Pol. J. Appl. Sci., **2017**, *3*, 136-140

TEACHING PROGRAMMING

Aneta A. Wiktorzak[1], Natalia Malenko[2]

[1]*Faculty of Computer Science and Food Science*
Lomza State University of Applied Sciences, Lomza, Poland
[2]*Faculty of Social Sciences and Humanities*
Lomza State University of Applied Sciences, Lomza, Poland

E-mail: awiktorzak@pwsip.edu.pl

Abstract: Our relationship with new technologies is becoming increasingly close, and the language we use to develop them becomes Latin of the digital age. The basic knowledge of programming languages has become over time indispensable in order to deal with the changing world. The article brings together the most important aspects of programming, especially teaching programming to the youngest students. The aspects of the ministerial plans concerning the implementation of the new core curriculum in teaching IT, as well as the perspectives and possibilities it provides in reference to the development of logical thinking since the early age are discussed here. The implementation of the new IT curriculum will be accompanied by the preparation of professional materials, primarily for teachers: curriculum proposals for all educational stages, didactic materials that cover the curriculum and materials for working with students (e.g. Scratch, Hour of Code, Baltie). The materials for education and training of computer science teachers will constitute a separate package. Learning digital languages teaches them analytical thinking and problem-solving based on causal logic. These skills are not only useful in everyday life but also in learning foreign languages or other subjects. But above all, programming is the competence that familiarizes children with technologies which will play an increasingly important role in our lives.

Key words: digital age, programming language, teaching programming, computational thinking, Scratch, Hour of Code, Baltie, IT, logical thinking, computer science

Introduction

Our relationship with new technologies is becoming increasingly close, and the language we use to develop it is becoming the *lingua franca* of the digital age. The basic knowledge of programming languages has over time become indispensable in order to deal with the changing world. In Poland, we have observed the benefits of the rapid increase in the development of digital competencies among the youngest children. A lot of them are enrolled in programming courses, and many educational institutions are involved in teaching programming even to toddlers. The knowledge of programming will also become a constant element taught in schools, included in the earliest stages. It does not come as a surprise. Not only can programming prepare children to cope with the challenges of the surrounding world, but it also helps them to grow at every stage of adolescence – especially the earliest ones, when knowledge is acquired most efficiently, and the competences gained will provide them with benefits for the rest of their lives.

Programming is easy to learn. Contrary to popular belief – anyone can learn it from scratch. As Barack Obama once stated, nobody was born a computer scientist. Computer studies – is not a hermetic area, accessible only to scientific minds and unattainable for those predisposed to humanities. This artificial divide is ridiculous. Anyone can learn simple coding languages and understand the basics of their operation. It all depends on how the knowledge is transferred. Further stages of education are adjusted to the student's age. In the case of the youngest children – first graders or even preschoolers – programming is mostly about developing a way of thinking, about the perception of coding languages as a path to a goal, and a way to learn how to come to logical conclusions. Steve Jobs once said: "Everybody in this country should learn to program a computer, because it teaches you how to think." Although it was said in reference to America, its importance is no less relevant in reference to any country, and Poland in particular.

Teaching programming in IT classes at all levels of education

Programming as a school subject has already been introduced in many schools in western Europe. The educational benefits of such a solution have also been recognized by the Polish Government. The Ministry of Education is planning to introduce a pilot program starting with the current (2016/2017) school year [1]. If the pilot program is successful, programming will have a permanent position in the curriculum for Polish students. The Council of Computerization of Education in the Ministry of National Education has presented a proposal of a new core curriculum for IT education which includes the subject of computer science which also comprises programming. The core curriculum

will cover all the students, from the first to the last grade in school.

Information technology is becoming a common language in almost every field of human activities, and it equips students with new methods and tools. Nowadays, the basic task of school is defined as the development of students' literacy in reading, writing, as well as math which requires expansion into other skills called "computational thinking", so that students are able to solve problems in various fields using methods and tools derived from computer science.

According to Jeannette Wing, who introduced this term in 2006, computational thinking defines useful attitudes and skills that every computer user, not just an IT specialist, should strive to develop and apply. With a broad outlook on IT competencies, computer science is not limited to knowledge about computers, but it provides strategies for mental activities that can be used to the benefit of other fields as well as in everyday life [2].

The term computational thinking defines the thought processes that accompany the formulation of problems and problem-solving in a way that enables them to be effectively implemented using a computer. It covers a wide range of useful intellectual strategies and tools that can be used to solve problems in a variety of areas with the help of computers and computer-based methods derived from computer information processing and computer problem - solving in various fields. Finally, it integrates human thinking with computer capabilities.

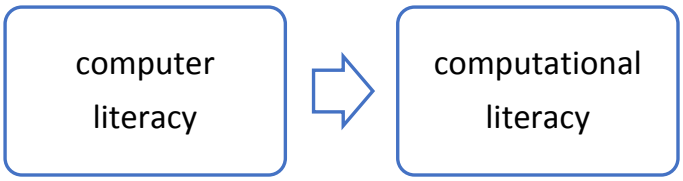

Fig. 1: The transition from computer literacy to computational literacy

One of the aims of general IT education is to raise awareness of the importance of computer science as an independent discipline. Early school contact with computer science and programming should familiarize students with the value of the field and its applications and stimulate their interest and motivation in the choice of further education and career prospects. No other discipline creates such opportunities for the future as computer science, regardless of the chosen field of study and the occupational interests of the learner [3].

In the coming years, there will be a shortage of IT staff in the world. The governments of the largest states have promptly responded to the necessity of taking decisive steps to introduce computer science, alongside programming, into the canon of education of all the pupils from an early age. Our educational system should not overlook this trend; especially for the reason that IT has been taught for over 30 years in our country. The first curriculum for secondary schools, which contained the elements of programming, was approved in 1985, and since then IT has permanently entered the core curriculum. First of all, IT subjects have recently been introduced at all educational stages; secondly, IT teachers work in schools; thirdly, schools are equipped with basic IT devices and tools, and finally, the access to computer programming environments is widely available and free. The prospective changes in IT education empower educational authorities to focus intently on activities such as: the new core curriculum for IT, which should be implemented simultaneously at every stage of education in all grades and schools.

In the submitted proposal of the Ministry of Education, programming is a part of the IT course from a young age. In accordance with the proposal, IT is not taught as a separate skill or an independent subject, it is more of a tool for problem-solving with the help of a computer. It serves, among others, to introduce IT concepts and developing IT methods. Programming teaches such important skills as logical thinking and clear presentation of thoughts and ideas. It fosters proper organization of work while solving problems and builds up the co-operative competencies which are necessary in almost every profession today. The skills acquired while learning programming are not only necessary for IT subjects, but they are also useful in other ones as well as in different occupations, which the present-day students will perform in the future. These skills allow learners to move from a position of digital consumers to that of digital creators, who act as rulers of technology, and are not the ones who merely subject to it [4].

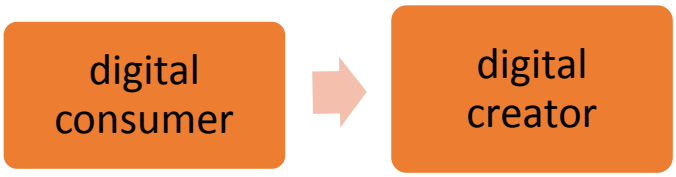

Fig. 2: The transition from a digital consumer to a digital creator

Another argument that justifies the front-line implementation of the proposed changes is the widespread willingness of students to participate in activities especially devoted to the development of programming skills. Students do not want to wait until the proposed reform of IT education has been implemented. If it is initiated soon enough, they will have the abovementioned subjects in the school timetables in a few years. At present, having no such offer from schools, they participate in extracurricular activities, which do not satisfy the demand among all pupils at all educational stages. The success in the implementation of the proposed changes will depend on how soon the teachers will be qualified and ready to make such changes, and how soon school management will support them. Many

teachers will have to expand their competences to a level that enables them to implement the new core curriculum. The scope of the necessary competences will be determined by the standards in education of IT teachers. The adequate qualifications will be confirmed by a certificate, which the teachers will be able to apply for. The teacher will be able to evaluate his/her competences during the classroom activities with the students. The importance of these solutions should be emphasized here – standards, evaluation systems and certification are not discouraging barriers or hurdles for the teachers. On the contrary, these will support their development. At the same time, they will determine the training scope, level and directions of teacher education and training in colleges and other educational institutions. The front-line implementation of the new IT core curriculum will be accompanied by front-line teacher training, mainly through education at a university level.

The implementation of the new IT curriculum will be accompanied by the preparation of professional materials, primarily for teachers: curriculum proposals for all educational stages, didactic materials that cover the curriculum and materials for working with students. The materials for the education and training of computer science teachers will constitute a separate package [5].

Programmable blocks

New technologies provide tremendous opportunities for teaching algorithmic thinking. It is a good idea to use a computer program that is a puzzle of matching "blocks". Scratch (scratch.mit.edu) has become the benchmark for many ventures [6]. This visual programming language was created relatively recently, in 2003. Its creator is a professor at MIT (Massachusetts Institute of Technology) Mitchel Resnick, who continues to work on the improvement of the language. The scholar promotes and supports the community of Scratch users. Scratch is free of charge. It works both in a web browser and as a separate offline program which can be downloaded and installed on a computer.

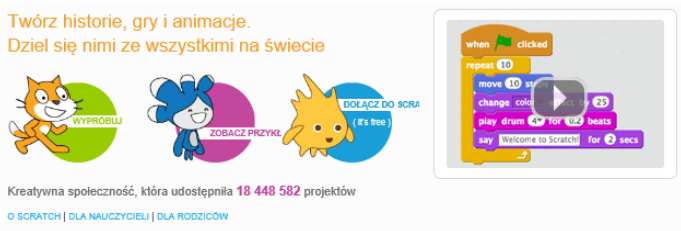

Fig. 3: Scratch (the official site sratch.mit.edu/pl)

Scratch does not only develop algorithmic thinking, but it also understands and tests the specific concepts used in the majority of most popular programming languages, such as "loop", "variables", "conditions", or "functions". The tool can be used to a limited extent by the introduction of simple exercises or activities into the lesson or carrying out larger projects. In the choice of a project pathway, students usually have to solve a more complex problem; e.g. it can be related to a given object or a part of the material. This approach is extremely important because it teaches how the mono process of creating a program should be linked to its design. In addition, the teacher can make students aware that programming is, de facto, the ability to solve problems and respond to challenges in a creative way. The Scratch system is widely used in Polish schools in connection with the Masters of Coding programs. Under its framework, teachers of different educational levels across the country have been trained in the basics of programming in Scratch (and beyond). With Masters of Coding, there are also free technical resources available to help educators get started with programming. Materials are available on wiki.mistrzowiekodowania.pl [7].

In addition to Scratch and its variants, it is impossible not to mention the tools provided within an Hour of Code by Code.org [8]. Here we also program by dragging and putting together ready-made blocks, but unlike in the case of the previously mentioned tool we have limited possibilities to create our own designs. We can write our own mini games, but the choice of heroes is limited.

Using the thematic paths (inspired by, for example, Minecraft, Star Wars, Ice Land, Angry Birds, Flappy Bird and the general paths – Play Labs and the Artist), we go through the stages of connecting blocks in increasingly complex configurations or create our own projects. Undoubtedly, an advantage of this tool is the ability to administer the virtual classroom and track the progress of individual students and the gaming system (cups and medals for the completed task). Teacher experience shows that this form of programming is very appealing to children because they meet in virtual space with their favourite characters. Additionally, the completion of each version of Hour of Code is confirmed by a professional diploma that can be personalized to download and printed out or shared in the social networking media.

One hundred million students have participated In the Hour of Code, which is an annual event organized by Code.org, in the last two years. Microsoft and Code.org have created a tutorial for students. It is available for free and contains 14 stages, visually, as if taken from the actual game "Minecraft". The course looks exactly like a very popular (also among young players) production, but its job is more concrete than the survival in a built-of-blocks world. The course designed by Microsoft and Code.org teaches the fundamentals of programming and has a very clear interface. The learners "drag and drop" the elements combining them into a series of commands. Then they just click "run" and see

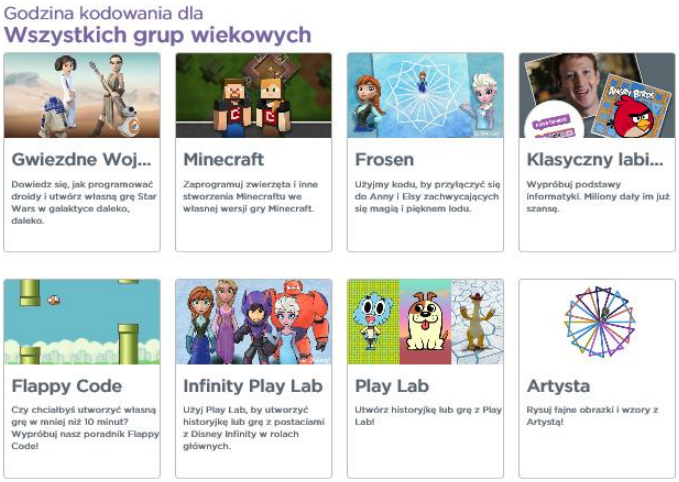

Fig. 4: Project proposals within the Hour of Code [9].

the effect – a Minecraft hero executes the actions they had programmed. According to Hadi Partovi of Code.org, this visual exercise is similar to the way the beginner programmers learn the fundamentals of programming philosophy. Partovi believes that while using such an extremely popular game, children will learn to think like programmers and will be curious to take a bigger challenge, i.e. learning the actual programming languages. It is even better if, thanks to "Minecraft," children will understand the fundamentals of programming more easily.

Fig. 5: The Tutorial of the "Minecraft" project for the Hour of Code program.

It is worth mentioning that other "brick" systems, such as Baltie, for instance, (sgpsys.com/pl) [10], have been successfully used in many Polish schools for years. Baltie is also a free tool. A very interesting feature of this system is the competition server that automatically checks the work of students competing against one another while performing their tasks. Its use does not require the ability to read – all programming elements are pictorial. Undoubtedly, this is a precursor program, which was created in 1999 by a Czech, Bohumir Soukup. The community of users and the amount of material are impressive, as well as the number of projects initiated around Baltie, which offers professional tools and learning paths from the initial drawings to programming in C ++. Due to the pixel-based graphic interface, Baltie is not as appealing to young coders as the previously described solutions, but the idea behind it is the same – it has to entertain, teach, and develop broadly understood programming thinking.

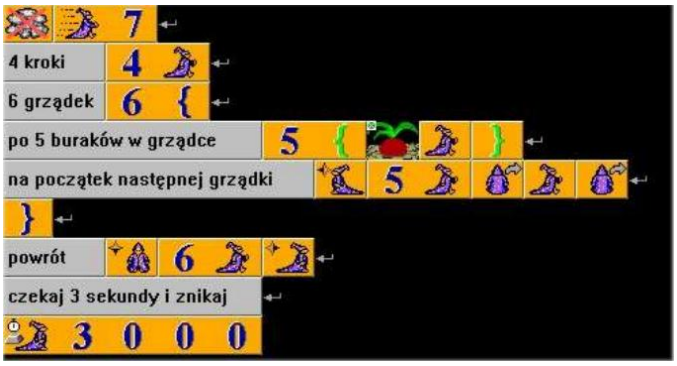

Fig. 6: Fragment of Baltie programming code [11].

It is worth noting that no matter which tool we eventually choose to work with students, it is not an aim in itself. You may as well try to create (software) (programming) blocks on paper cards (sheets of paper) – making simple programs out of them (with their help) can make your students think that the most important thing is how we think and not the language we use. Let us give ourselves and our students a chance to develop through programming. The concept of blocks – both analogue and digital – that have to fit together to create a logical whole, is understood even by toddlers. Using this analogy, learning programming systems perfectly help to shape logical thinking [12].

Summary

The article brings together the most important aspects of programming, especially teaching programming to the youngest students. The aspects of the ministerial plans concerning the implementation of the new core curriculum in teaching IT, as well as the perspectives and possibilities it provides in reference to the development of logical thinking from an early age are discussed here. Bill Gates states that "[l]earning to write programs is brain gymnastics. It allows you to develop the ability to effectively think about things that are not related to IT." These, together with information and communication skills, creativity and the realization of one's own projects as well as the intellectual development reinforce the sense of self-esteem in children, which results in a better possibility of getting a good job in the future and achieving their goals. Programming gives learners the tools to use technology creatively. Due to that, they will not be just passive recipients of "infoglut" but will develop and actively interact with digital reality. Moreover, programming skills can be used in other areas,

such as programming a database of newly learned English words. The undeniable advantage of programming for the youngest pupils is also its role in their development and adolescence. Learning digital languages teaches them analytical thinking and problem-solving based on causal logic. These skills are not only useful in everyday life but also in learning foreign languages or other subjects. But above all, programming is the competence that familiarizes children with technologies which will play an increasingly important role in our lives. And children who know the fundamentals of coding will simply be better able to handle the digital world surrounding them.

Literature

[1] Ministerstwo Edukacji Narodowej. Pilotaż programowania. http://programowanie.men.gov.pl/.

[2] Wing J. M. Computational thinking. *COMMUNICATIONS OF THE ACM*, 49(3):35, 2006.

[3] Wiktorzak A.A. Nowy wymiar informatyki. wsparcie rozwoju kompetencji nauczycieli w zakresie nauczania programowania, materiały ze szkolenia dyrektorów oraz nauczycieli nauczania początkowego społeczno-oświatowego stowarzyszenia pomocy pokrzywdzonym i niepełnosprawnym edukator., 22 października 2016.

[4] Ministerstwo Edukacji Narodowej. Wprowadzenie do propozycji zmian w obowiązującej podstawie programowej z przedmiotów informatycznych. https://men.gov.pl/wp-content/uploads/2016/05/propozycje-zmian-do-projektu-nowej-podstawy-programowej-1.pdf.

[5] Sysło M. Nauka programowania w ramach kształcenia informatycznego na wszystkich poziomach kształcenia. *IT w Edukacji*, (1):10, 2016.

[6] https://scratch.mit.edu/, strona nauki programowania z językiem Scratch.

[7] http://mistrzowiekodowania.pl, strona nauki programowania, Program Mistrzowie Kodowania.

[8] https://code.org, strona nauki programowania, gry, zabawy, aktywności.

[9] https://studio.code.org/s/minecraft/stage/1/puzzle/1, strona Godziny Kodowania z Minecraftem.

[10] http://sgpsys.com/pl/Default.asp, strona nauki kodowania i programowania Baltie.

[11] http://sgpsys.com/pl/Lekcje/GBlekcja3.pdf, strona z przykładową lekcją kodowania w Baltie.

[12] Krawczyk A. Programowalne klocki – to się dobrze składa. *IT w Edukacji*, (1):18–20, 2016.

Received: 2017
Accepted: 2017