ADAM NARUSZEWICZ
ORCID 0000-0003-4159-2324

Implementation of Programming Contents at the Early School Stage – Outline

Realizacja treści dotyczących programowania na etapie wczesnoszkolnym – zarys tematyki

DOI 10.25951/4671

Programming – about a third language, important in modern life

Today's teachers and their students live in a world where it is extremely difficult and in many cases, it is impossible to function properly without digital skills. In order to carry out many of today's tasks, mainly professional, knowledge of the mother tongue and foreign language may turn out to be insufficient, especially in a situation where it will be necessary to control a digital device. In such situations, it is essential to know the rules of programming. Furthermore, knowledge of at least one of the many programming languages understood by machines is necessary. It is important to know the syntax, keywords and commands used in a given programming language, which in the form of lines of codes are like sentences in communication with a digital machine. Programming languages are called machine languages (Jernajczyk, Skowron, Drapała 2013). They are examples of the so-called a third language (Kuźmińska-Sołśnia 2018), and their knowledge is very important in the era of constantly progressing digitization.

Examples of programming languages include, among others, the Assembler. The commands issued by the user in this kind of language correspond directly to the binary instructions understood by the processor. It is worth noting that programming is essentially issuing instructions to the processor, and it involves replacing more human-accessible commands with zero-one codes that are interpreted by the processor. To put it simply – on this principle assemblers as well as other programming languages, mediate communication between the user and the digital machine.

Examples of programming languages also include Basic, Pascal, object-oriented languages such as C ++ and Delphi. These are higher-generation languages that are no longer directly related to the processor as Assemblers. The programming languages of these levels run under specific hardware and operating system. This does not apply to the languages of the highest generations, thanks to which it is possible to write programs independently of the hardware and operating system.

Knowledge of the rules and at least one of the numerous programming languages makes it possible to use various types of digital devices effectively and creatively. Programming is related to math. It comes from computer science, where theoretical knowledge and knowledge of numerous concepts are important. Programming supports algorithmic thinking, which should be shaped from an early age (Bobko, Bubula, Marek, Sala, Wójciak 2018). Computational thinking is also associated with this activity. It consists in skilful problem solving with the use of IT methods and tools (Wing 2006; Rostański, Borczyk, Lipka 2016). "The theoretical and practical nature of programming makes it an important and necessary and even essential element of human education" (Jernajczyk, Skowron, Drapała 2013, p. 1). Programming is the ability to use digital machines in a conscious manner, thus ensuring their users to harness their capabilities in full.

Programming skills – so far associated with a specific group of people, enthusiasts or professionals, are now being developed among the youngest. The recipients of the content constituting the basis for programming today are preschool children and younger school-age children (Kuźmińska-Sołśnia and Ziębakowska-Cecot 2017a). Currently, conducting classes related to programming is part of the responsibilities of not only IT teachers, but also early education teachers (Kuźmińska-Sołśnia and Ziębakowska-Cecot 2017b). It results, inter alia, from the provisions of the core curriculum.

Contents of the core curriculum for programming at the stage of early education

The core curriculum for early childhood education, in force since 2017, includes the learning results related to programming. As part of the detailed requirements regarding the content of teaching in IT education, it is assumed that a student completing class III should program "visually: simple situations or stories according to their own ideas and ideas developed together with other students, individual commands, as well as their sequences controlling the object on the computer screen or another digital device" (Ministry of National Education

2017, p. 44). This goal should be pursued in conjunction with the content of different types of education – in line with integrated education.

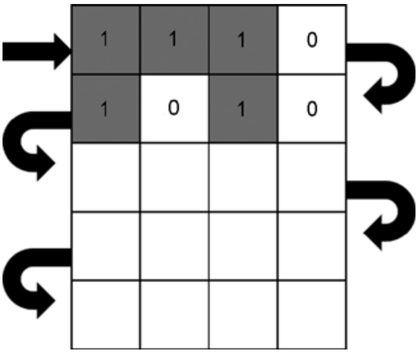
In the core curriculum, the concept of programming is understood broadly as it is not limited only to writing computer programs. The content of the preamble to the curriculum describes programming as a complex problem-solving process, which "supports the development of such skills as logical thinking, precise presentation of thoughts and ideas, promotes good work organization, builds competencies needed for teamwork and effective project implementation" (Ministry of National Education 2017, p. 44). Moreover, children learn about the concepts and methods of computer science while developing logical and computational thinking skills.

The core curriculum document enables teachers to freely choose methods, forms and tools in working with children. The teacher can therefore adopt his own strategy of implementing programming lessons, including coding, choosing traditional methods (without the use of computer equipment) or using digital technologies in work with children.

In the literature on methodology related to computer science education in classes I–III, the term coding appears in the context of programming. In the education of children, coding consists in creating or interpreting codes transmitted, among others, in a text or visual form. This operation can be carried out without the use of computer hardware as opposed to professional programming where instructions are conveyed through a line of computer program code. The graphic below shows an example of graphic dictation, which is one example of a computer-free exercise aimed at developing coding skills.

Write the numbers in the squares according to the arrows.
Then color only the squares with ones to decode the letter sign.
The first two rows have already been completed.

~~1110~~ ~~0101~~ 1110 0101 0000



1	1	1	0
1	0	1	0

Figure 1. Example of a task in a form similar to graphic dictation

Source: own study.

The implementation of issues related to programming and coding in classes I–III can take place without the use of electronic equipment – with the use of traditional means or by using computers and robots. In the traditional approach, aids such as board games, natural material or various artistic and technical forms will work. An example is the CodeBook board game (Kolarz, Tłuczykont 2017). Examples of computer-free activities include coding on the carpet (Świć 2017), where generally accessible items can be used. Among the examples of items, the author indicates, inter alia, paper cards, sashes, and chairs, describing in detail the method of using coloured cups. Children can use them to arrange various "(…) towers, following the code provided, which may define the number and colour of the cups, and even precisely indicate the colour of their arrangement" (p. 61). When deciding to use digital technologies, the teacher can use computer programs and/or devices connected to the computer, including robots, which will help children learn the basics of programming.

In pedagogical practice – introducing children to the idea of coding and programming is based on playful activities during which they learn the informal meanings of IT concepts. Among them, there are concepts such as algorithm and instruction (Bobko, Bubula, Marek, Sala, Wójciak 2018). It is worth noting that the instructions are an element that creates the algorithm, that is, "a procedure leading to the solution of a given problem, specifying a sequence of elementary steps that must be performed for this purpose" (PWN Internet Encyclopedia). Recipes are popular and frequently used examples of algorithms. In the literature, professional programmers refer, among others, to examples with pasta. It may seem very mundane, but the introduction to teaching programming – also for adults – begins with an attempt to develop this type of recipe oneself. The following content presents an example recipe taken from the publication (Kierzkowski 2012) containing exercises for teaching programming mainly for adult audiences.

1. Boil water in a pot.
2. Put the pasta into boiling water so that it is immersed in it.
3. Add salt to taste (in the kitchen this concept is more acceptable than in computer science – here you would have to define exactly what it means to taste, and perhaps design a system advising if the amount of salt is sufficient; because we want to create a simple algorithm and exact, let's accept my standard – 3/4 tablespoon of table salt for 5 litres of water).
4. Cook for about 8 minutes, stirring occasionally.
5. Drain the boiled pasta using a colander.

6. Also, using a colander, pour cold water over the pasta to prevent it from sticking together.
7. Pour the pasta onto the plate (Kierzkowski 2012, p. 12).

The content indicates that the algorithm is a step-by-step description of the task implementation method. It is also worth noting that the algorithm is a very precise instruction, sometimes containing obvious issues. Precision and clarity are particularly important when creating an algorithm for machines that operate with mathematical precision. Omitting an issue that may seem obvious to a human will cause the program/machine to malfunction or stop it completely.

By creating similar algorithms, children can learn the precision of planning a sequence of activities assuming their execution by a machine or a human. In the case of machines, the algorithm must be expressed in a form that can be interpreted by the device. Examples include algorithms expressed as instructions in the form of lines of code written in a specific programming language or graphic codes composed of a sequence of colours used, among others, in robots for teaching programming/coding.

Another term that is associated with the algorithm is a block diagram, a way of visual presentation of an algorithm, which, unlike a line of code, is more accessible to people. The graphic below presents in a simplified way an example of an algorithm written in two forms: text and block diagram. The algorithm used in the example describes the way of moving the gray square on the 4×4 board from the field A to field B.

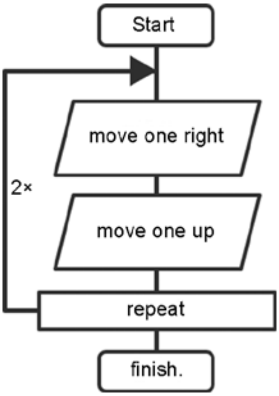
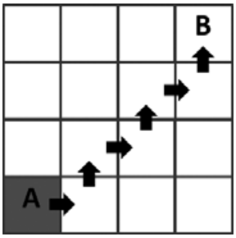
Algorithm with instructions in the form of text	Algorithm with instructions in the form of a block diagram	Expected way to move the gray square
Start; repeat three times: move one right move one up finish. The content of instructions for computer programs varies syntactically depending on the programming language.	 Start move one right 2× move one up repeat finish.	

Figure 2. Examples of ways to present an algorithm

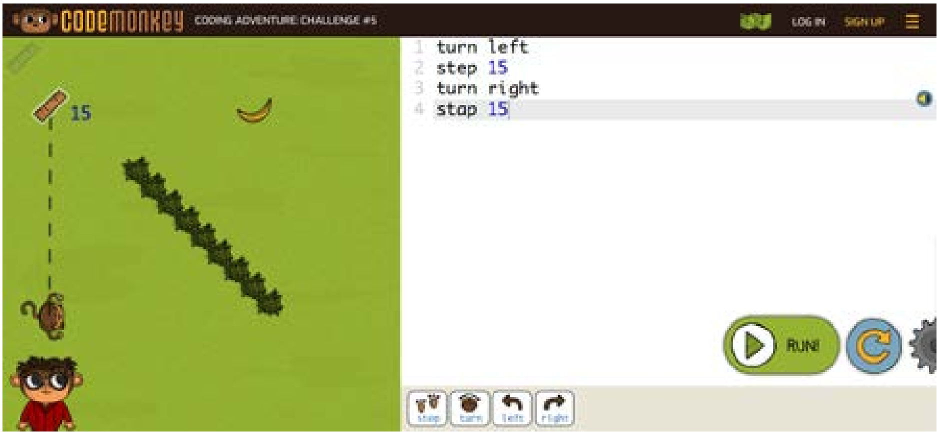
Source: own study based on Jernajczyk, Skowron, Drapała (2013).

As part of the presented example of the algorithm, the program loop instruction was used, which is often used in programming. The program loop is also accompanied by a conditional statement that stops the loop after the appropriate number of repetitions. The issues related to algorithms and various types of instructions were described in detail, among others, in publications (Jernajczyk, Skowron, Drapała 2013; Bobko, Bubula, Marek, Sala, Wójciak 2018).

Computer software for child programming

As part of the on-line resources, various types of tools are available to support teaching programming. Many of them do not require installation on a computer disk and are in the form of computer games available directly from a web browser. The principle of operation of this type of tools is to control the object on the computer screen in an appropriate manner by issuing commands in the form of lines of codes. The interface of such programs usually consists of two parts: the space for creating the program and the window in which the effect of the program is presented.

CodeMonkey is an example of one of the computer games that use the above-described principle to develop programming skills. With proper use of the game by the teacher, children have the opportunity to learn the basics of programming through play. The application is commercial in nature, but its free functionalities allow going through several stages.



```
turn left
step 15
turn right
step 15
```

Figure 3. Code Monkey game user interface screenshot

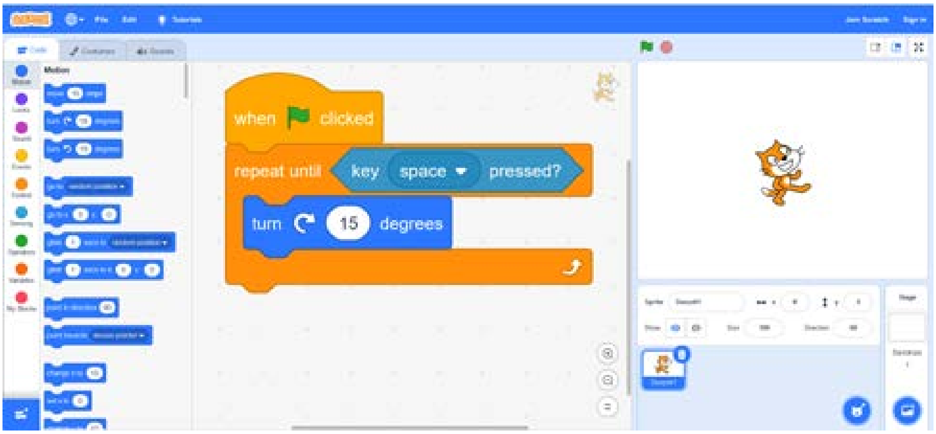
Source: Screenshot from game website https://app.codemonkey.com/challenges/5 (access date: 2.01.2021).

This is enough to familiarize children with the basic features of computer programs. The goal of CodeMonkey is to get the monkey to get a check-up.

The space on the left shows the current problem to be solved by the child. The effect of programming is shown in the same window after pressing the RUN button. The right part shows the program code along with the parameters of individual instructions. Numerical parameters are marked in different colour. If the monkey does not reach the banana due to incorrectly written instructions, the program indicates errors and suggests how to solve the problem.

This will require the child to control the monkey and other elements on the board – initially by using single instructions. As part of the subsequent difficulty levels, the pupil will have to use, among others, a program loop, thanks to which the hero controlled on the screen will be able to perform a series of cyclical activities. While programming, the game requires simple calculations and estimates. In most cases, one of the required instruction parameters, in addition to the direction of movement, will be the monkey's distance from the banana that the child will have to measure or estimate.

Among the tools supporting teaching programming, there are also those that use the idea of a block diagram in their operation. This provides the possibility of visual programming, in accordance with the assumptions of the core curriculum. In practice, the child can create programs by arranging blocks representing individual instructions or their parameters. Examples of this type of software include Scratch (Biała, Nowicki 2011) and PixBlock.



```
when 🏁 clicked
repeat until key space ▾ pressed?
    turn ↻ 15 degrees
```

Figure 4. Scratch game user interface screenshot

Source: Screenshot from game website https://scratch.mit.edu/projects/editor/?tutorial=getStarted (access date: 2.01.2021)

The graphic above shows a view of the Scratch application window containing a block diagram, in which there is an instruction made of blocks by the user that determines how to move the character shown on the right.

With Scratch and other similar tools, the child can develop the program in a way that is similar to building with blocks. On the left side of the application window, there is a tray with blocks representing various instructions to be used in the program.

Among other things, these instructions allow moving the character on the screen, change its appearance, repeat a part of the program using a program loop, or execute a part of the program under a certain condition by using a conditional statement or conditional loop. Moreover, it is possible to create programs that interact with the user. Thanks to this, children can develop simple computer games in Scratch.

As part of the program website, pupils have the opportunity to publish their work. In addition to their own computer games, children publish there, among others, quizzes and short animated stories. Among the many student projects, there are also interactive teaching aids, often created during classes.

Robots to learn programming

Introducing children to the world of programming can take place with the use of educational robots, which in many cases work in conjunction with dedicated to them computer applications. The possibility of implementing the new core curriculum through the use of robotics is highlighted by, among others, Jurkiewicz (2018). There are many offers of producers of educational aids related to the use of robots in kindergartens and schools.

Examples of this type of offer appeared in the publication entitled Myśliwiec (2017), where the author undertook to discuss the subject of a child who is a robot builder. The educational offer includes devices in the form of structurally different robots – but their functionality and principle of operation are in most cases similar. Among the educational robots on the commercial market, there are devices dedicated to educational institutions. Among the articles published in the press for teachers, the following devices are described: BEE-BOT (Cugier 2017), Ozobot (Taniewicz 2017) and JIMU Robot (Zasoński 2017a). The above list contains only a few examples of the use of robots in teaching, because the offer is very wide.

There are robots on the market that can be disassembled and assembled by a child in various configurations. Examples of such devices are mBoT, and the LEGO® EducationWeDo 2.0 set. Thanks to this type of robots, it is possible to implement STEM classes, the idea of which is to combine natural, technical, mathematical and engineering contents (Plebańska, Trojańska 2018). An example of classes using robots for self-assembly is described in the article by Zasoński (2017b). The author uses littleBits robots to carry out the activities.

Ozobots are small devices with a base similar to a wheel. Their sizes fit within a cube with an edge of 3 cm. They are equipped with a series of optical sensors located on the underside of the housing to collect information about colours. Used sensors and a drive allow the device to move along the mapped route, in the form of a black line approximately 0.5 cm wide.

The figure below shows the Ozobot and the intersection designed for the children. Through a code composed of three colours, the children indicated to the robot to turn right of go to straight at the nearest intersection.

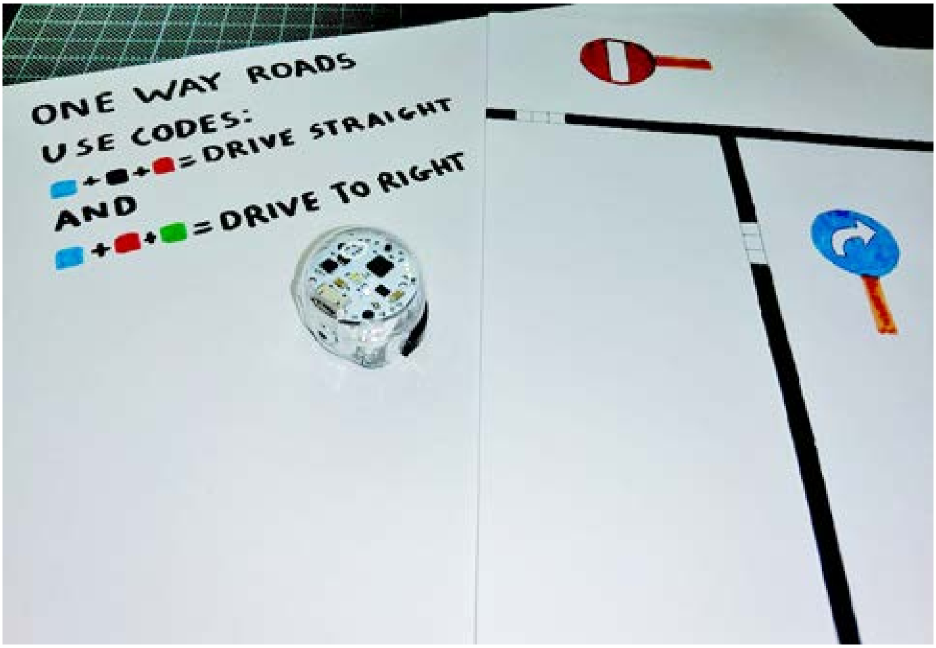


Figure 5. Image shows a robot and school task for kids about one way roads prepared by early education teacher

Source: own materials.

While moving – the device can read colour sequences (graphic codes) placed on the black line marking the route. The device reads codes in the form of two, three or four adjacent occurring squares in a combination of colours – red, green, blue and black. Lines and codes can be printed as well as drawn by hand with coloured markers.

Within the sequence of colours, commands are encoded, carried out by the robot as various actions. Among other things, the following commands are provided for: the turning direction of the device at the intersection and the change of the direction of movement to the opposite one, speed of movement or stopping time. For example: in order for a robot to go straight at the next intersection, it is necessary to place a code in front of it consisting of consecutively arranged squares in the following colors: blue, black and red. In addition, the robot can perform actions combining movement and lighting effects, because it is equipped with LEDs. In addition to generating lighting effects, the LEDs inform about the status of the device and display a colour similar to the ground on which the device is located. Ozobots also cooperate with an application dedicated to them, which allows issuing commands through a block diagram.

The role of the teacher is paramount

Hardware and software are very useful tools for a teacher to use. However, they are not always necessary. All it takes is an imaginative teacher. An example of conducting classes without the use of a computer is coding on the carpet (Świć 2017), where both robots – Ozobots, as well as simple generally accessible objects are used. Among the examples of items, the author indicates, among others, paper cards, sashes, and chairs, describing in detail the method of using coloured cups. Children can use them to arrange various "(…) towers, following the code provided, which may define the number and colour of the cups, and even precisely indicate the colour of their arrangement" (p. 61). In her work with cups, the author also uses an encoding mat resembling a chessboard or a board game. The mat is used, inter alia, for graphic dictations, during which children arrange coloured cups or squares according to the teacher's instructions.

Bearing in mind the content of this article – it is important to make children aware that controlling an object consists in issuing instructions using the code in the right form. In carrying out this task, the teacher can use various types of help, but the most important thing is his ingenuity. This article does not exhaust the issues discussed in it, but only indicates the direction of further research within the broad subject of programming/coding.

The number of services and hardware supporting the learning of programming is very large. This text presents only examples. In developing coding skills in children at the younger school age, it is extremely important to choose the right tools and working methods. The teacher's role in this is therefore important. The selection of the right tool should take into account the age of the pupils, their level of advancement in the field of coding and programming and the individual needs of the pupils.

BIBLIOGRAFIA

Biała P., Nowicki M. (2011), *Scratch – An Environment for Learning Programming. IT in Education: Materials from the 7th IwE 2010 Conference*, pp. 33–40. Toruń: Scientific Publishing House of the Nicolaus Copernicus University.

Bobko K., Bubula M., Marek J., Sala W., Wójciak P. (2018), *Programming and Robotics in Early Childhood Education*, Kraków: Lesser Poland Teacher Training Center, www.mcdn.edu.pl/wp-content/uploads/2019/05/Programowanie.pdf (access date: 20.02.2021).

Cugier I. (2017), *Travels with Bee Bot*, ICT in Education, June, pp. 22–23.

Jernajczyk J., Skowron B., Drapała J. (2013), *Machine Language*, Wrocław: Academy of Young Scholars and Artists, https://www.wroclaw.pl/files/wroclaw kodje/ Language of the scripts.pdf (access date: 10.01.2021).

Jurkiewicz A. (2018), *Programming Robots on the Computer Screen – According to the New Software Basis*, in: A.B. Kwiatkowska, M.M. Sysło (ed.), *Informatics in Education: Think Computationally*, Toruń: IwE (pp. 134–138).

Kierzkowski A. (2012), *Pascal. Practical Exercises*, Edition III, Gliwice: Helion.

Kolarz W., Tłuczykont K. (2017), *CodeBook Board Game – Fun and Education*, ICT in Education, December, pp. 50–51.

Kuźmińska-Sołśnia B. (2018), *Learning Programming / Coding in the Education of Children*, "Dydaktyka Informatyki", 13, pp. 121–128, http://www.di.univ.rzeszow.pl/Dydaktyka_informatyki_13_2018.pdf (access date: 2.01.2021).

Kuźmińska-Sołśnia B., Ziębakowska-Cecot K. (2017a), *Implementation of Programming Learning in Preschool and Early Childhood Education in Poland*, XXXth DidMatTech Dunajska Streda, Slovakia: Trnava University, Faculty of Education, www.bks.pr.radom.pl/publikacje/KZC_BKS_17.pdf (access date: 2.01.2021).

Kuźmińska-Sołśnia B., Ziębakowska-Cecot K. (2017b), *Preparation of Future Teachers for the Implementation of Programming Science in Elementary Education*, "Education – Technology – Information Technology", 21 (3), pp. 145–150, www.eti.rzeszow.pl/docs/ETI_7_3.pdf (access date: 2.01.2021).

Myśliwiec K. (2017), *Programming in Early Childhood Education*, "Education – Technology – Information Technology", 20 (2), pp. 51–55, www.eti.rzeszow.pl/docs/ETI_7_2.pdf (access date: 2.01.2021).

Plebańska M., Trojańska K. (2018), *Steam Lessons,* Warsaw: e-Litera Publishing House.

Rostański M., Borczyk W., Lipka R. (2018), *Logic, Algorithmic, Programming, At the Early School Stage on the Example of Game Code Lab,* Toruń: Nicolaus Copernicus University in Toruń, https://kometa.edu.pl/uploads/publication/169/1d86_A_111.pdf (access date: 2.01.2021).

Regulation of the Minister of National Education of February 14, 2017 on the core curriculum for preschool education and the core curriculum for general education for primary schools, including students with moderate or severe intellectual disabilities, general education for the first-level vocational school, general education for a special school preparing for work and general education for post-secondary schools (Journal of Laws 2017, item 356, p. 27).

Swić A. (2017), *Coding on the Carpet.* ICT in Education, June, pp. 60–65.

Taniewicz B. (2017), *Ozobots' space journey,* ICT in Education, December, pp. 34–35.

Wing J. (2006), *Computational Thinking,* "CACM", 3 (49), pp. 33–36.

Zasoński S. (2017a), *Build, Program and Have Fun with JIMU Robot,* ICT in Education, June, pp. 32–35.

Zasoński S. (2017b), *STEM Basics with littleBits,* ICT in Education, September, pp. 28–30.

https://encyklopedia.pwn.pl/ (access date: 2.01.2021).

https://edu-sense.com/pl/ (access date: 2.01.2021).

SUMMARY

This article is aimed mainly at students or novice teachers of early childhood education and is an introduction to the problem of developing programming skills among children at earlier school age. The text contains an overview of programming and coding issues and indicates examples of tools supporting the implementation of the core curriculum objective related to programming and coding in early childhood education.

KEYWORDS: early education, information technology, programming, coding, robots in education

STRESZCZENIE

Artykuł skierowany jest głównie do uczniów lub początkujących nauczycieli edukacji wczesnoszkolnej i stanowi wprowadzenie do problematyki rozwijania umiejętności programowania wśród dzieci w młodszym wieku szkolnym. Tekst zawiera przegląd problematyki programowania i kodowania oraz wskazuje przykłady narzędzi wspierających realizację celu podstawy programowej związanego z programowaniem i kodowaniem w edukacji wczesnoszkolnej.

SŁOWA KLUCZOWE: edukacja wczesnoszkolna, informatyka, programowanie, kodowanie, robotyka w edukacji

ADAM NARUSZEWICZ – Uniwersytet w Białymstoku

Pedagogika / Pedagogy
Przysłano do redakcji / Received: 10.03.2021
Daty recenzji / Revised: 27.09.2021; 4.10.2021
Data akceptacji do publikacji / Accepted: 15.11.2021