Proceedings of International Scientific-Practical Conference «Sustainable Development Goals: Youth Policy and Innovative Technologies», February 15-16, 2023, Oguz Han Engineering and Technology University of Turkmenistan, Ashgabat, Turkmenistan

METHODS OF TEACHING PROGRAMMING

Tachjemal Gurbanova*

Oguz Han Engineering and Technology University of Turkmenistan, Ashgabat, Turkmenistan.

*Corresponding author

DoI: https://doi.org/10.5281/zenodo.7783019

The current stage of the development of society is characterized by the integration of information technologies in all spheres of human activity, so we must not only be able to use them for their intended purpose, but also understand how they are created and possibly even create more practical, more useful and, most importantly, safe for human health.

Computer science, despite the fact that it is the youngest of all sciences, is rapidly developing and popular today. And one of the areas of this science is programming, respectively, in a developing society, the question arises about its study and teaching methods [1].

And if we talk about learning, questions arise at “What age you need to start learning?”, “Where to start?” and “Whether you need to study programming at all, if in future you do not choose a profession related to programming, and indeed to computer engineering?”.

As for the issue of teaching methods, computer science is a very young science, unlike other sciences, so there are not so many scientific works on the topic of teaching methods. But based on them, as well as on my small teaching experience, I would like to note that the methodology should be compiled based on the age, knowledge base and individual abilities of the student [2].

And if you follow the line "Algorithms and executors", which is a necessary basis for the assimilation of students, both the basics of programming and the technological components of computer science, then the programming methodology consists of:

1. Focus on practice as much as possible, but we must not forget about theory, because in order to find something, you need to know what to look for. And if the developer does not know the theory, then most often he does not even suspect the existence of the knowledge necessary to effectively solve a non-trivial problem. Programming is not only the ability to write code, but also the ability to choose the best one for given specific conditions from dozens of different algorithms for solving a problem. And the theory is needed in order to:

• Learn to see diversity algorithms to solve problem.

• Learn to evaluate the optimality of each of these algorithms under given conditions.

Excluding the theory, you get regular coders mindlessly copying parts of the code and not able to evaluate the effectiveness of the decisions made [3].

2. In order for the lessons to be visual, the practice should be compiled on the basis of real problems that programmers constantly face.

3. Students should share their solutions to the problem and discuss them. Students learn theory from different sources - this leads to the fact that the solutions to problems are different for everyone. Project defense is part of the learning process and is based on checking each other's tasks. Thus, students understand the difference between different solutions. They don’t just copy what was shown at the lecture, but they see the difference by comparing it with their own solution. This much faster leads to the realization that there is no one correct solution for the problem, but there are a lot of them, they are different and each is correct in its own way.

4. Another effective way to teach is to write code in front of students, since the teacher's question is “what, if?” and predicting the student's response will lead to the correct mastering of the material.

5. The mentor should divide the students into groups (2-5 people). Partners can not only help each other during practical exercises, but also explain the proposed solutions [4].

In conclusion, if you want to teach someone how to code, you can't just talk and let them get the feeling that they can't succeed (if they didn't start at a young age) or that any skill they have (whether when and how it was acquired) is useless. Let your students feel they can be programmers, and they may be very well.

REFERENCES

[1]. Ю. А. Аляев, Методика быстрого обучения программированию на основе изучения классов задач. Образовательные ресурсы и технологии, 1(9), с. 3-14, 2015 г. (in Russian).
[2]. Porter L, Guzdial M, McDowell C, Simon B. Success in introductory programming: What works? Communications of the ACM. 56 (8), 2013
[3]. Council NR. Reaching Students: What Research Says About Effective Instruction in Undergraduate Science and Engineering. Washington, DC: The National Academies Press; 2015
[4]. M.J. Rubin The Effectiveness of Live-coding to Teach Introductory Programming. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education. SIGCSE '13. New York, NY, USA: ACM; 2013. pp. 651–656