

**CREATION AND USAGE METHOD OF SOFTWARE TOOLS AIMED AT DEVELOPING
STUDENTS' ALGORITHMIC THINKING IN PROGRAMMING**

Mirsanov Uralboy Muxammadiyevich,

Associate Professor at Navoi State Pedagogical Institute, Doctor of Pedagogical Sciences (DSc)

Abstract. This article highlights the advantages of programming languages and the challenges of developing students' algorithmic thinking related to programming. Additionally, this article presents the method of creating software tools aimed at developing students' algorithmic thinking in programming, along with suggestions and recommendations for their usage.

Keywords: digital technology, algorithmic thinking, modern programming language, Python, C++, software tool.

Programming is one of the essential parts of informatics-related subjects. This is because programming languages are used to solve digital technology problems and to create various modern software products [1, 2]. Therefore, improving the methodology of teaching modern programming languages in general secondary schools, in particular developing students' algorithmic thinking in programming, remains one of today's pressing issues [3].

Within the scope of this research, a “Digital software tool for analyzing program code” aimed at developing students' algorithmic thinking was created. The structural composition of this “Digital software tool for analyzing program code” is illustrated in the following diagram (see Figure 1).

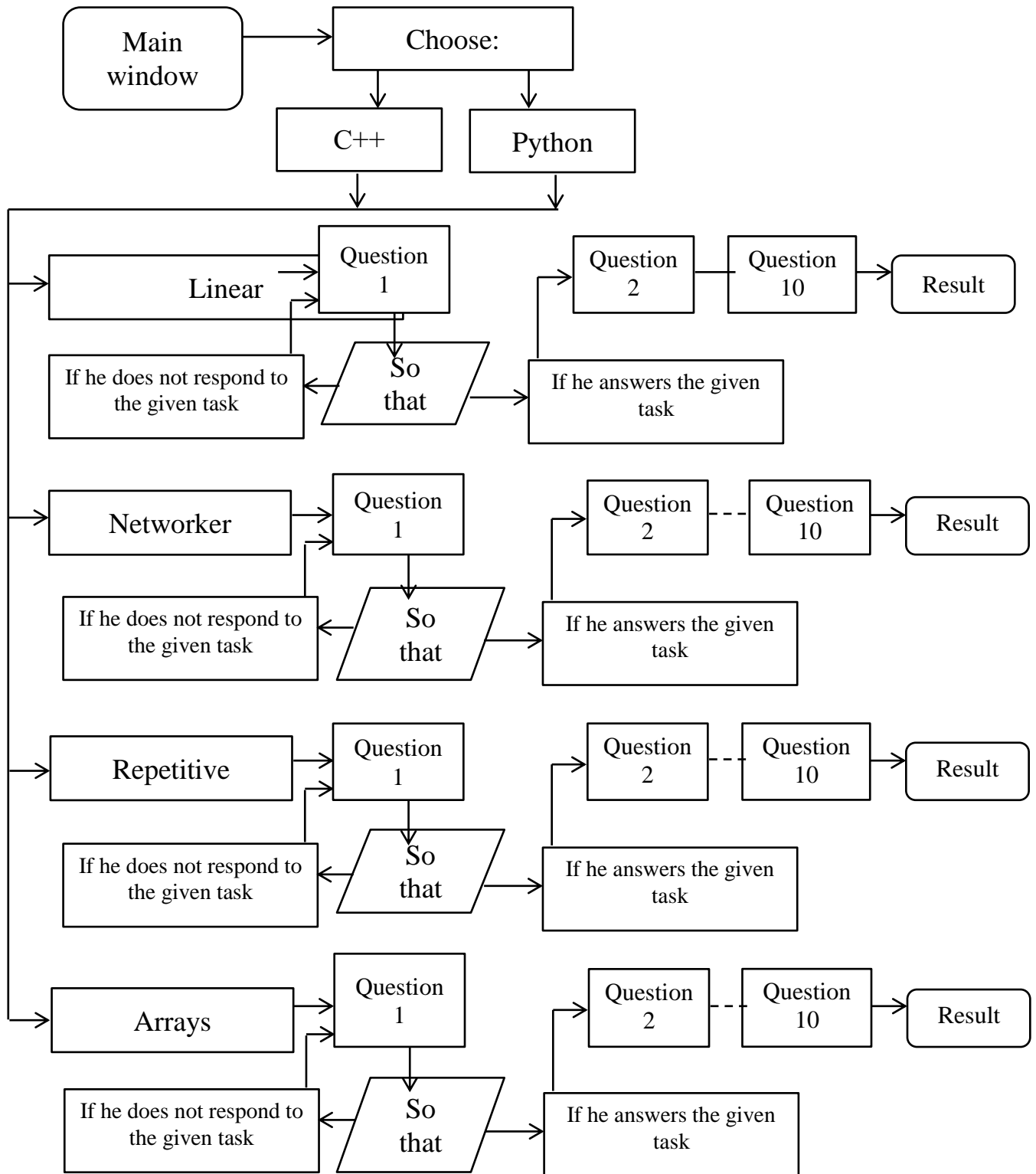


Figure 1. Structural structure of a digital software tool that analyzes program code

This software tool was created based on the principles of programmed educational technology, which aims to determine practical knowledge of Python, C++ programming languages. The main window of this software is shown in Figure 2.

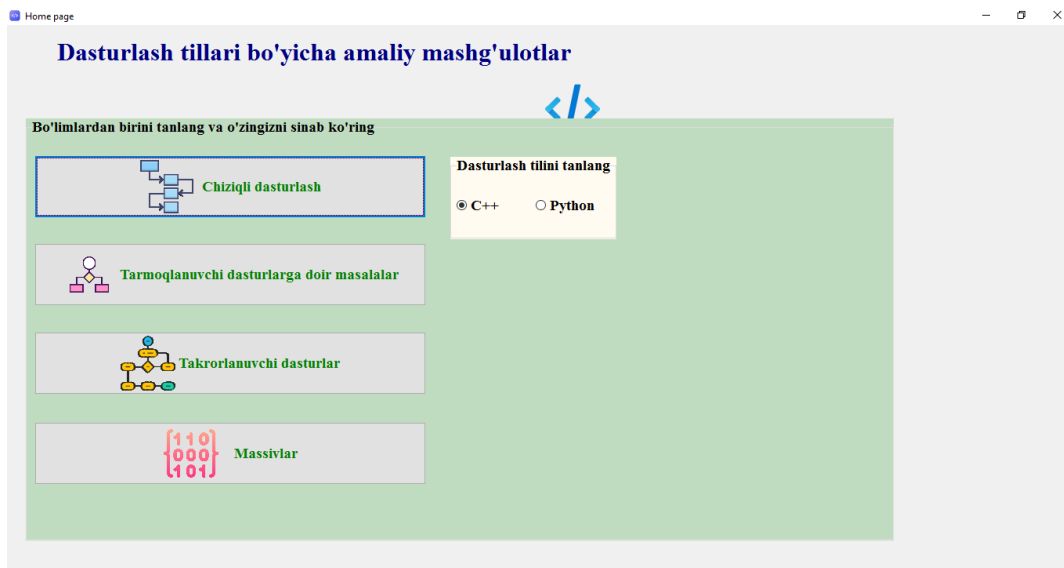


Figure 2. The process of using a digital software tool that analyzes program code

By selecting any Python or S++ item from the main window of this software tool, the necessary sections are passed (linear, branching, iterative, arrays).

After selecting the desired section, two small windows will appear in the program window. At the top of the first window is the condition of the problem. In the lower part, the program code of the problem is given in a confusing form.

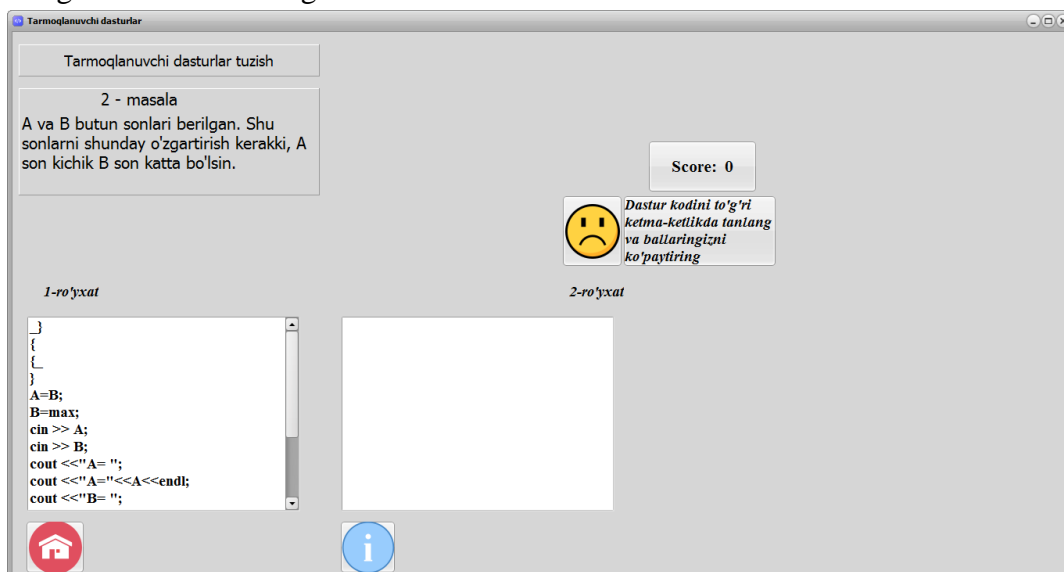


Figure 3. The process of using tasks related to branching processes

In this case, the main task of the user is to transfer to the next one by selecting the given program

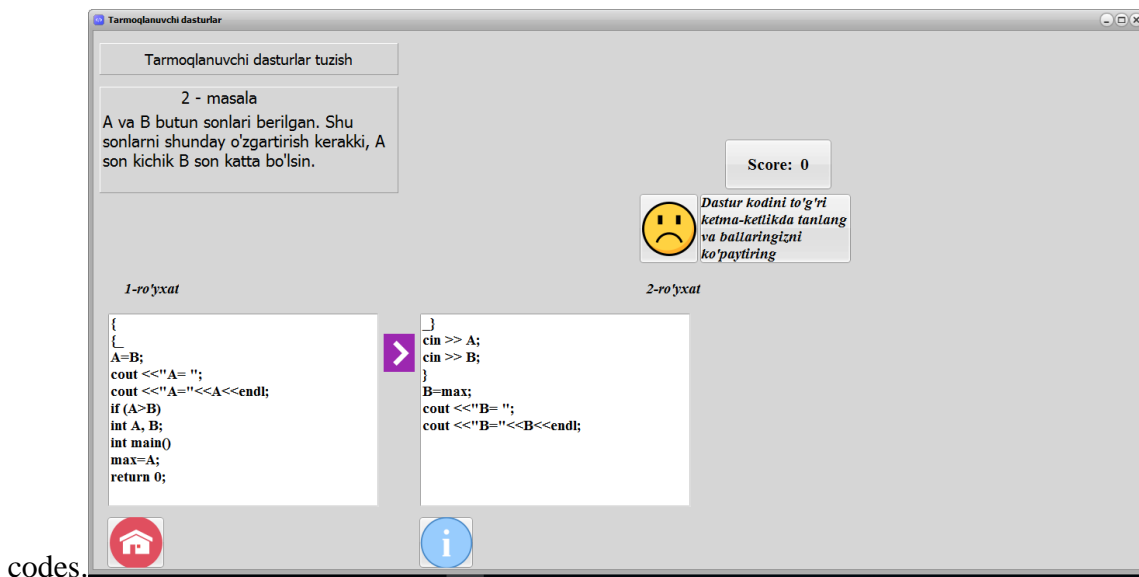


Figure 4. The process of using tasks related to branching processes

Another convenience of this software tool is that other functions are not activated until the program code in the first window is completely transferred to the second window. After the program code is completely transferred to the second window, a test button will appear at the top of the second window.

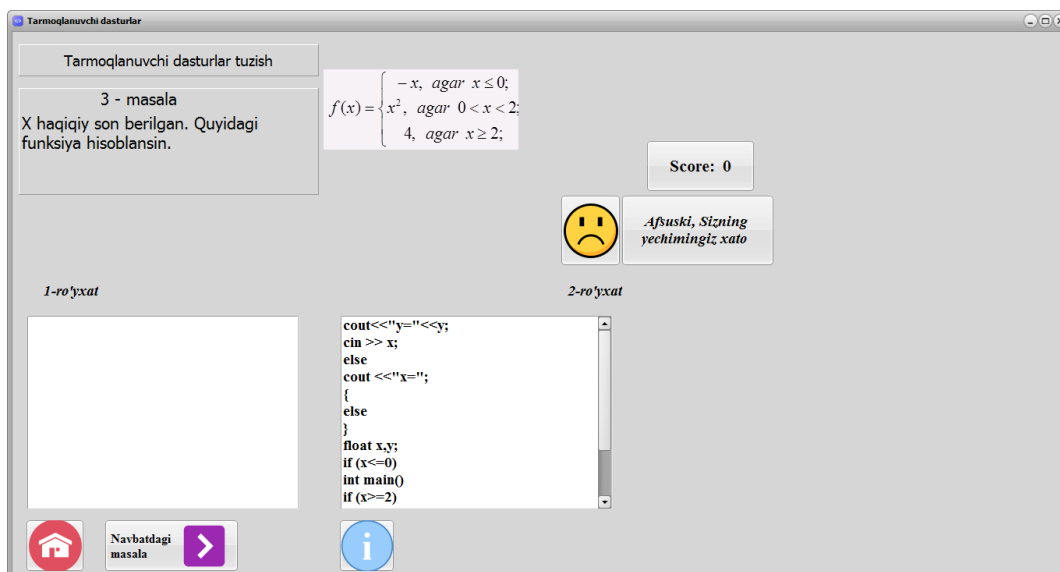


Figure 5. The process of using tasks related to branching processes

By selecting this "Check" button, students will have the opportunity to find out whether the sequence of the program code of the given problem is correctly or incorrectly placed. If the user correctly selects the sequence of the given program code, a value is generated in the "Score" item at the top of the second window. Otherwise, a null value is assigned. Also, next to this button, it will inform you whether you selected the program code sequence correctly or incorrectly. After that, you can go to the next issue by selecting the "Next Issue" button.

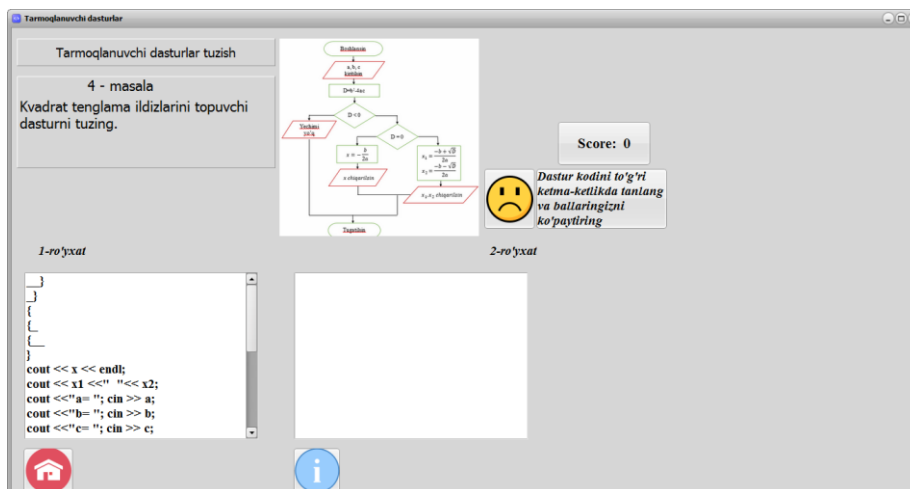


Figure 6. The process of using tasks related to branching processes

By completing the next issues of the window in the order given above, the next task will be passed. In this order, the sum of the points given to the problems, i.e. 0 or 1, is calculated.

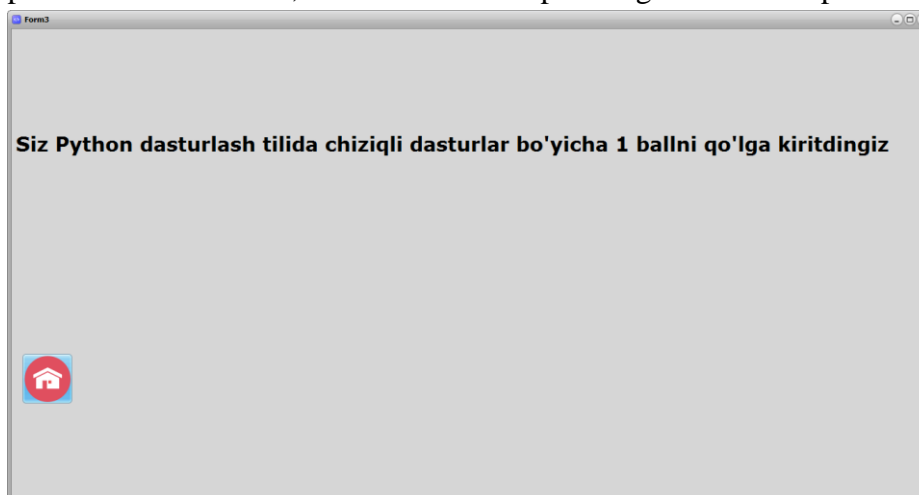


Figure 7. The process of providing user results

After completing the same sequence, the user's score will automatically appear in the window. By using this software tool, it is possible to evaluate students' knowledge of programming and develop their algorithmic thinking.

In conclusion, we recommend using the software developed within the framework of the research to develop the algorithmic thinking of students of the general secondary school regarding programming, including Python and C++ programming languages.

References:

1. Monika Mladenović. Poučavanje početnog programiranja oblikovanjem računalnih igara // Doktorski rad. Prirodoslovno-matematički fakultet. – Split, 2019. –222 p.
2. Гаврилова И.В. Трит-методика решения алгоритмических задач на уроках информатики в основной школе // Автореферат диссертации на соискание ученой степени кандидата педагогических наук. – Красноярск, 2019. – 24 с.
3. Mirsanov U.M. Uzluksiz ta'lim tizimida dasturlash texnologiyalarini o'qitish metodikasini takomillashtirish // Pedagogika fanlari doktori (DSc) ilmiy darajasini olish uchun tayyorlangan dissertatsiya. – Navoiy, 2023. – 332 b.