Педагогическо списание на
Великотърновския университет
„Св. св. Кирил и Методий“

ПЕДАГОГИЧЕСКИ АЛМАНАХ

Брой 2, 2021

DOI: 10.54664/AIFZ2054

МАТЕМАТИЧЕСКО ОБУЧЕНИЕ

MATHEMATICAL TRAINING

PROJECTS IN COMPUTER SCIENCE TRAINING – AN EXAMPLE WITH NUMERICAL SERIES

Ivaylo Donchev *, Gabriela Chotova**

Abstract: *When it comes to teaching computer science, working on software projects is a quite natural pedagogical approach. Its main advantage is the increased motivation for learning, which arises from the opportunity to immediately see the application of acquired knowledge. In this paper, the authors share their experience of creating a project in computer science for high school students, making a cross-curricular connection with mathematics. Emphasis is placed on interactive teaching methods in the classroom.*

Keywords: *teaching; programming; project-based; development environment.*

INTRODUCTION

Working on real projects has a highly motivating effect on students, and this has been proven many times in our practice. There are many reasons for that:

- Students gain autonomy which provides them with a sense of independence, ownership and self-worth. Autonomy of work is a greater motivator than rewards (grades) or punishments (Pink 2011: 22);
- Classrooms become collaborative communities where students do interesting work with supportive colleagues;
- Instructors provide strategic positive encouragement and feedback;
- Students are motivated by challenges and gain flexibility and initiative;
- They acquire knowledge and skills in how to deal with and learn from their failures.

One of the projects we developed for the school course is related to arithmetic progression. The idea arose in the process of work. Once in computer science class, students were curious about how to

* **Ivaylo Donchev** – Associate Professor, PhD. Lecturer at the Department of Information Technology, Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of of Veliko Tarnovo, Bulgaria, Email: i.donchev@ts.uni-vt.bg

** **Gabriela Chotova –** PhD student in Informatics and IT Teaching Methodology, Department of Information Technology, Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria, Email: gabriela.chotova@gmail.com

implement arithmetic progression in a programming language. However, the work on the project requires the acquisition of a lot of knowledge of the important concepts and mechanisms in the programming and design of software applications, as well as development of skills for effective use of a modern development environment and modern programming language. That is why we rely on a spiral approach with gradual upgrading of functionalities and parallel work on several projects.

The benefit of building interdisciplinary connections is indisputable and clarified in the works of the classics Comenius, Pestalozzi, Disterweg and Ushinsky. These connections help both to achieve lasting knowledge in related subjects and to acquire practical skills for dealing with difficult real problems. The practical realization of these connections requires a lot of effort from teachers who must first identify and then show students the similarities or dependencies of one subject with another in an understandable way.

Education is a coordinated process, and its main goal is to combine knowledge from different subject areas. The education system should help learners to acquire a reasonable level of knowledge and skills for widely applicable approaches to formulate and solve problems. Various studies conducted in the last few years show that students are often unable to apply their mathematical knowledge outside the classroom, as well as to link this knowledge and skills with the curriculum in other disciplines (Darling-Hammond et al. 2020). Learning outcomes are usually assessed on the achievement of students rather than on the didactic goals achieved by teachers – they are usually related to the expectations of the learner for his/her knowledge, understanding and skills at the completion of a level or module of education (Chotova 2019).

METHODOLOGY

Pedagogical practice shows that the effectiveness of teaching strongly depends on whether students actively participate in the learning process and see the results of their work directly or not (Nikolova 2016: p. 105). It is essential for them to realize in the foreseeable future the profit of knowledge they were forced to acquire. This can be achieved by considering real cases in class and solving real (with practical applicability) problems i.e. problem-based learning. It is a suitable environment for collaborative learning. The aim is to build skills for applying what has been learned in typical and specific situations. A key skill is the adaptation of students to work with information resources. They must turn the information obtained into an application. An important element in the process is the student-student interaction, which leads to the development of teamwork skills. This method yields best results when working in a team because thinking and exchanging ideas and experience in collaboration ensure that even the most complex problem can be solved (Nikolova 2016: 83).

Students learn effectively through research. “Training is not a sports competition in which students are spectators. They should not just stand in class, listen to the teacher, memorize his/her words, and answer his/her questions. They need to discuss what they are learning, to connect it with their personal experience, and to apply it in their daily life. In doing so, they study a part of themselves” (Ivanov 2005).

The approach we propose in this article is a variant of the objects-first strategy (The Joint Task Force on Computing Curricula 2001: 34). Objects are introduced early (objects-early), but without underestimating the procedural language constructions. Our practice shows that in computer science education, it is best to combine ideas from several approaches. Our option relies on additional motivation of students through:

- Use of a modern development environment (current version of Eclipse, NetBeans or Visual Studio) and building applications with graphical interface;
- Management of physical systems.

It is essential for contemporary students to see the benefits of the material studied. So, from the very beginning, before moving on to writing code, in the form of a demonstration, they get a clear idea of why they need to learn all this. Demonstrations include software for library, hotel room reservation or facility management in the Scratch (visual programming language) environment by applying various commands.

After the demonstration of different types of ready-made software and the use of programmable, interactive resources, the lessons follow the topics provided in the curriculum.

An important part of our approach are the examples and tasks that develop students' skills and consolidate their knowledge. After the introduction of each of the concepts set in the curriculum, it is necessary to operate with them. The tasks we prefer are of an applied nature: creation of different types of calculators – graphical, geometric, calculation of body mass index; car rental application; game of sea chess and others. Our experience shows that the creation of games has a strong positive impact (Donchev, Chotova 2019).

Constructions in programming languages are built on the basis of formal mathematical logic, which in turn can be described by program code. Programming languages are very similar to mathematical language. Students learn to read, understand, and formulate their actions through the given programming language. There is a lot of talk in literature about computational thinking (Wing 2006). It is related to the formulation of problems and their solutions. The term "computational thinking" was first used by Wing and refers to the set of concepts and strategies (abstraction, recursion, iteration) used by computer specialists and scientists for formulating and solving problems. Wing determines computational thinking as a new thinking paradigm, allowing innovative approaches to dealing with problems from everyday life and from many fields of science. We can summarize that computational thinking skills prepare students to make real decisions regarding problems outside the classroom. Consequently, the basis of informatics is not only writing code (Chotova 2019).

RESULTS AND DISCUSSION
PROGRAM SOLVING OF ARITHMETIC PROGRESSION

Arithmetic progression is studied in Grade 10 in mathematics. In the computer science course, it can be presented by studying a cyclic algorithm that is embedded in Grade 8.

The lesson of arithmetic progression begins with defining the concept of sequence. In Figures 1–3, sequences are presented, and the dependence of each row is discussed.

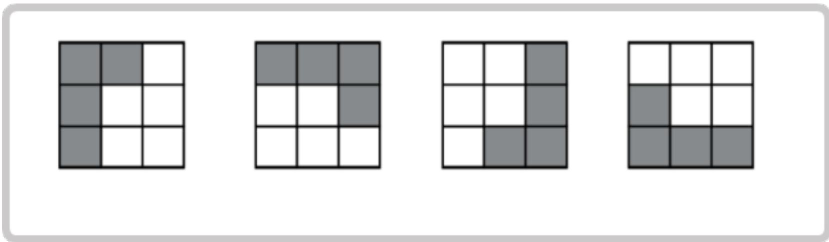

Figure 1. Sequence

In the second row, the block of three black tiles rotates sequentially at the vertices of the square.
To get the next house, how many sticks do I have to add? With the next one?

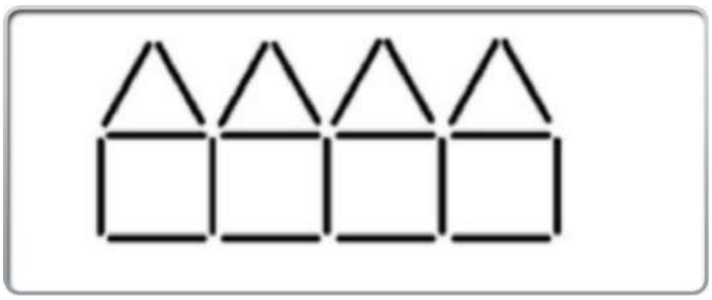

Figure 2. Sequence

In this row, at each step, the polygon loses one side, leaving only a segment.

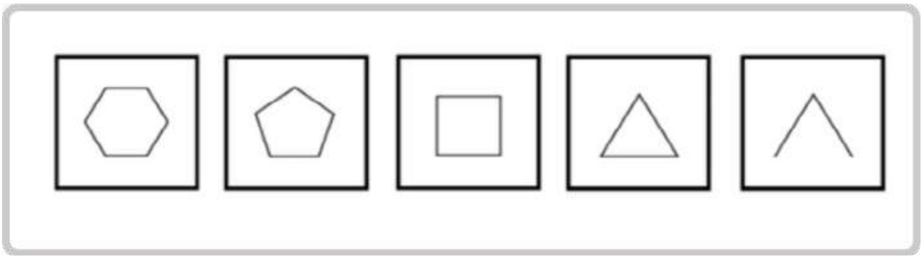

Figure 3. Sequence

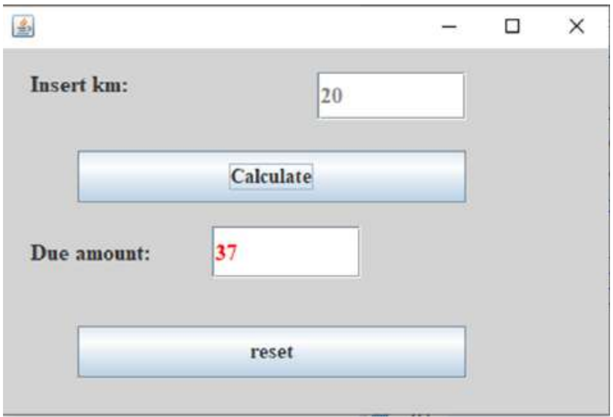

Figure 4. Software for calculating the amount due when using a taxi

For the software solution, we consider appropriate the task of developing a Windows Forms desktop application, since console programming demotivates students to some extent who prefer to use the interface they are used to as users from an early age. The students are learning the Java programming language, but some of them, who are additionally involved in programming, said that C# is easier for them to use. Because when they put a new shape control in Java in Eclipse, the code for size, colour, font, etc. is automatically generated, while in C# these things are hidden. They also said that some methods are more powerful in C# than in Java. Figure 4 shows the execution of a program with specific data.

Task: Boarding a taxi (which includes 3 km) costs BGN 3. Each additional kilometer is paid at a rate of BGN 2 per km. The travel distance is 20 km. Calculate the cost of the trip.

The algorithm is implemented in the `btnCalculate` method after clicking the *Calculate* button. Figure 5 shows the algorithm.

```
btnCalculate = new JButton("Calculate");
btnCalculate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        int wholeDistance=Integer.parseInt(distance.getText());
        int distance1=wholeDistance-3;
        int fee=(distance1*2)+3;
        Result.setText(String.valueOf(fee));
    }
});
```

Figure 5. The algorithm in btn Calculate method

The variable `wholeDistance` is used to store the distance that will be traveled by taxi. The variable `distance1` is used to store the new value that is obtained when removing 3 km initial fee. Each additional kilometer is paid at a rate of BGN 2 per km. The first term in this problem will be equal to `a1 = 3p`. The difference in progression `d = 2`.

For homework, students were assigned the task to calculate the required number of seats for a stadium. This helps to consolidate knowledge. The class commented on the validation of the input data (the size of the stadium). For the convenience of users, the students suggested the form app to have a drop-down list or several options for the size of the stadium as this will prevent mistakes when entering data manually. During our discussion, questions arose about solving the problem of using arithmetic progression. The students encountered difficulties in calculating the size of the stadium. The city of Veliko Tarnovo has a very nice stadium, and we went on site to discuss how to calculate its size.

Several surveys were conducted among students in grades 8 and 11. The first survey was completed with the new approach using textbooks approved by the Ministry of Education. The other surveys we conducted were about the performance of the students in mathematics, about the tasks and projects they prefer to create, and about where they encounter the most difficulties.

Here are three questions in our first survey:

1. Was the information in the Computer Science textbook for Grade 8 concerning the development of your first program understandable to you?
2. Do you grasp math lessons better after taking the basic programming concepts?
3. Would you recommend learning the basic concepts of programming and creating a form app in parallel?

The response of the students is positive (Figure 6). Learning computer science in Grade 8 only through textbooks proves completely insufficient and incomprehensible for them. The approach of learning the basic concepts and creating a form app in parallel is very well accepted by them. Our practice shows that what 8^{th} graders find most difficult are the access modifiers. There are plenty of them in contemporary programming languages. Another difficulty is related to the development environment we use. It requires to manually write some commands for the program to work. Therefore, the students must remember these commands in advance (they have not reached this point of the curriculum yet). That is why we think that using C# and Visual Studio is most convenient for school students – the programming environment does most of this job. They do not have to remember anything without understanding it.

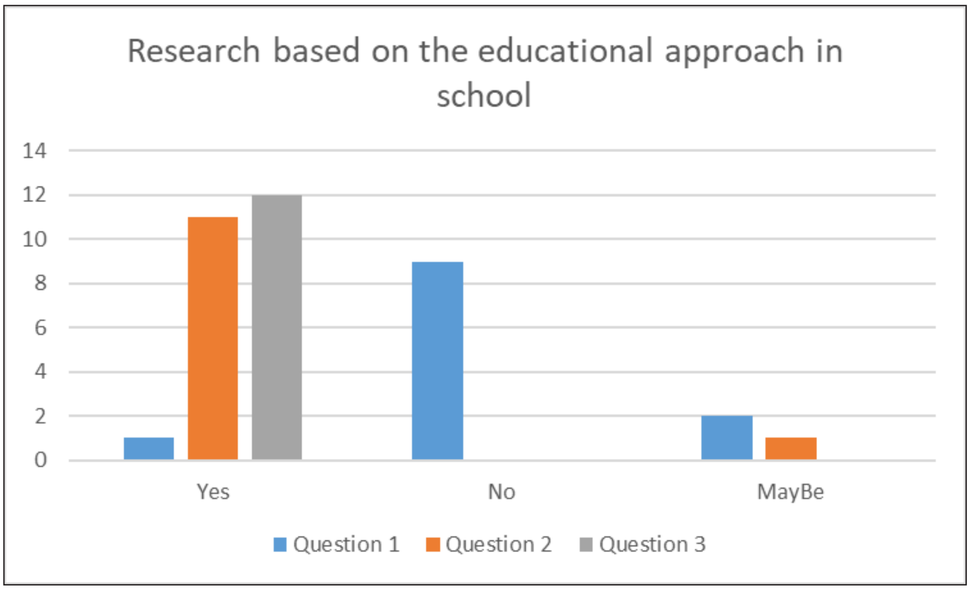

Figure 6. Opinion of 8th grade students about the new approach

Our approach manages to improve students' understanding of the application of programming and mathematics in real life. Through this approach, they can solve real problems using mathematics, and thus create software for the implementation of all ideas. The difficulty is in finding or creating suitable tasks or projects for this purpose.

From our work with students and the surveys conducted among them, we can conclude that students prefer to solve mathematical and economic problems, and that they mostly like to create games. What we notice is that they talk about the subject of mathematics with love. They attend computer science classes with ideas for projects, using what they have learned in both subjects. In the last few months, ideas for projects in computer science classes have come from students. This is the greatest success so far because students are becoming the engine of the class.

CONCLUSION

The need for students to find application of the knowledge they acquire in school is more than compulsory. The theoretical and practical connection between information from different subject areas leads to deepening of their competencies. The link between mathematics and computer science provides timely challenge and opportunity for curriculum development in schools. There are many tools for reviving it between the two subjects. The lesson presented in this paper demonstrates only a small part of the opportunities that both disciplines provide.

Teaching computer science and maths has always been a challenge. The difficulties arise from both the constant development of technology and the changing psychological profile of students. Experimenting with new approaches, tools and instruments is part of every educator's job. The approach we apply combines ideas and good practices from the classical approaches with the use of innovative tools. An essential element of the approach is motivation by confronting students with a real problem; spiral upgrade of the developed applications in mastering new techniques and integrating them into the training methods from the software production process, such as pair programming.

Our two-year observations show that students are very enthusiastic about this type of learning because they really see application in both subjects – mathematics and computer science. For our future work, we are currently discussing new projects related to statistics, trigonometric functions, and quadratic function.

BIBLIOGRAPHY

Chotova, G. (2019). Programming as a Means of Implementation of Cross-Curriculum Relations, *Scientific Conference of Young Researchers*, Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo.

Donchev, Iv., G. Chotova (2019). Objects-Early Approach to Teaching Informatics at School Level, *Mathematics, Computer Science and Education*, Vol. 2, No. 2, pp. 93 – 100.

Darling-Hammond, L., L. Flook, Ch. Cook-Harvey, Br. Barron, D. Osher (2020). Implications for Educational Practice of the Science of Learning and Development. Applied *Developmental Science*, 24:2, pp. 97 – 140, DOI: 10.1080/10888691.2018.1537791

Ivanov, I. (2005). Interactive Teaching Methods, *Jubilee Scientific Conference with International Participation, dedicated to the 50th anniversary of the Department for Information and In-Service Training of Teachers (DIPKU) – Varna*, on the following topic: "Education and Qualification of Teachers – Development and Projections in the 21st Century".

Nikolova, N. (2016). *Design-Research Approach in Teaching Informatics and Information Technology*. Dissertation for obtaining a doctoral degree at the St. Kliment Ohridski University of Sofia, Faculty of Mathematics and Informatics.

Pink, D. (2011). *Drive: The Surprising Truth About What Motivates Us.* New York: Riverhead Books.

Scratch Official Website, https://scratch.mit.edu/

The Joint Task Force on Computing Curricula (2001), *Computing Curricula 2001*. Computer Science.

Wing, J. (2006). Computational Thinking, *Communications of the ACM*, 49, (3), pp. 33 – 35.