

METHODS OF TEACHING PROGRAMMING LANGUAGES TO SCHOOLCHILDREN: PROBLEMS AND SOLUTIONS

Zaripov Nozimbek Nayimovich

Doctoral student of Bukhara State Pedagogical Institute (DSc), (PhD), Docent, Bukhara, UZBEKISTAN
e-mail: zaripov9898@mail.ru

ABSTRACT

The relevance of this topic is associated with the lack of quality specialists in the field of IT technologies with the development of computer technologies. In developed countries, digital technologies are primary and we cannot imagine our life without a smartphone, a personal computer or a software that serves in any field. Digital technologies are used in all areas of our society, especially in medicine, health care, information protection, education, filmmaking, designing space models, etc. Therefore, it is important to train quality specialists in this field. In order to train high-quality personnel, teaching programming languages to students through correct, specific and targeted methods is one of the urgent problems of today. This article analyzes the problems of teaching programming languages to schoolchildren and suggests solutions.

Keywords: Computer science, algorithmic thinking, algorithmization, programming, block diagram, Python, Scratch, programming language, mobile application.

INTRODUCTION, LITERATURE REVIEW AND DISCUSSION

In the conditions of the development of digital technologies, it is important to prepare the young generation in the field of informatics and information communication technologies. One of the didactic tasks of the educational institution is the intellectual development of the student, that is an important component of algorithmic thinking. The science of informatics and information technologies has the greatest opportunities for the development of algorithmic thinking in schoolchildren, and is one of the main branches of scientific knowledge that forms the analysis of the world around us, information processes and a systematic-informational approach.

The role of computer education in the development of thinking is largely related to modeling and design techniques, especially modern developments in the field of object-oriented modeling. The ability to define a system of concepts for any field of science, to present them as a set of attributes and actions, to describe algorithms of actions and schemes of logical conclusions, indicates the degree to which a person's thinking in this field of science is developed. For example, in the process of studying the topic "Algorithms and programming", students develop thinking skills such as developing a plan for solving a problem, proposing and proving hypotheses, predicting the results of solutions, analyzing and finding rational methods. The ability to think algorithmically means the ability to solve various problems that require the creation of a plan of action to achieve the desired result.

Development of logical thinking: programming requires an analytical and logical approach to solving problems, which helps to develop the thinking of students.

Future Career Preparation: programming is becoming an increasingly sought-after skill in today's world and knowing the basics of programming will help students in their future careers.

Improve technical literacy: coding teaches not only how to create software, but also to understand the basic principles of technology that are useful in the digital world.

Development of creative thinking: creating programs requires a creative approach and helps to develop creativity in school students.

Real-world problem solving: coding allows students to create real-world projects and solve real-world problems, helping them put their knowledge into practice.

The main problem of teaching programming languages at school is that a clear systematic approach to this issue is not properly organized. The problem is that at school, students learn only the theoretical aspects of the programming language instead of the development of new applications by creating a program. Moreover, the students do not have the skills to work with small projects and solve a specific problem. Few students with programming skills learn it on their own, and many students find the idea of programming to be "boring". Of course, not all students will become programmers in the future, but the skills they acquire in the process of programming will be a good helper in their future life.

In today's information age, informatization covers all areas of human activity, and this phenomenon requires a high information culture from everyone. Therefore, it is necessary to enrich the educational programs of schools with information that ensures the development of knowledge, skills and abilities of students. To do this, teaching students to think creatively and make independent decisions leads to an increase in the quality of education. The solution to this problem naturally depends on the quality of training of pedagogues, the main organizer of this process. The following 3 qualities are required from each teacher entering the class:

- The teacher is knowledgeable, that is, he knows the subject he teaches well;
- The teacher's pedagogical skills, that is, the ability to convey the subject to students;
- The teacher's psychological state, for example, his ability to create a mental atmosphere in the lesson is important.

These 3 qualities serve to create a healthy environment in the classroom. Solving the following problems is considered to be one of the important tasks in making schoolchildren interested in programming languages, and in forming their knowledge, skills, and abilities:

- **Reliance of students on the teacher.** When working with students individually or in groups during the lesson, when they are asked questions, the students remain silent, and even when they know the correct answer, they expect the teacher to give the correct answer. The student knows the right answer, says the answer, but still wants to hear the answer from the teacher. In the course of the lesson, reducing such situations, the teacher should teach students to think independently, express their opinion freely, build self-confidence, and encourage them.
- **Persistent dominance of one or more students.** During the lesson, it is possible to see that excellent students are always active, and unsatisfactory students are not able to fully master the lesson. Such situations should not be allowed in computer classes. It is necessary to create an equal opportunity for all students to actively participate in the lesson and distribute time correctly.
- **Pupils coming to class unprepared.** The first reason for this situation is that the teacher does not demand enough tasks for mastering the lesson, and the second reason is that parents do not devote time to their children at home, and leave the tasks and daily grades unattended. Another reason why students come to class unprepared may be "I don't understand". In order to prevent such situations, the teacher should be able to clearly explain the task, make sure that each student understands the task, and give instructions for the correct execution of the task.

- **Boredom of students.** The traditional way of writing a lesson in a notebook or lecturing can cause students to get bored. Therefore, it is appropriate to use modern information technologies (projector, electronic board, tablet, computer), slideshows made through infographics in the organization of the classroom.

One of the best programs for teaching programming to schoolchildren is the Scratch programming environment. Scratch is a visual programming language developed by the MIT Media Lab to teach computer science and programming to students. This program allows you to create animations, games, cartoons and other interactive projects using blocks. In Scratch, students use blocks to create programs, making the process more visual and understandable.

Scratch is an introduction to programming, preparing students for more advanced programming languages. Scratch is a powerful tool for developing computer literacy and logical thinking, allowing programmers to create fun and educational projects without having to dive into complex syntax.

When teaching programming languages to students at school, it is important to select programs that take into account their youth. Programming languages for elementary school students should be primarily based on visual development. **Scratch Junior** and **Kodu Game Lab** programs provide an opportunity to independently create a virtual world. Students are very interested in this type of work, and the level of mastery of the subject increases significantly. Scratch Junior is a visual code generator. This programming language is the first language that elementary school students should learn. It is a set of blocks that students put together like a Lego constructor. These programs provide an opportunity for schoolchildren to learn the basics of programming during lessons, students independently create animations, cartoons and simple games that include their favorite characters. Kodu Game Lab is a special program for game development from Microsoft. During the learning process, the student acquires the skills of building game logic, learns to create unique 3D worlds, creates signs and chains of their interaction. Kodu Game Lab functionality has a lot in common with the Scratch Junior resource.

For middle age schoolchildren, Minecraft, Python, and Roblox Studio are available. Because not all students of this age know exactly what they want. Therefore, in such a case, the teacher can choose mixed education. Roblox Studio is a program for creating games. This language is an object-oriented programming language, and the work is done using the Lua programming language. In this process, students will have the skills to create their own game world, independently create landscapes, invent characters and objects and add them to the application. Unity is another very popular game development software. Games like Genshin Impact, Subway Surfers, Cities: Skylines, Cuphead, Ori and the Blind Forest, Hollow Knight and others were created using this programming language. Another programming language with many users today is Python. Python is a high-level programming language whose popularity is recognized all over the world. Using the Python programming language, it is possible to create various applications, chatbots, websites, game programs [1, p. 45].

The following principles are important for developing students' intellectual potential for programming:

- Development of the ability to obtain an optimal solution or variable solutions in a difficult situation;
- Development of thinking (visual, algorithmic, theoretical, programmer thinking style);
- Formation of information processing skills (based on the use of databases, data processing systems, as well as information search systems);

- Implementation of software and methodical provision of modern computers, their purpose is transfer of knowledge, simulation of educational situations, implementation of training and control of training results;
- Understanding the use of artificial intelligence systems and their possibilities in the educational process;
- Use of object-oriented software tools and systems (word preparation systems, databases, spreadsheets, and others).

The principles listed above are the basis for students to master the programming environment without difficulty.

It is appropriate to use the following technologies of person-oriented education in the teaching of computer science [3, p. 948]:

- Gaming technologies;
- Problem teaching;
- Programmed training;
- Computerized teaching;
- Algorithmic teaching;
- Modular training.

These technologies are designed to organize stimulating and stimulating activities of students, to develop the skills of working with projects, to quickly join a new team, and to educate spiritual, moral and aesthetic views.

One of the new innovative trends today is the creation of mobile applications. Because at the same time, almost everyone has a mobile phone and several gadgets on it. In the development of mobile applications, students must have a good knowledge of the algorithm and programming language. Writing special programs for phones and smartphones will increase students' interest in programming language. In the development of mobile applications, the student is not limited to the lessons, but tries to create something new on his own. Student-developed apps also motivate other students around them. This, in turn, leads to the formation of modern trends. These trends can be divided into the following areas [7, p. 107]:

- 1) Students should use the project approach in the course of the lesson when developing a software product;
- 2) Use of partially completed programs in order to increase the activity of students in the educational process;
- 3) Use of graphic applications and virtual environments that imply social views of students during the lesson;
- 4) Teaching the Delphi programming environment using the capabilities of open public online courses and distance learning systems (LMS);
- 5) Making students interested in lessons by developing robotics, mobile applications and design elements;

The use of websites, platforms, and multimedia tools is effective in implementing the listed trends. The goal of these is the use of interactive technologies in teaching programming languages. Through this, students will have experience working with various programming methods.

Python is an open source programming language with a simple and concise syntax, excellent libraries, cross-platform and easy to learn. It is perfect for beginner programmers (school students), advanced programmers (students) and professionals (university graduates, IT professionals). Python programming language can be taught at the following levels, taking into account the age of students:

1. Beginner level - learning the basics of circuits and algorithms as a learning tool. Linear, branching and iterative algorithms are also suitable.

2. Intermediate level - in this study the basic theoretical bases of computer science, algorithms, number system, amount of information, logical operations, coding rules.

3. Higher level - working with projects, learning to create programs using procedures, functions, libraries.

CONCLUSION

From the above, we can conclude that it is important to solve the following problems in the training of quality personnel in the field of programming:

- Increasing the number of classes specialized in IT in schools and increasing the hours of computer science;
- Establishing cooperation with IT companies;
- Development of textbooks and educational literature together with higher education institutions;
- Organization of regular training courses for teachers;
- Organization of Olympiads and various competitions among students.

Teaching students programming languages develops logical thinking: helps them develop the ability to analyze, solve problems, and make logical decisions. Today, programming is one of the key skills that opens up many possibilities. Programming allows students to develop their ability to create new programs and applications, develop their imagination and realize their ideas. Learning programming languages helps students understand how different technologies work, which makes them more competent and independent when using computers and the Internet.

REFERENCES

1. Liu et al., 2013 – Liu, A., Newsom, J., Schunn, Ch. and Shoop, R. (2013). Students Learn Programming Faster Through Robotic Simulation. *Techdirections*. Vol. 72, n.8.
2. Golitsyna, O. L. and Popov, I. I. *Basics of Algorithmization and Programming: A Tutorial*. Moscow: Forum: Infra-M, 2004
3. Fernandes, L., & Ferreira, A. (2020). Teaching Python Programming: Gamification, Peer Learning, and Online Tools. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 948-953.
4. Musa et al., 2015 – Musa S., Ziatdinov, R., Sozcu, O. F. and Griffiths, C. (2015). Developing Educational Computer Animation Based on Human Personality Types. *European Journal of Contemporary Education*, Vol. 11, Issue 1, pp. 52-71.
5. Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 1-15.
6. Bosova, L. L., Bosova, A. Yu. *Computer Science. Grades 10–11. Basic Level: A Manual*. Moscow: BINOM. Knowledge Laboratory, 2016. 56 p.
7. Samylkina, N. N., Kalinin, I. A. *Computer Science. Grades 10–11. Advanced Level: A Manual*. Moscow: BINOM. Laboratory of knowledge, 2016. 168 p
8. Rich, K. M., Strickland, C., & Franklin, D. (2017). A Literature Review through the Lens of Computer Science Learning Goals: Theories, Practices, and Principles. *Computer Science Education*, 27(3-4), 182-205.

9. Liu M., Williams D., Pedersen, S. *Alien rescue: A problem-based hypermedia learning environment for middle school science // Journal of Educational Technology Systems. 2002. № 30 (3). P. 255–270.*
10. Werner M. *Teaching graphics programming on mobile devices // Journal of Computing Sciences in Colleges. pp. 2013. P. 125–131.*