). The importance of studying programming was supported by "vital" arguments, for example, such: "It is natural for a person to be in a comfort zone. You have already tentatively identified your future path and tried to avoid unnecessary information, you are sure that you do not need it. However even if you are not going to devote your life to computer technology, it is very useful to have the skills of algorithmic thinking, formation of which contributes to the programming study. Algorithmic thinking skills are applied in a variety of spheres, from travel planning to public health. To make the right decisions aimed at a certain result, it is always useful to highlight the main elements of the problem and understand how they interconnected. If you have mastered these methods (algorithms, approaches), everything becomes much easier. Therefore, we suggest you trying to think in computer categories and make sure that such thinking is logical, structured, consistent, and, as a result, effective.

Stimulation and psychological aspiration. To remove the block, which slows down understanding of programming, there were given examples and statistics of how many people with non-mathematical mind set reach success in programming, what percentage of people have

become programmers already in the adulthood. The bias against programming of girls we tried to change, noting that the first programmer in the world was a woman – Ada Lovelace [17].

The rise and initial emotional lift of the first lesson was maintained throughout the whole course of study. According to I. Dedinskii [5], while teaching programming, it is important not only to teach, but also to educate, so we especially paid attention to the formation of such key psychological qualities of the personality as efficiency, self-esteem, sociability, responsibility, effectiveness, creativity, imagination.

The diametrically opposite problem that slowed down the programming learning process was the illusion of some pupils in their own competence to work with computer technology (they know how to download music or install Skype). For such pupils were developed special tasks, performing which, they personally could make sure of the absence of the necessary knowledge and skills. According to their own learning results and individual tasks, an individual educational trajectory with the study of programming techniques was built for them.

At this step, the work of program code and programming environments was demonstrated on the example available to pupils. The positive results of the training were promoted by the application of a visual approach based on the use of a visual programming language with simple syntax (HTML+CSS, JavaScript, Python) [7].

Game methods. Support of pupils' motivation and their attitude to the study of programming necessitated the use of appropriate techniques that would allow teaching and educating effectively. Focusing on the slogan "Programming is a game" and the pedagogical experience of experts, special attention was paid to gaming technologies. The choice of such a strategy was based on [8, 9]. According to the scientists, beginners begin to program for two reasons – learning and entertainment. In the first case, the interest fades because of the boring process and frequent confusion and misunderstanding of the algorithms logic. However, in the second case, where programming is regarded as a game, there is interest in programming and enthusiasm of pupils, which determines further success.

The nature of the game is clear and attractive to pupils because of its brightness, and emotional rise, the ability to compete and win. Learning the programming and mastering the skills and strategies of any game have a lot in common. A sequence of steps to understand the rules and the concept; failure at the first attempt; excitement in the pursuit for victory, which forces to act until the last; a thorough analysis of tactics that need to be mastered; emotional rise from the result (victory). An additional positive effect is the transition of the pupil to higher levels of hierarchy in the game, and therefore during the programming training. This further encourages intensive programming practice and assimilation of the basic methods of programming.

The attitude to programming as a game determines the active use of role-playing games in the modelling situations, competitions and other interactive methods that can explain visually and clearly the complex elements of programming, which usually cause the greatest difficulties and loss of interest. Therefore, to understand the concept of the algorithm and the properties of the algorithm firstly game-moderation is conducted. Pupils become conditional computers and must complete a specific task. For example, if you have a loaf, butter, jam, knife (input), you have to make a sandwich (result). Pupils must ensure necessary algorithm properties (finiteness, definiteness, input, output, effectiveness). After such a game, there is no longer a problem with the explanation of these properties. Pupils determine the type of construction (linear). The cyclic construction is well shown by filling 3-liter jar of water with half a litter jar. While reproduction of all constructions all actions are recorded as pseudo-code.

Teamwork. Modern IT-industry is mainly focused on teamwork on projects [18, 19], so pupils mostly learned to work in a team on joint projects. For attracting to the team work it was used the practice of cross-code exchange with the task to understand someone else's code and to supplement it, and to write reviews on someone else's code. The exchanges of thoughts, knowledge, help, and joint generation of ideas were encouraged. The teamwork activities contributed to the socialization of pupils and developed their communication skills.

Independent work and reflection. Independent work of pupils is the main means of mastering educational material during free from compulsory educational tasks time and is an important component in training to programming [10]. Independent work was interesting because of the possibility of networking, in particular, there were used such interactive web services as learningapps.org, www.pythontutor.com, www.e-olymp.com, http://cppstudio.com, designed to support the educational process, in particular, the practice of programming [20]. One of the options for interaction with pupils became the teacher's offer to perform certain tasks, import them in unified network presentation using Google Docs service. The advantage of the chosen service was the presence of built-in chat, which made it possible to discuss the results online.

At each lesson, attention was paid to the development of reflective thinking, which was formed by analyzing the tasks performed and reflected in the answers to the questions: "Why is this task necessary? How to apply it in practice? What difficulties arose? Are there alternative ways to solve it?" In addition, pupils were asked to give examples that would demonstrate the implementation of ideas into practice.

These aspects will be effective only in the conditions of an integral educational, subject-developing environment, which significantly affects the consciousness and behaviour of the personality.

The effectiveness of authors results is confirmed for secondary school. It should be noted that until the 9th grade pupils do not carry out professionally oriented work. But there are situations where although a pupil is interested in Informatics, but the complexity of some topics or insufficient amount of time to master them negatively impact on the willingness to study Informatics, and therefore programming. Taking into account the pupil's certainty in choosing a senior school profile it is considered inappropriate additional motivation and active learning methods. In this period, non-specialized subjects, unlike profile ones are perceived as not important. Therefore, the Informatics teacher has a task to teach programming that does not attract. At the same time, awareness of the leading role of IT in society and programming as the main means of its implementation, on the one hand, and the perception of each pupil as a creative personality and able to manifest themselves in IT, on the other hand, prompt changing the established approaches to teaching Informatics and Programming. Therefore, we believe that the strategies for teaching pupils to program in non-specialized classes based on enhanced motivation to study programming, on stimulating learning activities through personal examples of success and on psychological guides to success, on game technologies and teamwork, as well as on independent work and reflection of one's own activities for senior school are also effective.

Let us note that selected approaches to learning (motivation to study of programing; the stimulation and psychological aspiration; the use of game technologies; the use of teamwork; independent work and reflection) were considered in the complex, and therefore we cannot talk about the effectiveness of each approach separately. At the same time, we note that the increase in motivation took place at the first lessons, in particular, when studying the topics "Stages of computer solving of problem", "The notion of programming languages". Stimulation of the study of programming and psychological aspiration were used in the study of the topic "Embedded algorithmic structures. Repetition and branching". Game techniques were used when studying topics "Change the properties of an object in the program", "Creating a program that displays a message window". Teamwork was used when performing learning projects. Independent work and reflection were used when performing home-based tasks of practical content.

6. Conclusions

The dissemination of the experiment results among teachers contributed to the change of traditional approaches to Informatics teaching and the reflection of their own professional activities. The psychological factors that negatively influence the studying of programming (the low level of interest in the study of programming, the lack of motivation; the presence of psychological blocks on their own abilities in the context of programming) can be eliminated by changing the approaches to teaching (factors) that are effective in teaching secondary school pupils.

Such factors are motivation, stimulation and removal of psychological blocks for the study of programming, teamwork, as well as technological, among which we distinguish game technologies and balanced and pre-thought-out independent work and reflection of learning activities.

The analysis of foreign findings, which highlight the problems of learning to program, shows the following. The growing relevance and at the same time fragmentary solution / unresolved of these problems. Active search for effective ways of interest of young people in mastering programming skills both at the level of individual teachers, the scientific community within educational institutions, and at the level of state administrations around the world. This determines the expediency of taking into account local experience, in particular Ukraine, in solving this problem.

The prospects of further scientific work are aimed at deepening the research of psychological and pedagogical impact on the quality of teaching programming of primary and secondary school children, as well as expanding the range of use in the educational process of pedagogical technologies of distance learning and the use of gaming technologies.

References

[1] *Unified State Electronic Educational Base.* URL: https://vstup2018.edbo.gov.ua/statistics/konkurs-universities/ (2019).

[2] Zhaldak, M. I.: Informatics is a fundamental scientific discipline, *Computer in school and family*, 2, 39-43 (2010).

[3] Mintii, I. S.: Forms of training organization for the formation of competencies in programming, *Pedagogy of Senior and Middle School*, 41, 91-96 (2014).

[4] Spivakovskyi, O. V.: Future of School Informatics, *Naukovyi chasopys NPU imeni M. P. Drahomanova*, 3, 226-234 (2010).

[5] Dedinskii, I.: Analytical campaign for pre-university teaching of programming. URL: http://ded32.net.ru/news/2011-04-03-58 (2011).

[6] Ischeriakov, C.: *Should I study programming at a school or university?* Educational policy. Portal of Public Experts. URL: http://osvita.ua/school/54063/ (2017).

[7] Bazurin, V.: Programming environments as a means of teaching pupils to programming basics, *Information technologies and learning tools*, 59 (3), 13-27. doi: 10.33407/itlt.v59i3.1601 (2017).

[8] Papadakis, S.: Evaluating a game-development approach to teach introductory programming concepts in secondary

education, *International Journal of Technology Enhanced Learning*, 12, 2, 127-145 (2020).

[9] da Silva J., Silveira, I.: A Systematic Review on Open Educational Games for Programming Learning and Teaching, *International Journal of Emerging Technologies in Learning*, 15, 9, 156-172. doi: 10.3991/ijet.v15i09.12437 (2020).

[10] Yan, Y., Nakano, H., Hara, K., Kazuma, T., He, A.: A Web Service for C Programming Learning and Teaching, *Proceedings of 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems*, 414-419 (2016).

[11] Rudenko, V. M.: *Mathematical Statistics*. Kyiv: Center for Educational Literature (2012).

[12] Johnson, P.: *Reading, writing and refactoring: How 7 forward-thinking countries are teaching kids to code, IT-World.* URL: https://www.itworld.com/article/2824003/cloud-computing/161754-Reading-writing-and-refactoring-How-7-forward-thinking-countries-are-teaching-kids-to-code.html (2014).

[13] McCaskill, S.: *New National Curriculum To Teach Five Year Olds Computer Programming.* URL: http://www.silicon.co.uk/workspace/national-curriculum-ict-education-computing-121214 (2013).

[14] *Ginni Rometty's Letter to the U.S. President-Elect*., IBM CEO. URL: http://ibm.com/blogs/policy/ibm-ceo-ginni-romettys-letter-u-s-president-elect (2017).

[15] *Year of Code and £500,000 fund to inspire future tech experts launched.* URL: http://services.parliament.uk/bills/2015-16/educationandadoption.html (2014).

[16] Rudenko, Yu. O.: Professional adaptation of young teachers of computer science, *Fizyko-matematychna osvita – Physical and Mathematical Education*, 1(15), 281-284 (2018).

[17] Coe, I., Ferworn, A.: The Life and Contributions of Countess Ada Lovelace Unintended Consequences of Exclusion, Prejudice, and Stereotyping, *IEEE technology and society magazine*, 35 (4), 46-49. doi: 10.1109/MTS.2016.2618679 (2016).

[18] Gonzalez-Guerra, L., Leal-Flores, A.: Teamwork based learning: using an automatic tutor in programming courses, *13th International Technology, Education and Development Conference (INTED2019),* 72-78 (2019).

[19] Trejos-Buritica, O.: Methodology teamwork's 4321 to improve learning in computer programming using collaborative learning and cooperative learning, *Revista Educacion en Ingenieria*, 13, 25, p. 28-36. doi: 10.26507/rei.v13n25.821 (2018).

[20] Jung, I., Choi, J., Kim, I., Choi, C.: Interactive Learning Environment for Practical Programming Language based on Web Service, *15th International Conference on Information Technology Based Higher Education and Training (ITHET),* doi: 10.1109/ITHET.2016.7760705 (2016).