Aliaksandr DZIVINETS
Dzmitry KOSTIUK
Anastasia MARKINA
Vita RAZUMEICHIK
Brest State Technical University,
Brest, Republic of Belarus

ON MODERNIZING THE APPROACH TO TEACH THE ALGORITHMS AND PROGRAMMING BASICS

Introduction

The Computers & Systems Department of Brest Technical University provides higher education to students of the first and second stages, and one of the main special disciplines taught at the first stage is the "Basics of Algorithmization and Programming" (OAiP) subject. The goal of teaching this discipline is to master the methodology for setting, preparing and solving engineering problems on modern computers. The tasks are the study of a procedural programming language (currently, the C language is rigidly fixed in the basic curricula for all specialties) and automated development tools (IDE), laying the foundations of a programming style, acquiring development and debugging skills, and developing an algorithmic thinking. Additionally, the first-year students acquire skills of self-control, gain their interest in programming, and create the motivation for self-development.

The traditional problems that are observed are the low activity of students in terms of their participation in the educational process, adaptation to the requirements of the university after school, as well as

a low level of algorithmic thinking and low skills in program development. If these problems are not solved well enough in the first year, this has a strong negative effect on subsequent disciplines. Moreover, the relevance of effectively dealing with these problems within the discipline had predictably intensified when the division of higher education into first and second stages led to a decrease in the number of teaching hours (and laboratory classes were significantly affected).

In the process of trying to solve these problems, methodological changes were made to the course, which will be discussed below.

The technologies of teaching

The following methodological changes were made to the course in attempts to solve these problems.

A visualization-type lecture allows one to schematically and graphically present the needed supporting information, which is supplemented by video material. This type of lecture is the most important one for visual learner students, because the maximum assimilation of the material occurs with it. In addition, it increases knowledge in the field of algorithms, develops systems thinking, allows students to get clarity in the design and optimization of the code. The approach is applicable to all lecture topics.

A provocation-type lecture. This type of lecture allows one to control the awareness of the perceived material by the listeners, develops their critical thinking, increases the spirit of competition and satisfaction from the educational process. At the beginning, the lecturer announces how many mistakes will be made in the lecture. At the end, the mistakes found by the students are checked and discussed. We have used this type of lecture for the following subjects: "Loops", "Arrays", "Recursion", "Lists", and "Binary Search Trees".

Conference-type lecture. It is built in the form of a scientific and practical lesson with reports and speeches made by students (in small groups) on a previously prepared problem within the framework of the

curriculum. The lecturer sums up the results, clarifies information, formulates the main conclusions. This type of lecture allows us to develop communication skills, healthy competition, the level of criticality and satisfaction from the process. It is good to be used for final lectures which complete individual sections of the course.

Problem-type lecture is carried out in the form of a dialogue with students, during which a joint search for a solution to the problem takes place, and typical mistakes in work are analyzed. This approach allows one to develop the communication skills of young people, and also leads to an increased awareness of algorithms and the basics of procedural programming. We used this for lectures after the intermediate control.

In addition, during the lectures, there is a combination of listening and checking the knowledge of students (2 times per lecture), which makes it possible to better retain the attention of the audience.

In the course of laboratory studies, we use:

The "discussion" method, which is used during the discussion of the issued individual tasks. She maximally activates the thinking activity of students and increases the assimilation of theoretical conclusions. In addition, this method increases the motivation for learning.

The brainstorming method is aimed at reducing students' self-criticism, increasing self-confidence and developing the skill of a creative approach to solving problems. It is used in laboratory tasks on the following topics: "Loops", "Arrays", "Recursion", "Stack", and "Binary Search Trees".

We also use heuristic conversation when students defend laboratory work. This active teaching method is based on problem-oriented questions that the student provides answers to. As a result, thinking is activated and new knowledge is acquired.

It should be noted that heuristic conversation proved to be the least effective among those listed: in a number of cases, students got lost in the process of trying to answer (perhaps there is an external reason for this — a programming language that was not originally created for teaching students). To a lesser extent, problems arose with the brain-

storming method among some of the students who had difficulties with improvisation and free expression of their thoughts.

In the course there are forms of current (twice a semester) and final assessment. To increase interest and motivation for educational activities, we use a credit rating system of assessment during the academic semester: students get acquainted with the list of compulsory works, the requirements of teachers, activity and attendance are recorded. At the end of the semester, the rating of students is built according to the results of their work, and then it is taken into account in the final control.

Used software tools

The cross-platform nature of the C language allows students to use any OS and development environment on their own devices. However, Ubuntu Linux and the Qt Creator IDE are installed as the main systems in the classrooms. The advantage of using these instrumental environments is the initial formation of students' habit of the variety of graphical shells and interfaces. In addition, elements of command line and environment skills make it easier to study a number of courses in the next years of study, including such as "Program Design and Programming Languages" (the course includes creating graphical applications in Qt Creator), "Computer Architecture" (within which skills in writing elementary Linux kernel modules are acquired), etc.

Conclusion

Although the measures used are not a panacea, the past two years of study show positive dynamics in mastering the knowledge of students, interest in future activities increases, motivation to study increases, and the adaptation period of study is more easily tolerated.