Scientific journal
PHYSICAL AND MATHEMATICAL EDUCATION
Has been issued since 2013.

Науковий журнал
ФІЗИКО-МАТЕМАТИЧНА ОСВІТА
Видається з 2013.

ISSN 2413-158X (online)
ISSN 2413-1571 (print)

http://fmo-journal.fizmatsspu.sumy.ua/

Nosachenko D. Analysis of methods of evaluation and selection of tasks for the programming olympiad: features and success criteria. Фізико-математична освіта. 2020. Випуск 2(24). Частина 2. С. 73-78.

Nosachenko D. Analysis of methods of evaluation and selection of tasks for the programming olympiad: features and success criteria. Physical and Mathematical Education. 2020. Issue 2(24). Part 2. P. 73-78.

DOI 10.31110/2413-1571-2020-024-2-034

D. Nosachenko
Sumy State Pedagogical University named after A.S. Makarenko, Ukraine
ddddddaaaaa16@gmail.com
ORCID: 0009-0007-4564-9806

ANALYSIS OF METHODS OF EVALUATION AND SELECTION OF TASKS FOR THE PROGRAMMING OLYMPIAD: FEATURES AND SUCCESS CRITERIA

ABSTRACT

This article explores the significant role of Programming Olympiads in computer science education. Olympiads serve as a platform for enhancing computational thinking and problem-solving skills among students. They foster creativity, logical reasoning, and algorithmic thinking by challenging participants to solve complex problems.

The article highlights the growing interest in Olympiads, indicating that young people are increasingly seeking knowledge, personal growth, and intellectual engagement. Informatics Olympiads are particularly effective in identifying talented young individuals and preparing them for careers in the information technology field. They contribute to the preparation of top-tier staff in this field and aim to introduce students to 21st-century professions.

The article also delves into the intricacies of task evaluation and selection in programming Olympiads, discussing different methods used, their features, implementation, and their respective advantages and disadvantages. It provides a historical overview of programming Olympiads and the evolution of task evaluation and selection methods.

The purpose of this study is to analyze and compare various methods of task evaluation and selection utilized in programming Olympiads. The methodology employed for this investigation includes a comprehensive literature review, a detailed analysis of different task evaluation and selection methods, and an examination of case studies from various programming Olympiads.

The main findings of the study indicate that certain methods of task evaluation and selection prove to be more effective when assessed against defined success criteria. These criteria encompass factors such as fairness, appropriate difficulty level, diversity of tasks, and alignment with educational goals.

The study concludes that effective task selection and evaluation are integral to the success of programming Olympiads. These processes not only ensure the quality and relevance of the tasks but also significantly influence the learning outcomes for the participants. Therefore, continuous research and development in this area are essential to enhance the effectiveness of programming Olympiads as a tool for computer science education.

The article concludes with a discussion of the findings from the case studies, potential improvements to current methods, and future directions for research. It emphasizes the need for continuous research and development in this area to enhance the effectiveness of programming Olympiads as a tool for computer science education.

КЛЮЧОВІ СЛОВА: *Programming Olympiads, Computer Science Education, Informatics, Algorithmic Thinking, Problem-Solving Skills, Task Evaluation, Task Selection, Manual Selection, Algorithmic Selection.*

INTRODUCTION

Problem Statement. In the realm of computer science education, programming Olympiads play a pivotal role. They serve as platforms for enhancing computational thinking and problem-solving skills among students, fostering creativity, logical reasoning, and algorithmic thinking. However, the effectiveness of these competitions largely depends on the quality and fairness of the tasks presented to the participants. The tasks should be fair, challenging, diverse, and aligned with educational goals. Therefore, the methods used for task evaluation and selection are of paramount importance. Despite the evolution of these methods over the years, from manual selection by expert committees to algorithmic selection and crowd-sourced problem creation, each method has its strengths and weaknesses. This raises the question: How can we improve the task evaluation and selection processes in programming Olympiads to ensure fairness, appropriate difficulty level, diversity of tasks, and alignment with educational goals? This study aims to analyze and compare various methods of task evaluation and selection utilized in programming Olympiads to address this question.

Literature Review. The field of computer science education, particularly in the context of programming Olympiads, has seen significant evolution over the years. The methods of task evaluation and selection have evolved from manual selection by

expert committees to algorithmic selection and crowd-sourced problem creation. Each of these methods has its strengths and weaknesses, and their effectiveness varies based on the implementation details and the context of the competition.

The manual selection method, as used by the International Olympiad in Informatics (IOI), allows for expert judgment and careful calibration of task difficulty and diversity. However, it can be time-consuming and may lead to inconsistencies due to the involvement of different committee members. On the other hand, the crowd-sourced problem creation method, as used by Codeforces, generates a large number of diverse tasks and engages the community. Yet, it requires a robust review process to ensure task quality and fairness.

The article also discusses the defined criteria for successful task evaluation and selection, which include fairness, appropriate difficulty level, diversity of tasks, and alignment with educational goals. The success of the methods in terms of these criteria largely depends on the implementation details and the context of the competition.

The article concludes with a discussion of potential improvements to current methods, such as combining features of different methods to mitigate their respective weaknesses. A hybrid method could involve a committee of experts reviewing and refining tasks proposed by the community, combining the expertise of the committee with the diversity and innovation of crowd-sourced tasks.

Future research could explore new methods of task evaluation and selection, leveraging advanced technologies and innovative approaches. It could also investigate the impact of task selection on participant performance and learning outcomes, providing valuable insights for the improvement of programming Olympiads.

Research Objective. The purpose of this study is to analyze and compare various methods of task evaluation and selection utilized in programming Olympiads. The main findings of the study indicate that certain methods of task evaluation and selection prove to be more effective when assessed against defined success criteria. These criteria encompass factors such as fairness, appropriate difficulty level, diversity of tasks, and alignment with educational goals. The study concludes that effective task selection and evaluation are integral to the success of programming Olympiads. These processes not only ensure the quality and relevance of the tasks but also significantly influence the learning outcomes for the participants. Therefore, continuous research and development in this area are essential to enhance the effectiveness of programming Olympiads as a tool for computer science education.

METHODOLOGY

The methodology employed for this investigation includes a comprehensive literature review, a detailed analysis of different task evaluation and selection methods, and an examination of case studies from various programming Olympiads. The article delves into the intricacies of task evaluation and selection in programming Olympiads. It begins with a background section that provides a historical overview of programming Olympiads and the evolution of task evaluation and selection methods. The subsequent section offers a detailed analysis of different methods used for task evaluation and selection, discussing their features, implementation, and their respective advantages and disadvantages. This is followed by a section defining the success criteria for task evaluation and selection, and a presentation of case studies that have used different methods. The article concludes with a discussion of the findings from the case studies, potential improvements to current methods, and future directions for research.

PRESENTATION OF THE MAIN CONTENT

Programming Olympiads play a pivotal role in the realm of computer science education. They serve as a platform for enhancing computational thinking and problem-solving skills among students. These competitions challenge participants to solve complex problems using programming, thereby fostering creativity, logical reasoning, and algorithmic thinking. Moreover, they promote computer science education by providing a competitive and engaging environment for students to apply and extend their classroom learning.

As indicated by the data in [4], there is a growing interest and ambition to succeed in Olympiads. This trend suggests that young people are increasingly seeking knowledge, personal growth, and intellectual engagement.

Informatics olympiads are one of the most effective tools for identifying talented young people and preparing them for careers in the information technology field. They have proven their efficiency in discovering young talents and shaping high-quality specialists in the field of computer technology. Furthermore, the olympiads contribute to the preparation of top-tier staff in this field. The goal of informatics olympiads is to facilitate students' learning and introduce them to 21st-century professions, in a time when a person's professional and social mobility is of paramount importance. [3, p.171-173]

Olympiads contribute to the development of cognitive abilities and creative capacities. It is widely recognized that cognition and knowledge are the engines of progress. A person's cognitive abilities, which are the brain's properties to learn, analyze the environment, and find ways to apply the acquired information in practice, are enhanced through participation in these competitions. Participation in Olympiads and summer computer schools offers teenagers the chance to meet and interact with peers who share similar interests. This social aspect helps to address issues of youth isolation, aloofness, and insularity, as communication and camaraderie are fostered. [4, p. 114-117]

Programming competitions, including the International Olympiad in Informatics and various regional contests for secondary school students, provide an early opportunity for students to begin preparing for a career in this field. [2, p. 171]

Olympiads can take various forms, including individual and team-based, remote and on-site, as well as single-discipline and interdisciplinary. Regardless of their format, they all share numerous benefits, as they provide young people with a wealth of opportunities and play a significant role in fostering effective youth development. [4, p. 112]

While preparing for Olympiads, various online services can be used, for example Combéfis et al. (2013) highlighted, online platforms can also be instrumental in fostering algorithmic thinking through interactive problem-solving exercises, which are particularly useful for learning programming concepts. A case in point is LightBot (https://lightbot.com), an online platform that

introduces the concept of programming, with a special focus on recursion, through a simple game where users solve puzzles. Each puzzle requires the learner to construct a program using visual blocks to direct a robot towards a goal. Another interesting category of online programming learning platforms allows users to create their own games. On these platforms, learners are tasked with programming a game, typically using a visual programming language. Scratch (https://scratch.mit.edu), for instance, uses a visual block programming language to program the behavior of sprites. This category necessitates the learner to have programming capabilities, but not necessarily code-writing skills. The focus here is on fostering creativity and the skills required to design and architect an application. While offering an online platform is advantageous for making learning materials accessible to all learners, relying solely on the online aspect may not be enough to motivate them to learn. As discussed in the second section, the application of gamification strategies can be a potent tool to stimulate learners to make progress, thereby enhancing their engagement and motivation. [6, p. 48-52]

The national olympiads in informatics aim to identify promising students with an interest in informatics, enhance informatics education in schools, and improve students' computer literacy skills. They also promote information technologies, stimulate interest and motivation in this crucial area of scientific and technical progress among the younger generation, encourage students' professional orientation, and foster analytical and creative research, as well as new computational thinking focused on optimal decisions. [3, p. 168-169]

Informatics contests serve not only as a method for testing knowledge and classifying competitors based on their programming and algorithmic thinking skills, but also as a forum for algorithmic thinking. They encourage everyone who is fascinated and passionate about computer science to actively participate in the organization of these competitions, to discover (or rediscover) their own abilities and skills, to rebuild confidence in their intellectual powers, and to develop an 'appetite' for progress in various fields of computer science. [1, p.37]

The importance of programming Olympiads underscores the need for effective task evaluation and selection. The tasks presented in these competitions should be fair, offering an equal opportunity for all participants to demonstrate their skills. They should also be of an appropriate difficulty level, challenging enough to stimulate intellectual growth, yet not so difficult that they become discouraging. Furthermore, the tasks should be diverse, covering a broad range of topics and problem types to ensure a comprehensive assessment of participants' programming skills.

During the preparation of students for the respective competitions and in the subjects proposed for conducting the competition itself, there should be problems that involve the development of mathematical models from the perspective of interdisciplinarity and transdisciplinarity of the exact sciences. [1, p. 38]

Preparation for the informatics olympiads is a multi-year process. It often begins in grades 6-7 with the learning of basic programming and continues with the acquisition of algorithmic methods, typically during grades 7-8. Success at the national and international level depends on these algorithmic methods. Alongside this, students continue to deepen their programming and code-writing skills to facilitate quick and accurate solution creation in contests. [3, p. 171]

This article delves into the intricacies of task evaluation and selection in programming Olympiads. It begins with a background section that provides a historical overview of programming Olympiads and the evolution of task evaluation and selection methods. The subsequent section offers a detailed analysis of different methods used for task evaluation and selection, discussing their features, implementation, and their respective advantages and disadvantages. This is followed by a section defining the success criteria for task evaluation and selection, and a presentation of case studies that have used different methods. The article concludes with a discussion of the findings from the case studies, potential improvements to current methods, and future directions for research.

The history of programming Olympiads dates back to the late 20th century, with the first International Olympiad in Informatics (IOI) held in 1989. Initiated by UNESCO and IFIP, the IOI was designed to stimulate interest in informatics (computing science) and information technology among secondary school students. Since then, the IOI has grown exponentially, with over 80 countries participating annually. The competition has evolved to include a wide range of tasks, from algorithmic problems to software design and development tasks. [5, p. 15-16]

In the [3] article, it is written that the International Olympiad in Informatics (IOI) is the ultimate goal for national team aspirants. Winners and promising students from the final round are enlisted as candidates for the Georgian national team. The top 10 students from the senior and middle age groups, along with the five best students from the youngest group, are selected. The four IOI representatives are chosen from this group after two additional selection rounds. They, along with four reserve competitors, undergo a month-long intensive government-sponsored training period before the IOI. [3, p. 171-172]

The methods of task evaluation and selection for programming Olympiads have seen significant evolution since the inception of these competitions. In the early years, task selection was primarily manual, with a committee of experts designing and selecting problems. As technology advanced and the field of computer science expanded, the methods of task evaluation and selection also evolved. The advent of automated testing systems allowed for more complex tasks and more objective evaluation. The changing educational paradigms, emphasizing problem-solving skills and creativity, also influenced the types of tasks selected.

One of the characteristics of the university Olympiads in informatics, organized within the Tiraspol State University, is about solving problems that, as mentioned above, involve the development of mathematical models. In such situations, students develop their ability to develop mathematical models, and subsequently these models are solved through the studied algorithms. [1, p. 38-39]

The first-ever European Junior Olympiad in Informatics (EJOI 2017) was hosted in Sofia, Bulgaria, spanning from September 7th to September 13th, 2017. The eligibility criterion for EJOI 2017 was that students must have been born post-December 31st, 2001. EJOI, designed along the lines of the International Olympiad in Informatics, is a solo programming competition targeting young coders from countries within the Council of Europe. Each participating country had the opportunity to field a team comprising a maximum of 4 students, chosen based on their national selection procedures. The Olympiad was

structured into 2 competition days, each day presenting 3 tasks to be tackled within a 4-hour timeframe. The programming languages permitted were C/C++ and Java, and the operating systems available were Windows and Linux. The EJOI adopted the CMS grading system. The translation of tasks was facilitated by the IOI's translation system. [2, p.171-172]

The EJOI's official online platform, www.ejoi.org, serves as a comprehensive resource hub. It offers access to a range of materials such as the draft Rules of the Olympiad, hyperlinks to websites of past JBOI events and tournaments featuring tasks that were used (these can serve as a reference for the type of tasks that might be anticipated until an official EJOI Curriculum is established), a roster of countries that have shown interest in participating, among other contest-related information. [2, p.171-172]

The Belgian Olympiad of Informatics (be-OI) incorporates existing online contests into its team selection process for the International Olympiad in Informatics (IOI). The be-OI consists of several stages: multiple local semi-finals, a national final, and an IOI selection process. Specifically, for the IOI selection, a group of top-scoring finalists are invited to join a pool of contestants. From this pool, the four Belgian representatives for the IOI are chosen. Contestants from the pool are encouraged to participate in various selected online programming contests. Their scores are tracked and displayed on an online wiki, allowing the contestants to gauge their performance against others. These scores factor into the IOI team selection process, with higher scores improving a contestant's chances of selection. [7, p. 28-30]

Competition-style programming problems, similar to those used in contests, can be used to promote informatics, as discussed in Voigt et al., 2010. The paper outlines how these problems can be combined with CSUnplugged activities (Bell et al., 2009) to bridge programming and computer science concepts.

In the current state, a variety of methods are used for task evaluation and selection across different countries and competitions. Some Olympiads still rely on expert committees for task selection, while others use algorithmic methods or crowd-sourced problem creation. The evaluation of tasks has also become more sophisticated, with automated systems checking not only the correctness of solutions but also their efficiency. Despite these advancements, the process of task evaluation and selection remains a challenging aspect of organizing programming Olympiads, requiring a careful balance of various factors to ensure a fair and effective competition.

There are several methods used for task evaluation and selection in programming Olympiads, each with its unique features, advantages, and disadvantages [5, p.16-22]:

1.Manual Selection by Committee: This method involves a committee of experts who design and select tasks for the competition. The committee typically consists of experienced educators and professionals in computer science. They create tasks that focus on various aspects of programming, such as problem-solving skills, creativity, and knowledge of specific algorithms. The advantage of this method is that it allows for expert judgment and the inclusion of a wide range of task types. However, it may also introduce bias, as the selection is influenced by the committee members' personal preferences and perspectives.

2.Algorithmic Selection: This method uses algorithms to select tasks based on predefined criteria. The criteria may include factors like task difficulty, diversity, and relevance to the competition's objectives. Algorithmic selection is objective and can handle a large number of tasks efficiently. However, it may not capture all desirable task features, as it relies on quantifiable criteria. It may also overlook innovative or unconventional tasks that do not fit the predefined criteria.

3.Crowd-Sourced Problem Creation: This method involves the competition community in task creation. Participants, alumni, or the broader public can propose tasks, which are then reviewed and selected by a committee or through voting. This method can generate a large number of diverse tasks and engage the community in the competition preparation. However, it requires a robust review process to ensure the quality of the tasks, and it may result in tasks that are too diverse or inconsistent in terms of style and difficulty.

Each of these methods has its place in programming Olympiads, and the choice of method depends on the competition's context and objectives. Some competitions may even use a combination of these methods to leverage their advantages and mitigate their disadvantages.

In [7] article, the experience of the Belgian Olympiad of Informatics is described and the application of the following criteria for competitions is indicated. The first set of criteria pertains to general information about the contest. The first criterion considers whether the contest is for individual contestants or team participation. The second criterion relates to the eligibility conditions that contestants must meet, such as age, gender, or specific academic year requirements. The third criterion concerns the accepted programming languages. The fourth criterion is about the duration of the contest, or the time frame within which contestants can work on tasks and submit their solutions. The fifth criterion pertains to the frequency of the contest. [7, p. 21-23]

The success of task evaluation and selection in programming Olympiads can be assessed based on several criteria, including fairness, appropriate difficulty level, diversity of tasks, and alignment with educational goals.

Fairness: Fairness in task selection ensures that all participants have an equal opportunity to demonstrate their skills and knowledge. Tasks should not favor participants with access to specific resources or advanced training. Fairness also implies that tasks should be free from cultural, linguistic, or other biases that could disadvantage certain participants. A fair task selection contributes to a successful Olympiad by promoting inclusivity and ensuring that the competition results truly reflect the participants' abilities.

Appropriate Difficulty Level: The tasks in a programming Olympiad should be challenging but achievable. They should push the participants to apply and extend their knowledge, stimulating intellectual growth. However, if the tasks are too difficult, they can become discouraging and hinder participation. Therefore, an appropriate difficulty level is crucial for maintaining participant engagement and ensuring a meaningful learning experience.

Diversity of Tasks: A diverse set of tasks allows for a comprehensive assessment of participants' programming skills. It ensures that the competition is not dominated by a single type of problem or a specific area of computer science. Diversity also

makes the competition more interesting and engaging for the participants. A successful Olympiad should include tasks that cover various topics, problem types, and difficulty levels.

Alignment with Educational Goals: The tasks in a programming Olympiad should align with the educational goals of computer science education. They should reinforce the concepts and skills taught in the classroom and promote deeper understanding and application. Alignment with educational goals ensures that the Olympiad contributes to the participants' overall educational journey and not just their performance in the competition.

Each of these criteria plays a vital role in defining the success of task evaluation and selection in programming Olympiads. They ensure that the competition is fair, engaging, educational, and representative of the breadth and depth of computer science.

In this section, we will examine two specific Olympiads that have used different methods of task evaluation and selection, and analyze their success based on the defined criteria.

In order to choose the best algorithm, these algorithms must be analyzed to determine their efficiency and, as far as possible, their optimality. Students are reminded that the efficiency of an algorithm is evaluated from two points of view: from the point of view of the memory space needed to memorize the values of the variables involved in the algorithm; from the point of view of the execution time. [1, p.42-44]

One of the central reasons, for which it is organized separately by years of study, relates to the level of student preparation and the subject matter studied in the various specialized disciplines. In this context students are proposed no less than 8 - 10 problems with different levels of complexity, so that the student can select the subjects on their own taste, from different compartments of the computer sciences, according to their own knowledge and skills, which could lead to solving the problem. [1, p.42-44]

The International Olympiad in Informatics (IOI) is one of the most prestigious computer science competitions for secondary school students worldwide. The tasks for the IOI are selected by a committee of experts, making it an example of manual selection.

The IOI has been successful in ensuring fairness, as the tasks are designed to be culture-free and language-independent. The difficulty level is also carefully calibrated, with tasks ranging from relatively simple to highly complex, allowing students of various skill levels to participate. The tasks are diverse, covering a wide range of topics in algorithms and data structures. They align with educational goals, reinforcing and extending the computer science curriculum.

However, the manual selection method also has its challenges. The task selection process is time-consuming and requires a high level of expertise. There may also be inconsistencies in task style and difficulty due to the involvement of different committee members.

Codeforces is an online platform that hosts competitive programming contests. It uses a crowd-sourced problem creation method, with tasks proposed by the community and reviewed by a team of coordinators.

Codeforces has been successful in generating a large number of diverse tasks. The community involvement ensures a constant supply of fresh and innovative tasks. The tasks also cover a wide range of difficulty levels, catering to programmers of different skill levels.

However, the crowd-sourced method also presents challenges. Ensuring fairness can be difficult, as some tasks may inadvertently favor certain problem-solving approaches or programming languages. The quality and difficulty level of tasks can also vary, requiring a robust review process.

In conclusion, both the manual selection and crowd-sourced problem creation methods have their strengths and weaknesses. Their success in terms of the defined criteria depends largely on the implementation details and the context of the competition.

The case studies of the International Olympiad in Informatics (IOI) and Codeforces provide valuable insights into the effectiveness of different methods of task evaluation and selection. The manual selection method used by the IOI allows for expert judgment and careful calibration of task difficulty and diversity. However, it can be time-consuming and may lead to inconsistencies. On the other hand, the crowd-sourced problem creation method used by Codeforces generates a large number of diverse tasks and engages the community. Yet, it requires a robust review process to ensure task quality and fairness.

These findings suggest potential improvements to the current methods. For instance, combining features of different methods could mitigate their respective weaknesses. A hybrid method could involve a committee of experts reviewing and refining tasks proposed by the community, combining the expertise of the committee with the diversity and innovation of crowd-sourced tasks.

Future research could further investigate the impact of task selection on participant performance and learning outcomes. It could also explore the use of advanced technologies, such as artificial intelligence, in task evaluation and selection. Such research would contribute to the continuous improvement of programming Olympiads and their role in computer science education.

CONCLUSION

In conclusion, the evaluation and selection of tasks play a significant role in shaping the overall competition and are fundamental aspects of programming Olympiads. These processes are responsible for determining the fairness of the competition, ensuring that all participants, regardless of their background or resources, have an equal opportunity to demonstrate their skills and knowledge. They also calibrate the difficulty level of the tasks, aiming to strike a balance between challenge and achievability to stimulate intellectual growth without causing discouragement.

Moreover, task evaluation and selection are responsible for the diversity of tasks in the competition. A diverse set of tasks allows for a comprehensive assessment of participants' programming skills, ensuring that the competition is not dominated by a single type of problem or a specific area of computer science. This diversity also contributes to the engagement and interest of the participants, making the competition a more enriching and enjoyable experience.

Furthermore, task evaluation and selection ensure the alignment of the competition with educational goals. The tasks in a programming Olympiad should reinforce the concepts and skills taught in the classroom, promoting deeper understanding and application. They should also extend beyond the classroom learning, introducing students to new concepts and challenging them to solve real-world problems. This alignment ensures that the Olympiad contributes to the participants' overall educational journey, enhancing their learning outcomes and preparing them for future studies and careers in computer science.

The methods of task evaluation and selection have seen significant evolution since the inception of programming Olympiads. Influenced by technological advancements and changing educational paradigms, these methods have moved from manual selection by expert committees to algorithmic selection and crowd-sourced problem creation. Each method has its strengths and weaknesses, offering unique advantages and facing distinct challenges.

The manual selection by committee allows for expert judgment and a wide range of task types, but it can be time-consuming and may introduce bias. Algorithmic selection is objective and efficient, but it may not capture all desirable task features. Crowd-sourced problem creation generates a large number of diverse tasks and engages the community, but it requires a robust review process to ensure task quality and fairness.

Continuous research and development in this area is essential to enhance the effectiveness of programming Olympiads as a tool for computer science education. Future research could explore new methods of task evaluation and selection, leveraging advanced technologies and innovative approaches. It could also investigate the impact of task selection on participant performance and learning outcomes, providing valuable insights for the improvement of programming Olympiads. Through these efforts, programming Olympiads can continue to evolve and excel, thereby inspiring and educating the next generation of computer scientists.

References

1. Chiriac, L., & Mihalache, L. (2019). The methodology for preparing undergraduate students for Olympiads in informatics in extended format. *Acta et Commentationes, Sciences of Education*, 4 (18), 37-45.
2. Manev, Kr., & Yovcheva, B. (2017). First European Junior Olympiad in Informatics. *Olympiads in Informatics*, 11, 171-173.
3. Mandaria, G. (2013). Olympiads in Informatics: the Georgian Experience. *Olympiads in Informatics*, 7, 168-174.
4. Pavlova, O., & Yanova, E. (2017). Olympiads in Informatics as a Mechanism of Training World-Class Professionals in ICT. *Olympiads in Informatics*, 11, 109-121.
5. Combéfis, S., & Paques, A. (2015). Organising National Olympiads in Informatics:a Review of Selection Processes, Trainings and Promotion Activities. *Olympiads in Informatics*, 9, 15–26.
6. Combéfis, S., Beresnevičius, G., & Dagienė, V. (2016). Learning Programming through Games and Contests: Overview, Characterisation and Discussion. *Olympiads in Informatics*, 10, 39-60.
7. Combéfis, S., & Wautelet, J. (2014). Programming Trainings and Informatics Teaching Through Online Contests. *Olympiads in Informatics,* 8, 21-34.