International Journal of Social Science Research and Review

http://ijssrr.com
editor@ijssrr.com
Volume 5, Issue 6
June, 2022
Pages: 229-234

Analysis of the Problems of the Teaching Object-Oriented Programming to Students

Dilfuza Ergashovna Saidova

Department of Algorithms and Programming Technologies, Karshi State University, Karshi, Uzbekistan

Email: dilfuza.e.s.89@mail.ru

http://dx.doi.org/10.47814/ijssrr.v5i6.418

Abstract

In modern programming, the object-oriented approach is one of the most popular. Object-oriented programming has already become an integral part of university curricula. Today, the university faces a new urgent task - to prepare the future teacher of computer science for teaching this topic, to equip him with sufficient subject and methodological tools. This article provides an analysis of the problems of teaching object-oriented programming to students.

Keywords: *Object-Oriented Programming; Learning Problems; Computer Science Teacher Training; Object-Oriented Paradigm; Object-Oriented Concepts; OOP Languages; Programming Science*

Introduction

In modern programming, an object-oriented approach is widely used, which allows you to improve the quality of programs, the productivity of the programmer, and the efficiency of teamwork. In the process of teaching object-oriented programming (OOP), one has to face some difficulties: understanding the basic concepts and choosing a programming language. [1].

Really, the object-oriented style is used in the development of a wide range of applications. The ability of students to think objectively is formed when developing visual applications using standard objects (components) of the OOP system.

When studying object-oriented programming, students are encouraged to develop visual projects in specific programming languages. Students need to be trained to apply knowledge in real situations, to expand the scope of possible application of OOP. To do this, it is recommended to solve problems that have objects whose prototypes are real-life natural objects and structures.

Before OOP, a different approach was used in development - procedural. The program is represented in it as a set of procedures and functions - subroutines that execute a specific block of code with the necessary input data. Procedural programming is well suited for lightweight programs without complex structure. But if the code blocks are large, and there are hundreds of functions, you will have to edit each of them, think over new logic. Unlike procedural programming, object-oriented programming allows you to make changes once - to the object. It is he who is the key element of the program. All operations are represented as an interaction between objects. At the same time, the code is more readable and understandable, the program is easier to scale.

We know that the transition from the old imperative procedural style of programming to the object-oriented style is still a difficult task. Because object orientation requires a paradigm shift for students. It is known that learning object-oriented programming is more difficult after procedural programming experience. It has been observed that it takes a certain amount of time for the average student to switch from a procedural to an object-oriented approach.

Significance of The System

The paper mainly focuses on analysis object-oriented programming teaching problems, and object-oriented programming teaching practice and organization are described. The study of literature survey is presented in section III, Methodology is explained in section IV, section V covers the experimental results of the study, and section VI discusses the future study and Conclusion.

Literature Survey

The article [1] discusses the issues of teaching the methodology of object-oriented design. Two types of problems are identified that one has to face in the process of teaching object-oriented programming (understanding the basic concepts, choosing a programming language).

In literature [2] object-oriented programming teaching problems are formulated. Object-oriented programming teaching practice and organization are described. The article [3] outlines the main approaches to teaching students object-oriented programming and design. The author highlights the contradictions associated with the teaching of OOP in higher education and gives recommendations on object-oriented design.

The dissertation [4] presents the main results of the study:

1) the role and place of teaching programming in the system of training bachelors of applied informatics are analysed. The content of training in object-oriented programming has been specified;

2) the competence of the students in the field of object-oriented programming is described (definition, components, levels);

2) developed a methodology for teaching the basics of object-oriented programming using visual learning environments;

4) experimental work was carried out to evaluate the effectiveness of the methodology for teaching the basics of object-oriented programming to students.

Methodology

In the process of teaching object-oriented programming (OOP), traditional forms of education are mainly used, such as lectures and seminars, which have proven themselves well in the process of teaching structured programming, but in modern conditions are clearly insufficient for studying such a complex programming methodology as OOP.

Students' practice of OOP principles outside the context of the benefits that this programming methodology offers can lead to formalism in their learning process. Students will have the necessary theoretical knowledge about OOP, but will not be able to effectively apply this knowledge in the implementation of complex projects in practice. Therefore, in the process of teaching object-oriented programming, it is necessary to pay special attention to students not only to consider the advantages of OOP, but also to learn how to implement these advantages in practice.

Professor I.A.Barkov listed the main problems that arise in the educational process for mastering object-oriented programming, and we consider such an analysis to be correct [2]:

First, object-oriented programming is focused primarily on the creation of complex programs, and the student, performing laboratory work, solves simple problems. In term papers, the student most often solves local problems. The project manager is the teacher, he is responsible for the project as a whole. As a result, most students do not gain experience in creating complex programs and, moreover, in managing the process of their development.

Secondly, object-oriented programming creates good opportunities for modifying programs, but these opportunities are laid down at the stage of program design. The student is not motivated to ensure the modifiability of the program.

Third, most of the programs that serve as examples of successful programs are designed using structural design technology. Therefore, the teaching of object-oriented programming is influenced by the ideas of structured program design.

Fourthly, the science of programming is developing rapidly, therefore, the knowledge of programming techniques usually obtained at the initial courses by the time of graduation from the university is largely outdated.

Fifth, the complexity of automation tasks is steadily increasing. Therefore, a young specialist, having received an idea about one level of complexity of software systems, falls into the conditions of the next level of complexity. Therefore, the stage of adaptation to new conditions is inevitable. From the graduate, the qualities of prompt response to changing working conditions are again required.

Sixth, a high-quality program should not only solve an applied problem, but also be easy to read and understand. The use of good identifier naming conventions and similar practices greatly enhances continuity. Therefore, the availability of program understanding is, first of all, a practical principle: as scientific studies show, when debugging programs, up to 90% of the time is spent reading the source text.

Seventh, the peculiarity of the organization of the educational process in a technical university is that the student must study a large number of general educational, general technical and special disciplines. The differentiation of sciences has reached a large scale and has not passed by object-oriented programming, which is based on the theory of algorithms, the theory of abstract data types, the theory of program design, etc. As a rule, the development of different disciplines is carried out under the guidance of different teachers, as a result of which students have difficulties in understanding the integrity of the educational material distributed over different disciplines.

Eighth, the teacher faces the problem of assessing the knowledge and skills of students in the field of object-oriented programming, which largely affects the interests of the student. Given that the technology under discussion is focused on creating complex programs, the teacher has to overcome serious difficulties in choosing adequate controls or simplifying the picture. Knowledge control at the syntax level of programming language constructs can hardly be called satisfactory for object-oriented programming.

Ninth, the creators of software, as a rule, consider the implementation of programs a trade secret, only the external logic of the program is disclosed in detail. Therefore, there are virtually no good examples of the implementation of object-oriented programs with comments on the choice of solutions. For this reason, when offering examples of programs to students, the teacher acts subjectively, and students are dependent on the professional preferences of a particular teacher.

Finally, there are traditional problems for the educational process. Firstly, the student body is heterogeneous: some students came to get a profession, others – a diploma, others – on the recommendation, and not to their liking. Recently, a specific student has appeared – a non-state employee. The teacher faces the problem of choice: who to focus on? Secondly, practical studies are ahead of theoretical studies. The teacher must organize the independent work of students on the preliminary mastery of theoretical knowledge. Thirdly, the sequence of disciplines in the educational process often does not meet the requirements of primary-secondary knowledge acquisition. And the reason here is not in the poor organization of the educational process, but in the strong interconnection of many disciplines within the same specialty. In the latter case, the teacher often has to be distracted by brief excursions into other disciplines.

In the process of teaching students object-oriented programming, the following educational tasks should be solved [3]:

- formation of ideas about the OOP;
- training in the principles of OOP;
- training in object-oriented design;
- teaching object decomposition;
- learning to create reusable program code.

To solve the set educational tasks, it is necessary to use not only such organizational forms of training as individual and group, but also a collective form of organizing the learning process, which is most appropriate in working on projects.

In the practice of teaching OOP in higher education, the following contradictions are observed:

- the methodology of object-oriented programming is superficially presented, while it is one of the main ones;
- students study predominantly algorithmic decomposition, and object decomposition is considered insufficiently;
- students do not gain experience in using object-oriented design tools, which is so necessary for them in their future work.

Thus, it is necessary to pay attention to the formation of students' ideas about OOP and object-oriented design. In the process of learning OOP, it is recommended to use object-oriented design tools (for example, the UML language).

The problem of improving the process of teaching students in the field of object-oriented programming requires, first of all, the specification of the content of such training, which is realized in the dissertation [4] through the description of the corresponding competence.

The competence of students in the field of object-oriented programming is a set of personal qualities, knowledge and skills in this area, providing the ability to develop software using modern programming languages and related tools, readiness to master the profession of a programmer, perceived as personally significant.

Experimental Results

Experiments with students have shown that there are several principles based on teaching object-oriented programming, and these include:

1) the correct choice of OOP languages is at the discretion of the students (Table 1);

Table 1. The overall result of a survey of students on the choice of an OOP language

Languages	Quant. (%)	Languages	Quant. (%)	Languages	Quant. (%)	Languages	Quant. (%)
Python	40	Smalltalk	17	VB Net	54	Python	63
Eiffel	40	C#	47	JavaScript	33	VB Net	22
C++	20	Delphi	36	Object Pascal	13	C#	15

2) it is important to start teaching OOP in the early lessons;

3) from the outset, students must work in a powerful software environment with many classes, creating effective GUI applications from ready-made components;

4) to work in such an environment, knowledge of interfaces built on contracts is sufficient. From the very beginning, students develop an understanding of the importance of the specification of the software being developed. The code of all software provided to students is open, which allows you to move at the right stage from understanding the interface to understanding the implementation.

Conclusion and Future Work

OOP is used to: structure information and avoid confusion; accurately determine the interaction of some elements with others; improve program manageability; scale code faster for different tasks; better understand what is written; support ready-made programs more effectively; implement changes without having to rewrite all the code, etc.

OOP features support most popular programming languages, including JavaScript, PHP, Python, Eiffel, Smalltalk, Object Pascal, C++, C#, VB NET, and more. However, since most students choose Python, in our next study we will look at OOP teaching methods in a Python environment.

References

1. Seidametova Z.S., Abduramanov Z.Sh., Asanova U.B. Pedagogical features of the formation of object thinking and programming style // Crimean Scientific Bulletin, No. 4, 2015. p.10-22.

2. Barkov I.A. Object-oriented programming: textbook. – St. Petersburg: Lan, 2019. – p.700

3. Petrov A.N. Basic approaches to teaching students about object-oriented programming and design // Journal “Fundamental Research”, No. 4, 2008. p.80-82.

4. Shkarban F.V. Methodology for teaching the basics of object-oriented programming for bachelors of applied informatics using visual learning environments. Abstract of the dissertation for the degree of candidate of pedagogical sciences in the specialty 13.00.02 - theory and methodology of training and education (computer science). Volgograd - 2018.

5. Kölling, M., The Problem of Teaching Object-Oriented Programming, Part 1: Languages, Journal of Object-Oriented Programming, 11(8): 8-15, 1999.

Copyrights