Acta Universitatis Nicolai Copernici • Pedagogika XLI/1/2021
Nauki Humanistyczno-Społeczne • Zeszyt 453

DOI: http://dx.doi.org/10.12775/AUNC_PED.2021.003

Małgorzata Skibińska
Uniwersytet Mikołaja Kopernika w Toruniu
ORCID: 0000-0001-8972-7529

Joanna Zacniewska
Akademia Marynarki Wojennej
ORCID: 0000-0001-9858-0768

Rozwijanie myślenia komputacyjnego u dzieci wczesnej edukacji

Developing Computational Thinking in Children of Early Childhood Education

Streszczenie

Wszechobecność technologii cyfrowych w życiu codziennym zasadniczo zmienia sposób, w jaki jednostki uzyskują dostęp do wiedzy i ją wykorzystują. Osoby te muszą przetwarzać złożone informacje, myśleć systematycznie i podejmować decyzje w oparciu o różne formy danych. Co ważniejsze, aby wykorzystać nowe możliwości, jakie otwierają technologie cyfrowe w wielu dziedzinach, należy rozwinąć odpowiedni zestaw umiejętności, aby efektywnie korzystać z tych technologii. Obok alfabetyzacji i biegłości w posługiwaniu się IT potrzebne są zdolności intelektualne, tj. myślenie komputacyjne. Celem tego artykułu jest przedstawienie możliwości rozwijania myślenia komputacyjnego u najmłodszych uczniów. Zawiera on przegląd podstawowych pojęć związanych z zagadnieniem myślenia komputacyjnego oraz wybranych sposobów rozwijania tego myślenia na początkowym etapie kształcenia.

Słowa kluczowe: myślenie komputacyjne, kodowanie, programowanie, robotyka

Abstract

The pervasiveness of digital technologies in daily life is fundamentally changing the way individuals access and elaborate knowledge. Individuals have to process complex information, think systematically and take decisions weighting different forms of data. More fundamentally, in order to seize the new opportunities that digital technologies are opening in many areas, individuals have to develop the right set of skills to use these technologies effectively. In addition to literacy and fluency with IT, intellectual abilities (i.e. computational thinking), are needed. The aim of this article is to provide an overview of the opportunities for developing computational thinking in the youngest learners. It includes an overview of the basic terms related to the issue of computational thinking and an outline of the possibilities of developing this thinking at the initial stage of education.

Keywords: computational thinking, coding, programming, robotics

Wstęp

Zgodnie z zapisami aktualnej podstawy programowej kształcenia ogólnego dla szkoły podstawowej[1] jednym z elementów powszechnego kształcenia jest umiejętność programowania, które zakłada rozwój myślenia komputacyjnego. Umiejętność ta ma uwzględniać zrównoważony rozwój ucznia i nie jest rozumiana jedynie jako zdolność do napisania programu w formalnym języku programowania, lecz szerzej, jako doskonalenie sprawności i zaangażowania w działania mające na celu rozwiązanie problemu. Może być ona realizowana zarówno za pomocą narzędzi informatycznych (tj. formalne i wizualne języki programowania, aplikacje użytkowe), jak i bez ich użycia.

[1] *Podstawa programowa kształcenia ogólnego z komentarzem. Szkoła podstawowa, informatyka*, https://www.ore.edu.pl/wp-content/uploads/2017/05/informatyka.-pp-z-komentarzem.-szkola-podstawowa-1.pdf (dostęp: 2021.04.22).

Istota myślenia komputacyjnego

Idea myślenia komputacyjnego jest rozwijana od lat 50. i 60. XX wieku w postaci myślenia algorytmicznego. Algorytmy leżą u podłoża najbardziej podstawowych zadań, w jakie angażuje się każdy, od wykonania prostego przepisu kulinarnego po realizowanie różnych zasad i wskazówek, tj. przestrzeganie zasad ruchu drogowego, programowanie i korzystanie z urządzeń gospodarstwa domowego zgodnie z instrukcją obsługi czy wypełnianie zeznania podatkowego w oparciu o wytyczne formularza.

W 1980 roku Seymour Papert, matematyk, uznawany za prekursora stosowania komputerów w edukacji, po raz pierwszy przywołał pojęcie myślenia komputacyjnego, a w 1996 roku użył tego terminu w odniesieniu do kształcenia matematycznego, wskazując na korzystny wpływ intuicji i sprawności obliczeniowych na kompetencje matematyczne.[2]

Popularyzatorka myślenia komputacyjnego Jeannette Wing wraz ze współpracownikami zdefiniowała tę aktywność umysłową jako:

> „procesy myślowe zaangażowane w formułowanie problemów i ich rozwiązań w taki sposób, że rozwiązania są przedstawiane w formie, która może być skuteczna”[3].

Rozwiązanie takiego problemu może być wykonane przez człowieka, maszynę lub przez połączenie ludzi i maszyn. Wing wskazuje także, że myślenie komputacyjne polega na

> „[...] rozbiciu trudnego problemu na mniejsze, bardziej znane, które można rozwiązać (dekompozycja problemu) używając zestawu reguł do znale-

---

[2] M. M. Sysło. *Na ratunek uczącym się matematyki w szkołach. Jak moglibyśmy się uczyć1, III*. „Przegląd Pedagogiczny”, 2019 nr 1, s. 279.

[3] J. Cuny, L. Snyder, J. M. Wing. *Demystifying computational thinking for non-computer scientists*, nieopublikowany tekst. 2010 za: J.M. Wing, *Computational Thinking: What and Why?,* http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf. s. 1. 2010 (dostęp: 2021.01.30).

zienia rozwiązań (algorytmów) i abstrakcji do uogólnienia tych rozwiązań na podobne problemy”[4].

Zatem myślenie komputacyjne pokrywa się z myśleniem logicznym i systemowym. Obejmuje myślenie algorytmiczne i myślenie równoległe, które angażują takie procesy myślowe, jak rozumowanie kompozycyjne, dopasowywanie wzorców, myślenie proceduralne i myślenie rekurencyjne[5]. W procesie rozwiązywania skomplikowanych problemów wykorzystuje się kolejno cztery kluczowe techniki myślenia komputacyjnego tj.: dekompozycję, analizę (rozpoznawanie wzorców), abstrahowanie (abstrakcja) i tworzenie algorytmu.

Dekompozycja polega na formułowaniu problemu i rozłożeniu go na łatwiejsze do opanowania lub znane elementy. Analiza obejmuje szukanie podobieństw – między problemami i w ich obrębie – umożliwiających identyfikację prawidłowości specyficznych dla danego problemu. Abstrahowanie wymaga skupienia się tylko na ważnych informacjach i eliminowaniu nieistotnych szczegółów. Tworzenie algorytmu, jako końcowy etap myślenia komputacyjnego, sprowadza się do opracowywania „krok po kroku” rozwiązania problemu lub reguł, według których należy postępować, aby rozwiązać dany problem. Ten etap wymaga także weryfikacji i testowania rozwiązania.

Etapowość myślenia komputacyjnego jest bliska założeniom pedagogicznym Johna Deweya, powszechnie uznawanego za prekursora współczesnego ruchu edukacyjnego na rzecz nauczania myślenia krytycznego. Dewey uznawał kształcenie umiejętności myślenia – które polega na przekształcaniu naturalnych zdolności wnioskowania, analizy i interpretacji w nawyk krytycznego dociekania, które skutkuje po-

---

[4] Za: A. Yadav, C. Stephenson, H. Hong. *Computational Thinking for Teacher Education*. “Communications of the ACM”, Volume 60, Issue 4, April 2017, s. 57.

[5] J. M. Wing. *Research Notebook: Computational thinking -what and why?*. “The Link” The magazine of Carnegie Mellon University’s School of Computer Science Magazine, https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why (dostęp: 23.01.2021 r.).

glądami spoczywającymi na solidnym fundamencie przesłanek[6] – za podstawowy cel i zadanie szkoły. Konceptualizacja myślenia w ujęciu Deweya zawiera szczegółową analizę procesu dociekania. Wyróżnił on podstawowe etapy tego procesu, tj.:

> „1) odczucie trudności, 2) wykrycie jej i określenie, 3) nasuwanie się możliwego rozwiązania, 4) wyprowadzenie przez rozumowanie wniosków z przypuszczalnego rozwiązania, 5) dalsze obserwacje i eksperymenty, prowadzące do przyjęcia lub odrzucenia przypuszczenia, czyli do wniosku zawierającego przeświadczenie pozytywne lub negatywne"[7].

Według Deweya stosowanie procedur logicznego rozumowania oraz metod naukowego dociekania jest podstawą wykształconego myślenia. Takie podejście pedagogiczne skłania do refleksji, by rozwijanie myślenia komputacyjnego u młodszych dzieci opierało się na uwrażliwianiu ich na problemy oraz poznawaniu i doskonaleniu metod przystępowania do nich i ich rozwiązywania. Jako takie, nie powinno być ograniczone wyłącznie do lekcji informatyki, lecz powinno być realizowane w całym programie kształcenia, by przygotować uczniów do rozpoznawania problemów, ich rozwiązywania oraz komunikowania się i współpracy z innymi.

Jak już wspomniano, praca z problemem odbywa się etapowo. Pierwszy etap – związany z określeniem problemu (tj. wyszczególnieniem danych i oczekiwanych rezultatów), potrzebnych pojęć, tworzeniem modeli i odkrywaniem rozwiązania – można realizować bez użycia komputera lub innego urządzenia elektronicznego. To czas przeznaczony na myślenie koncepcyjne, symulacje problemu z wykorzystaniem różnych metod i obiektów, przebiegający najczęściej w formie zabawy i sprzyjający samodzielnemu odkrywaniu algorytmów. Po tych działaniach może nastąpić etap drugi – zaprogramowania i testowania rozwiązania za pomocą urządzenia – podczas którego wykorzystywana może być umiejętność pracy z komputerem. Takie podejście nie tylko

---

[6] E. Wasilewska-Kamińska, *Kształcenie myślenia krytycznego – perspektywa historyczna*, w: taż, *Myślenie krytyczne jako cel kształcenia. Na przykładzie systemów edukacyjnych USA i Kanady*, Warszawa 2016, s. 22.

[7] J. Dewey, *Jak myślimy?*, Warszawa 2002, s. 77.

rozwiązuje instytucjonalny problem dostępu do sprzętu komputerowego, ale także ogranicza nadmierną ilość czasu spędzanego przy komputerze przez dzieci wczesnej edukacji i stwarza warunki do wykorzystania technologii tylko w uzasadnionych przypadkach. Czas poświęcony na poszczególne etapy może być zmienny, zależny od poziomu kompetencji cyfrowych uczniów, w tym umiejętności programowania, oraz ich wieku. Ze względu na wymienione czynniki większy nacisk może być kładziony na pierwszy lub drugi etap rozwiązywania problemu[8].

Chociaż myślenie komputacyjne ma swoje korzenie w informatyce, nie powinno się go utożsamiać z kształceniem informatycznym i programowaniem komputerów. Programowanie dostarcza bowiem komputerowi instrukcje na temat celu i sposobu działania, natomiast myślenie komputacyjne pozwala określić człowiekowi cel działania, zaplanować jego szczegóły oraz wymyślić sposób, by komputer mógł to działanie wykonać. Programowanie to przede wszystkim szukanie i wyznaczanie drogi pozwalającej dotrzeć do wybranego celu. Bazuje ono na świadomym sposobie myślenia, które pomaga w bardziej prosty i skuteczny sposób rozwiązywać problemy[9]. W tym ujęciu programowanie będzie kształtować umiejętności, tj.: logiczne, abstrakcyjne i algorytmiczne myślenie, precyzyjne prezentowanie myśli i pomysłów. Będzie także sprzyjać dobrej organizacji pracy, budowaniu kompetencji potrzebnych do pracy zespołowej i efektywnej realizacji projektów.

Na bazie tych rozważań można postawić pytanie: Dlaczego warto rozwijać myślenie komputacyjne? W dobie tak szybkich zmian nauka informatyki wydaje się być naturalnym procesem służącym nadążaniu za postępem technologicznym. Dynamika rozwoju cywilizacyjnego wymusza stałą konieczność adaptacji do zmiennych warunków życia i pracy. Atutem staje się umiejętność wykorzystywania informacji zawartych w otoczeniu, umiejętność radzenia sobie z nowością i zło-

---

[8] *Podstawa programowa* …. dz. cyt., s. 21.

[9] J. Siegmund, C. Kästner, S. Apel, Ch. Parnin, A. Bethmann, T. Leich, G. Saake, A. Brechmann, *Understanding Understanding Source Code with Functional Magnetic Resonance Imaging*, w: *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*, Association for Computing Machinery, New York, NY, USA, 378–389.

żonością codziennych sytuacji, a także łatwość zmiany i dostosowania strategii działania do wymogów określonej sytuacji problemowej. Dlatego nie należy uczyć się gotowych wzorów zachowań i rozwiązań, a raczej wypracować metodykę myślenia problemowego w oparciu o umiejętność analizowania problemu, testowania rozwiązań i wyciągania wniosków. Nie chodzi o to, aby wykształcić pokolenie informatyków, ale o to, aby nauczyć sposobu myślenia, który pozwoli realizować własne potrzeby i osiągnąć sukces w różnych dziedzinach życia społecznego. Popełnianie błędów nie powinno być demotywujące, lecz powinno uświadamiać, że rozwiązywanie problemów jest procesem poszukiwania i testowania rozwiązań[10].

Etapowe rozwiązywanie problemu sprzyja kształtowaniu wytrwałości w dążeniu do celu, opanowaniu w oczekiwaniu na efekty i świadomości konieczności ponoszenia wysiłku dla oczekiwanego rezultatu. W ten sposób wyrabia się cierpliwość oraz wiarę we własne siły w trakcie pokonywania trudności. Dzieci w młodszym wieku poznając język technologii, zastanawiają się nad jej ograniczeniami i przyjmują wobec niej krytyczną postawę. Zapobiegać to może niepożądanej sytuacji, gdy to maszyna przejmuje kontrolę nad zachowaniem młodego człowieka. Dzięki myśleniu komputacyjnemu kształtują się również postawy prospołeczne. Aby rozwiązać problem, dzieci często pracują w grupach lub zespole, co wymaga współpracy, sprawnej komunikacji i myślenia o drugim człowieku.

Znaczenie kształcenia tego rodzaju myślenia zostało także odnotowane w raporcie Narodowej Rady ds. Badań (USA) poświęconemu pedagogicznym aspektom myślenia komputacyjnego, w którym podkreślono, że jest to umiejętność poznawcza, jaką powinien posiadać przeciętny człowiek[11], bowiem wspiera ona uczenie się, rozumienie i kreatywność.

---

[10] Por. D. Powałowska, *Na czym polega myślenie komutacyjne?*, serwis AKCES Edukacja, https://akcesedukacja.pl/baza-wiedzy/blog/na-czym-polega-mysleni-komputacyjne (dostęp: 21.01.2021).

[11] National Research Council, *Report of a Workshop on the Pedagogical Aspects of Computational Thinking,* Washington 2011.

Kodowanie i programowanie bez narzędzi informatycznych na rzecz rozwoju myślenia komputacyjnego dzieci wczesnej edukacji

Na etapie edukacji wczesnoszkolnej rozwój myślenia komputacyjnego będzie wiązał się z kształtowaniem podczas zabaw dydaktycznych określonych umiejętności i postaw, w tym: logicznego myślenia, myślenia algorytmicznego, dostrzegania i rozwiązywania problemów, komunikowania się, współpracy w grupie oraz kreatywności. Na początek nauczyciel powinien więc planować zabawy i gry edukacyjne bez użycia narzędzi informatycznych, a dopiero w kolejnych krokach sięgać po aplikacje i programy komputerowe dostosowane do poziomu rozwoju dzieci.

Przy wyborze i tworzeniu zadań rozwijających myślenie algorytmiczne oraz umiejętność kodowania i programowania należy kierować się wykorzystaniem pojęć i umiejętności kluczowych dla efektywnego posługiwania się technikami myślenia komputacyjnego (dekompozycja, analiza (rozpoznawanie wzorców), abstrahowanie (abstrakcja) i tworzenie algorytmu). Nauczyciel powinien planować aktywności dzieci mające na celu ćwiczenie:

- rozpoznawania i nazywania problemów, zadawania odpowiednich pytań;
- zbierania, porządkowania danych, dzielenia zadań na mniejsze;
- tworzenia zbiorów, rozpoznawania podobieństw, znajdowania istotnych i nieistotnych różnic, uogólniania;
- usuwania zbędnych informacji, upraszczania, tworzenia modeli;
- ustalania kolejnych kroków i tworzenia zasad, sekwencji, rekurencji (powtarzalności procedur i czynności);
- wykrywania i analizowania błędów;
- formułowania zrozumiałych komunikatów, dostosowanych do odbiorcy (komputera lub innych ludzi), kodowania (używając ustalonych symboli i znaków);
- oceniania rozwiązań – rozpoznawania kryteriów wartościowania, określania priorytetów, oceniania prototypów i rozwiązań;

- wyciągania wniosków, rozpoznawania błędów logicznych, argumentowania.

We wczesnych etapach edukacji można zrezygnować z narzędzi informatycznych i na bazie materiałów dydaktycznych rozwijać logiczne myślenie różnymi metodami. Pedagog w procesie nauczania powinien stwarzać odpowiednie warunki, przestrzeń i pomoce dydaktyczne oraz aranżować zadania w taki sposób, by dzieci mogły z jego dyskretnym wsparciem odkrywać nowe zależności otaczającego je świata i aktywnie uczestniczyć w rozwiązywaniu problemów. Zadania mogą polegać na kodowaniu i odkodowaniu według klucza, porządkowaniu elementów, rozpoznawaniu i uzupełnianiu wzoru, wskazywaniu fałszu, rozkładaniu kodu na kawałki lub jego upraszczaniu przez kodowanie powtarzanych czynności, ustalaniu i wykonywaniu warunków jeśli-wtedy itp. Doskonale sprawdzają się tutaj zadania z klockami (np. klocki Dienesa), kodowanymi labiryntami, szyfrowaniem tekstu, graficznymi dyktandami (kodowanie pikselami), prostymi nonogramami (obrazkami logicznymi). Atrakcyjne są też maty do kodowania czy gra w szachy. Problemy, które rozwiązują dzieci, mogą polegać na wypełnianiu krzyżówek, sudoku, diagramów, planowaniu algorytmu przejścia przez labirynt czy rozkładaniu na kolejne kroki codziennych czynności lub aktywności[12]. Na rynku można znaleźć także gry planszowe, np. Robot Turtles oraz karty do gry Littlecodr, które pomogą wprowadzić elementy algorytmiki do zabawy z dziećmi nawet od trzeciego roku życia. Odpowiednio przygotowani pedagodzy już w przedszkolu uczą dzieci podstaw kodowania, korzystając z rozwiązań STEAMowych[13], zestawów Lego Education Coding Express, Magicznego

[12] A. Przytomska-Pietrzak, *Programowanie czas zacząć!*, „Biuletyn Nauczycieli Bibliotekarzy", 2017 nr 6, s. 64–72, http://bnb.oeiizk.waw.pl/6-2017/12_przytomska-pietrzak.pdf (dostęp 12.06.2020).

[13] STEAM łączy pięć dyscyplin: nauki ścisłe (Science), technologię (Technology), inżynierię (Engineering), sztukę (Arts) i matematykę (Mathematics), aby stworzyć środowisko uczenia się, które zachęca wszystkich uczniów do poszukiwania i odkrywania nowych i kreatywnych sposobów rozwiązywania problemów, prezentowania danych, wprowadzania innowacji i łączenia wielu dyscyplin.

Dywanu Funtronic, a nawet 6-latkom pozwalają na kreatywność podczas zajęć z robotyki[14].

Istotna jest atrakcyjność zadania, angażująca ucznia w jego wykonanie, a także zespołowy i prospołeczny charakter zajęć. W trakcie rozwiązywania problemu dzieci powinny być stroną aktywną. Powodzenie zajęć zależy od kreatywności nauczyciela oraz jego merytorycznego i metodycznego przygotowania. W Internecie dostępnych jest wiele materiałów wspomagających pracę nauczyciela w tym zakresie. Sieciowe materiały zawierają pomysły na realizację lekcji, scenariusze zajęć oraz pomoce dydaktyczne wykorzystywane podczas lekcji. Godne polecenia są takie strony internetowe i inicjatywy jak: Zaprogramuj Przyszłość (https://www.zaprogramujprzyszlosc.edu.pl), Uczymy Dzieci Programować (https://uczymydzieciprogramowac.pl), #SuperKoderzy (https://superkoderzy.pl) czy Koduj z klasą (https://kodujzklasa.ceo.org.pl). Warto także zajrzeć do serwisu Koduj.gov.pl, w którym zapoznać się można z materiałami wspomagającymi prowadzenie zajęć z kodowania i programowania offline (https://www.gov.pl/web/koduj) oraz przeglądem gier i zabaw z kodowaniem (https://www.gov.pl/web/koduj/gry-i-zabawy-offline).

W odniesieniu do dzieci wczesnej edukacji kodowanie nie musi oznaczać programowania z wykorzystaniem narzędzi informatycznych. Nie musi też ograniczać się do informatyki czy robotyki. Jednym z najlepszych sposobów na rozpoczęcie programowania jest nauczenie się kodowania „bez prądu". Dzięki temu można łatwo skupić się na podstawowych koncepcjach kodowania, które są fundamentalne w kształceniu tej umiejętności. To świetny sposób, aby poprzez zabawę i grywalizację zaangażować uczniów w programowanie.

Programowanie, jak już wspomniano, to sposób, w jaki człowiek komunikuje się za pomocą poleceń „krok po kroku". Te polecenia tworzą zestaw instrukcji do wykonania i osiągnięcia określonego wyniku. Kodowanie można także zdefiniować jako czynność zapisywania informacji za pomocą kodu, czyli systemu znaków, kolorów lub dźwięków, którym przypisano określone znaczenia zrozumiałe dla podmiotu

[14] A. Przytomska-Pietrzak, op. cit.

przetwarzającego informację. Wystarczy zatem, że w procesie przetwarzania informacji pojawia się myślenie algorytmiczne lub jego komponenty, które skłaniają do zastąpienia, zamiany, odszukania, skojarzenia, przyporządkowania, klasyfikowania, uogólniania, zapamiętania czy zastosowania/użycia inaczej określonych danych, a już można mówić o kodowaniu lub programowaniu offline (czy inaczej: na dywanie, unplugged, bez prądu, bez zasilania).

W dziecięcym programowaniu bez komputera ważne jest przekształcanie lub zastępowanie danych. Czynności te mają miejsce podczas np.:

- zastępowania znaku, słowa lub wyrażenia innym znakiem, słowem lub wyrażeniem;
- zastępowania znaku, słowa lub wyrażenia gestem lub ruchem;
- zastępowania znaku, słowa lub wyrażenia kolorem lub obrazem;
- zastępowania znaku, słowa lub wyrażenia dźwiękiem;
- przypisywania symbolom lub przedmiotom określonego znaczenia.

Korzyści wynikające z kodowania i programowania offline są dostrzegane zarówno przez rodziców, jak i nauczycieli, którzy akcentują[15]:

- rozwijanie logicznego myślenia,
- rozwijanie kompetencji społecznych i matematycznych,
- pobudzanie wyobraźni i kreatywności,
- korzystny wpływ na koordynację ruchową i lateralizację,
- rozwijanie kompetencji miękkich – dzieci wykonując wspólne zadania na macie (w grupie lub w parach) uczą się współpracy.

Programowanie offline to także sposób na wstępną naukę myślenia programistycznego z użyciem przedmiotów, które znajdują się w każdym domu czy szkole. Kartki papieru, kolorowe flamastry czy kostka do gry – tylko tyle potrzeba, aby pokój czy klasa zmieniły się w prawdziwą pracownię programistyczną. Programowania offline można

[15] A. Buchner, M. Kisilowska, M. Wierzbicka, *Mistrzowie Kodowania Junior. Raport z badań*, Centrum cyfrowe, Warszawa 2016, https://centrumcyfrowe.pl/wp-content/uploads/2016/03/Mistzowie-Kodowania-Junior-raport-ko%C5%84cowy.pdf (dostęp: 28.01.2021).

uczyć dzieci w każdym wieku. Można wspólnie ćwiczyć w domu, podczas podróży, a także w otwartej przestrzeni[16].

Programowanie z użyciem narzędzi informatycznych

Twierdzenie, że kultura komputerowa może kształtować myślenie dzieci, stając się obiektem ich myślenia, ma historyczne początki w pracach Seymoura Paperta, ucznia Jeana Piageta, prekursora konstruktywizmu rozwojowo-poznawczego. Papert (twórca języka programowania Logo) twierdził, że dzieci komunikując się z komputerem, programując go, będą zmagały się z potężnymi ideami i w ten sposób, próbując „nauczyć" czegoś maszyny same będą się uczyć. Tę myśl kontynuował uczeń Paperta, Mitchel Resnick (kierownik grupy badawczej, która opracowała wizualny język programowania o nazwie Scratch), głosząc ideę, że

> „myślenie komputacyjne to coś więcej niż programowanie, w ten sam sposób, w jaki umiejętność posługiwania się językiem jest czymś więcej niż pisanie. Oba są bardzo ważne. [...] Programowanie, podobnie jak pisanie, jest środkiem wyrazu i punktem wyjścia do rozwijania nowych sposobów myślenia"[17].

Obecnie dostępnych jest wiele narzędzi, które nie wymagają specjalistycznej wiedzy, a dzięki intuicyjnemu interfejsowi wizualnemu umożliwiają naukę programowania i kodowania poprzez zabawę. Dzięki temu uczniowie nabierają wprawy w programowaniu komputera oraz pewności, że mogą skutecznie się z nim komunikować, wykorzystując go do rozwiązywania złożonych problemów.

---

[16] *Zabawy w programowanie offline – publikacja*, serwis Koduj.gov.pl, https://www.gov.pl/web/koduj/zabawy-w-programowanie-offline--publikacja (dostęp: 25.01.2021).

[17] National Research Council, *Committee for the Workshops on Computational Thinking: Report of a workshop on the scope and nature of computational thinking*. Washington, 2010, s. 13.

Nauka programowania kształci także wyobraźnię dziecka i dostarcza narzędzi do twórczego stosowania technologii, po którą będzie sięgać, by realizować własne pomysły czy wyrażać siebie.

W obszarze badań nad wpływem programowania na procesy poznawcze uczniów uważa się, że zaangażowanie w programowanie umożliwia specyficzne uczenie się. Programowanie opiera się bowiem na zdolnościach metapoznawczych wykorzystywanych w myśleniu komputacyjnym, m.in. na łączeniu nowych informacji z wcześniejszą wiedzą, świadomym wyborze strategii myślenia, planowaniu, monitorowaniu i ocenie procesów myślowych, rozkładaniu złożonych działań na sekwencje warunkowe. Te zdolności metapoznawcze z kolei opierają się na funkcjach wykonawczych wykorzystujących pamięć roboczą, uwagę i kontrolę hamowania do planowania i wykonania działania ukierunkowanego na cel[18]. Zwraca się także uwagę, że efektywność takiego uczenia się wzrasta wraz poziomem zaangażowania w sytuację problemową i rozwojem zdolności poznawczych dzieci. W tym kontekście ważnym aspektem pedagogicznym jest właściwy dobór metod, środków dydaktycznych i poziomu trudności zadań problemowych, które rozwiązywać będą uczniowie. Z czasem, oprócz logicznego i kreatywnego myślenia, nauczą się oni także konsekwentnego dążenia do celu, ponieważ tworząc kod, trzeba stale przestrzegać pewnych zasad. Ćwiczyć będą również cierpliwość, którą będą musieli wykazać się chociażby w momencie szukania przyczyny błędu w kodzie programu. A co najważniejsze, świadomość uczniów, że samodzielnie stworzyli coś, co działa, pozwoli im uzyskać większą pewność siebie i wiarę we własne możliwości, co bez wątpienia znacznie ułatwi im wkroczenie w dorosłe życie[19]. Zatem nawet jeśli w przyszłości niewielu obecnych uczniów zostanie programistami, to nauka programowania sprawi, że zdobędą oni cenne kompetencje, które przydadzą im się w dorosłym życiu.

---

[18] Por. K. DePryck, *From computational thinking to coding and back*, w: *TEEM '16: Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, red. F. J. García-Peñalvo, New York 2016, s. 28.

[19] K. Rojewska. *Nauka programowania dla dzieci – gry edukacyjne*. No Fluff Jobs Blog, https://nofluffjobs.com/blog/nauka-programowania-dla-dzieci (dostęp: 29.01.2021).

Do nauki kodowania i programowania dostępnych jest wiele cyfrowych narzędzi angażujących sferę kognitywną i wolicjonalną i realizujących zasadę „uczyć – bawiąc":

- Kodable (https://www.kodable.com) to atrakcyjne narzędzie do nauki kodowania dla dzieci, a także wyjątkowe źródło pomocy dla nauczycieli w nauce kodowania. Kodable wprowadza dzieci w kluczowe pojęcia z zakresu programowania komputerowego poprzez serię gier zaprojektowanych tak, aby ich poziom trudności rósł wraz z rozwojem ucznia. Gry są osadzone w przestrzeni kosmicznej, a uczniowie przechodzą przez kolejne poziomy, ucząc się pojęć programistycznych w trakcie zabawy. Umiejętności zaczynają się od sekwencjonowania i przechodzą do pętli, warunków, funkcji, zmiennych i koncepcji programowania zorientowanego obiektowo, takich jak właściwości i klasy. Niestety darmowa wersja zawiera ograniczoną ilość materiałów do gry.
- Code.org (https://code.org) oferuje darmowe programy nauczania dla wszystkich poziomów nauczania informatyki i programowania. Materiały wideo pomagają dzieciom zrozumieć znaczenie programowania w dzisiejszym świecie. Dodatkowo, w ramach inicjatywy Hour of Code (Godzina kodowania (https://hourofcode.com/pl/learn)) dostępne są krótkie samouczki, które mają na celu zainteresowanie programowaniem. Uczniowie oglądają instrukcje wideo prowadzone przez znanych programistów, a następnie używają bloków kodu do programowania mini-gier z niektórymi znanymi postaciami z Minecrafta, Disneya i popularnych aplikacji do gier.
- Code for Life (https://www.codeforlife.education) jest stroną zaprojektowaną do nauki kodowania dla uczniów od szkoły podstawowej do liceum. Zaczynając od opartego na blokach języka kodowania i prostych łamigłówek, stopniowo przechodzi się aż do języka programowania Python. Każdy poziom zawiera szczegółowe scenariusze lekcji dla nauczycieli, którzy chcą lub potrzebują wsparcia w nauczaniu kodowania. Lekcje są dobrze przemyślane i łatwe do naśladowania.
- MicrosoftMakeCode (https://www.microsoft.com/en-us/makecode) oferuje coś dla niemal każdego ucznia pokazując różno-

rodne zastosowania kodu (tzn. nie służy on tylko do tworzenia aplikacji). MakeCode zawiera szeroki zakres wskazówek, samouczków i projektów dla nauczycieli rozpoczynających naukę kodowania oraz dla uczniów uczących się samodzielnie.

- Scratch (https://scratch.mit.edu) jest wersją wizualnego języka kodowania opartego na blokach, stworzonego przez grupę Lifelong Kindergarten Group z MIT. Można go używać online lub pobrać i stosować bez połączenia z Internetem. Scratch pozwala uczniom poznać i wykorzystać wszystkie istotne elementy kodowania i programowania. Uczniowie przeciągają i upuszczają bloki kodu, aby tworzyć animacje, cyfrowe opowieści, sztuki, proste programy matematyczne itp. Za pomocą Scratcha uczniowie mogą również programować różne urządzenia peryferyjne (takie jak micro:bit) do robotyki, nauki i inżynierii. Ekran Scratcha jest podzielony na trzy części: scenę po prawej stronie (gdzie można zobaczyć wyniki działania kodu), obszar roboczy w centrum (gdzie układa się kod jak puzzle) i paletę bloków po lewej stronie (gdzie znajdują się wszystkie bloki kodu). Uczniowie kodują działania wielu duszków (różnych postaci) lub elementów ekranu, a także mogą dodawać dźwięki, obrazy i elementy tekstowe, aby zbudować prawie wszystko na co mają ochotę.
- ScratchJunior to aplikacja mobilna, dzięki której mali programiści, którzy dopiero uczą się czytać i pisać mogą ożywiać proste i interaktywne sceny poprzez przeciąganie i upuszczanie grafiki na ekran. Przypominające klocki Lego zatrzaskujące się polecenia sprawiają, że podstawowe programy są łatwe do stworzenia i uruchomienia.
- Baltie to program, który można wykorzystać już w pierwszej klasie szkoły podstawowej, ponieważ bazuje na języku ikon, bez konieczności opanowania umiejętności czytania alfabetu. Złożoność prac można stopniować tak, by dzieci kolejno poznawały zasady programowania i mogły się zrealizować na każdym poziomie. Z tego powodu program ma trzy tryby pracy: Scena – dzieci uczą się używać myszki i klawiatury, poznają środowisko Baltiego, banki przedmiotów, mogą samodzielnie tworzyć własne przedmioty w edytorze graficznym; Czarowanie (rozkazywa-

nie) – dziecko uczy się myślenia, dzielenia zadania na mniejsze części – aż do pojedynczych rozkazów (poleceń), które wydaje czarodziejowi Baltiemu, by je natychmiast wykonał; Programowanie – na tym poziomie dzieci zaczynają pracować dopiero po opanowaniu poprzednich dwóch trybów pracy. Uczeń nie wydaje Baltiemu rozkazów interaktywnie, ale buduje z nich ciągi logiczne, czyli pisze program.

- Logo – Logo Komeniusz (nowsza wersja Logomocja) jest programem umożliwiającym naukę programowania poprzez m.in. tworzenie rysunków, melodii, wykonywanie obliczeń, definiowanie pojęć matematycznych, modelowanie i symulowanie różnych procesów (fizycznych, biologicznych) oraz realizowanie multimedialnych projektów. Środowisko programowania składa się z ekranu graficznego, po którym porusza się żółw Logo oraz ekranu tekstowego służącego głównie do wprowadzania poleceń, czyli rozkazów dla żółwia.
- PixBlocks (https://pixblocks.com) to aplikacja, która umożliwia naukę programowania od podstaw przy pomocy bloczków do zaawansowanych zagadnień języka Python. Na stronie dostępne są także materiały dla nauczycieli wspomagające pracę w tym środowisku programowania.
- Blockly Games (https://blockly.games) – seria gier edukacyjnych, które etapami uczą programowania. Przeznaczone są dla dzieci, które nie miały wcześniejszego doświadczenia z programowaniem.
- Scottie Go! (https://scottiego.com) to innowacyjne połączenie gry planszowej i gry mobilnej, które uczy programowania i rozwiązywania problemów. Jej celem jest nauka programowania.
- Code Monkey – to edukacyjne środowisko oparte na grach, w którym dzieci uczą się kodować bez żadnego wcześniejszego doświadczenia. Gra podzielona na kilka poziomów, a w każdym z nich dziecko musi przepisać kod, który umożliwi małpce dotarcie do banana.
- Godzina Kodu z NCLab! (https://hoc.nclab.com/karel/pl) to kolejne środowisko oparte na grach pozwalające poznać podstawy programowania proceduralnego. Uczniowie piszą skrypty, aby

sterować robotem o imieniu Karel. Pracują z warunkami, pętlami, funkcjami, listami itp.
- Code Combat – gra RPG przeznaczona zarówno dla dzieci, jak i dorosłych. Jest dostępna z poziomu przeglądarki i uczy programowania w znanych językach, takich jak Python czy JavaStript. Dziecko wciela się w niej w bohatera, któremu – za pomocą konkretnych komend – wydaje polecenia, przechodząc kolejne poziomy gry[20].

W sieci dostępnych jest coraz więcej narzędzi wspomagających naukę kodowania i programowania. W serwisie Koduj.gov.pl można zapoznać się z uzupełnieniem powyższego przeglądu gier i aplikacji online dedykowanych kodowaniu i programowaniu.

Robotyka

Robotyka to dziedzina interdyscyplinarna stanowiąca łącznik między informatyką a innymi dyscyplinami, m.in. mechaniką, inżynierią i fizyką. Co więcej, nauka robotyki nie wymaga opanowywania złożonych zagadnień teoretycznych. Uczniowie uczą się podczas zabawy pojedynczych poleceń lub ich sekwencji sterujących robotem lub obiektem na ekranie komputera bądź innego urządzenia cyfrowego[21].

Zagadnienia dotyczące robotyki wprowadzone zostały do szkół w 2017 roku wraz z nową podstawą programową. Nauczyciele wykorzystują narzędzia związane z programowaniem robotów, choć sama robotyka nie ma jeszcze ugruntowanej pozycji w szkole. Zajęcia z robotyki są atrakcyjne dla uczniów, zwłaszcza młodszych, dla których jest to przede wszystkim dobra zabawa. Roboty wraz z oprogramowaniem posiadają wiele zalet i pełnią ważne funkcje w procesie edukacyjnym. Nastawiają uczniów pozytywnie do procesu uczenia się, działają motywująco. Pełnią również funkcję poznawczą, zaznajamiając użytkowników z rzeczywistym działaniem robotów oraz funkcję kształcą-

[20] Por. ibidem.

[21] J. Stańdo, M. Spławska-Murmyło, *Aktywna realizacja zagadnień z zakresu robotyki*. Zestaw 7. Warszawa, 2017, s. 3.

cą, wpływając bezpośrednio na proces poznawczy i rozwój umiejętności uczenia się.

Przed podjęciem decyzji związanej z rozpoczęciem zajęć z robotyki trzeba wziąć pod uwagę wiek, wiedzę i możliwości uczniów, potencjał zestawu do nauki, jego wytrzymałość i całkowity koszt. Każdy sprzęt ma swoje ograniczenia – nie są to idealne rozwiązania i nie wszystkie będą odpowiadać oczekiwaniom i możliwościom nauczycieli. Dostępne na rynku zestawy do robotyki można podzielić na trzy grupy: programowalne roboty – zabawki, zestawy do samodzielnego montażu, aplikacje i rozwiązania wirtualne.

Do zajęć z robotyki można wykorzystać m.in.:

- Ozoboty – małe, sprytne roboty do nauki programowania. Ozobot jest niewielkim i niepozornym robotem, który porusza się samodzielnie. Wystarczy tylko narysować trasę – urządzenie potrafi się poruszać zarówno po zwykłym papierze, jak po ekranie smartfona lub tabletu. Dzięki temu gadżet można wykorzystać do prostych gier i zabaw z dziećmi.
- Roboty programowalne – np. Dash&Dot, Ozobot, Thymio 2, Cozmo, Photon – są one atrakcyjne wizualnie i do pewnego stopnia można programować ich ruch i zachowanie. Są to najczęściej konstrukcje całościowe, odpowiadające potrzebom zwłaszcza najmłodszych uczniów. Pozwalają one dzieciom na oswojenie się z programowalnymi maszynami. Oczywiście część robotów na rynku to bardziej zabawki niż narzędzia do nauki. Dzięki nim uczniowie są w stanie zrozumieć podstawowe polecenia i instrukcje programistyczne, są one jednak czasem ograniczone w kwestii samego programowania. Posiadają one również możliwość rozbudowy, kupna dodatkowych akcesoriów i są bardzo wytrzymałe.
- Zestawy do robotyki – kolejne z dostępnych rozwiązań, zestawy do samodzielnego montażu. Wśród nich największą popularnością cieszą się marki LEGO, VEX, LOFI Robot oraz mBot. Oprócz wielu zalet zestawy te pozwalają na wprowadzenie elementów mechaniki i praw fizyki, a także umożliwiają uczniom rozwija-

nie kreatywności i wyobraźni przestrzennej[22]. Popularna marka Lego, z seriami: WeDo i Mindstorms wykorzystywana jest w nauczaniu robotyki. Dzieci świetnie posługują się tymi klockami, a poprzez części elektroniczne i mechaniczne, rozwijają swoją kreatywność.

Lekcje robotyki w sposób naturalny uczą i inspirują, ponieważ są niezwykle atrakcyjne dla uczniów – szczególnie tych młodszych – i wpływają na poziom ich zaangażowania w wykonywane zadania. Mimo tego lekcje robotyki najczęściej realizuje się w formie zajęć pozalekcyjnych, często płatnych. Wpływa na to dość wysoki koszt robotów do nauki programowania i zestawów do robotyki, na zakup których brakuje środków w budżecie szkoły. Kolejną możliwą przyczyną może być brak przygotowania nauczycieli w zakresie programowania i robotyki[23].

Podsumowanie

Myślenie komputacyjne, umiejętność dekompozycji, wnioskowania, korekty błędów, rozwiązywania problemów, logicznego myślenia oraz możliwości poznawcze i kreatywność rozwijane są poprzez programowanie i kodowanie. Umiejętności te mogą być dla dzisiejszych uczniów, zwłaszcza w młodszym wieku, dużo ważniejsze i bardziej przydatne niż wiedza teoretyczna zdobywana w innych obszarach. Świat zmienia się bowiem tak szybko, że nie można przewidzieć, jakich informacji będą potrzebowały dzieci w swym dorosłym życiu.

Przenikanie technologii komputerowej i telekomunikacyjnej do obiektów i przedmiotów z naszego otoczenia wymaga pewnej spraw-

---

[22] A. Syrocka, *Zestaw LEGO Education WeDo 2.0. Kompleksowa recenzja 2021*. serwis RoboCamp, https://www.robocamp.pl/pl/blog/lego-education-wedo2-recenzja (dostęp: 17.01.2021).

[23] Zob. A. Raczykowska, *Programowanie i robotyka w nowej podstawie programowej wobec (nie)kompetencji nauczycieli informatyki*, „Problemy Profesjologii", 2019 nr 2; K. Majewska, *Trudności w nauczaniu programowania na poziomie edukacji wczesnoszkolnej z perspektywy nauczycieli – absolwentów szkół pedagogicznych*, „E-mentor", 2018 nr 3(75), s. 32–39; B. Kuźmińska-Sołśnia, K. Ziębakowska-Cecot, *Przygotowanie przyszłych nauczycieli do wdrażania nauki programowania w edukacji elementarnej*, „Edukacja – Technika – Informatyka", 2017 nr 3/21.

ności w ich użytkowaniu. Rozumienie tej technologii warunkuje jej efektywne wykorzystywanie w procesie rozwiązywania problemów. Obecnie większość urządzeń wyposażana jest w mikroprocesory i czujniki przetwarzające określone dane.

> „Co za tym idzie, bez względu na to, czy chodzi o prowadzenie biznesu, czy zarządzanie własnym zdrowiem, nasza praca i życie osobiste będą w coraz większym stopniu wymagać interakcji z danymi, dostrzegania w nich pewnych prawidłowości, podejmowania decyzji w oparciu o dane oraz wykorzystywania tych danych do projektowania żądanych rezultatów”[24].

Zatem myślenie komputacyjne nie tylko umożliwi właściwe funkcjonowanie w społeczeństwie cyfrowym, ale także będzie stanowić podstawę przyszłych kwalifikacji zawodowych. Potwierdzają to wyniki raportów badawczych, które wskazują elementy myślenia komputacyjnego, jako pożądane czy niezbędne umiejętności współczesności i przyszłości. W Raporcie Addecco pt. „Kompetencje przyszłości – czwarta rewolucja przemysłowa w Europie Wschodniej”[25] wśród kluczowych umiejętności wskazuje się m.in. kreatywność, elastyczne rozwiązywanie problemów, umiejętność analizowania danych. W raporcie Instytutu Przyszłości (Kompetencje Zawodowe Przyszłości 2020) dostrzeżono wagę niekonwencjonalnego i adaptacyjnego myślenia (twórcze myślenie, kreatywność, elastyczność, dostosowywanie się do zmieniających się warunków), a także myślenia komputacyjnego (rozumowania i tworzenia koncepcji opartych na mnóstwie danych) czy zarządzania informacjami i krytycznego myślenia upatrując ich znaczenia w rzeczywistości 2020 roku[26]. Ten sam instytut badawczy w kolejnym rapor-

---

[24] D. Zatoński, *Raport: Kompetencje Zawodowe Przyszłości 2020 (Institute for the Future)*, serwis Alogic.pl, https://alogic.pl/blog/raport-kompetencje-zawodowe-przyszlosci-2020-institute-for-the-future (dostęp: 1.03.2021).

[25] *Kompetencje przyszłości – czwarta rewolucja przemysłowa w Europie Wschodniej. Raport badawczy*, The Adecco Group, 2018, s. 6, http://www.outsourcingportal.eu/pl/userfiles/image/raporty/2018/Czerwiec/19/Inovantage_June_18th_Adecco_Group.pdf (dostęp: 20.02.2021).

[26] *Future Work Skills 2020. Raport badawczy*, Institute for the Future dla the University of Phoenix Research Institute, 2011, s. 8–12, https://kometa.edu.pl/uplo-

cie „Future of skills. Employment in 2030”[27] wskazuje, że w nadchodzących latach liczyć się będą umiejętności interpersonalne, społeczne, systemowe i poznawcze m.in. współdziałanie, kompleksowe rozwiązywanie problemów, myślenie logiczne i krytyczne, ocena i podejmowanie decyzji, jasne komunikowanie się, programowanie, ocena systemów i procesów, projektowanie technologii, wizualizowanie czy instruowanie. Podobne wnioski można znaleźć w raporcie „OECD Future of Education and Skills 2030”[28].

Wobec powyższego ważne jest, aby celowo i systematycznie kształtować niezbędne umiejętności. Warto więc podjąć trud rozwijania myślenia komputacyjnego już u dzieci wczesnej edukacji, przygotowując je stopniowo do dorosłego życia i wyposażając w narzędzia intelektualne umożliwiające w przyszłości ich skuteczne funkcjonowanie w coraz bardziej zmiennych, niepewnych i złożonych warunkach.

Bibliografia:

Buchner, Anna, Małgorzata Kisilowska, Maria Wierzbicka. *Mistrzowie Kodowania Junior. Raport z badań*. Centrum cyfrowe, Warszawa 2016. https://centrumcyfrowe.pl/wp-content/uploads/2016/03/Mistzowie-Kodowania-Junior-raport-ko%C5%84cowy.pdf.

Cuny, Jan, Lary Snyder, Jeannette Marie Wing. “Demystifying computational thinking for non-computer scientists”, nieopublikowany tekst. 2010 za: J.M. Wing, “Computational Thinking: What and Why?”. http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf.

DePryck, Koen. “From computational thinking to coding and back”. W: *TEEM ‘16: Proceedings of the Fourth International Conference on Technological Eco-*

ads/publication/620/6bfc_AA_SR-1382A_UPRI_future_work_skills_sm.pdf?v2.8 (dostęp: 20.02.2021).

[27] *Rynek pracy, a umiejętności. Wyniki badań*, Pearson Central Europe, http://umiejetnosci2030.pl/ (dostęp: 20.02.2021).

[28] *OECD Future of Education and Skills 2030. Concept Note*, OECD 2019, https://www.oecd.org/education/2030-project/teaching-and-learning/learning/skills/Skills_for_2030_concept_note.pdf (dostęp: 20.02.2021).

systems for Enhancing Multiculturality, red. F. J. García-Peñalvo, 27–29. New York: NY, USA: ACM, 2016. DOI: https://doi.org/10.1145/3012430.3012492.

Dewey, John. *Jak myślimy?*. Warszawa: Altaya Polska i' l De Agostini Polska, 2002.

Future Work Skills 2020. Raport badawczy. Institute for the Future dla the University of Phoenix Research Institute. 2011, https://kometa.edu.pl/uploads/publication/620/6bfc_AA_SR-1382A_UPRI_future_work_skills_sm.pdf?v2.8.

Kompetencje przyszłości – czwarta rewolucja przemysłowa w Europie Wschodniej. Raport badawczy. The Adecco Group, 2018. http://www.outsourcing-portal.eu/pl/userfiles/image/raporty/2018/Czerwiec/19/Inovantage_June_18th_Adecco_Group.pdf.

National Research Council. *Committee for the Workshops on Computational Thinking: Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press, 2010.

National Research Council. Report of a Workshop on the Pedagogical Aspects of Computational Thinking. Washington, DC: The National Academies Press, 2011. DOI: https://doi.org/10.17226/13170.

OECD Future of Education and Skills 2030. Concept Note. OECD 2019. https://www.oecd.org/education/2030-project/teaching-and-learning/learning/skills/Skills_for_2030_concept_note.pdf.

Podstawa programowa kształcenia ogólnego z komentarzem. Szkoła podstawowa. Informatyka. MEN 2017.

Podstawa programowa kształcenia ogólnego z komentarzem. Szkoła podstawowa, informatyka. https://www.ore.edu.pl/wp-content/uploads/2017/05/informatyka.-pp-z-komentarzem.-szkola-podstawowa-1.pdf.

Powałowska, Daria. *Na czym polega myślenie komputacyjne?*. serwis AKCES Edukacja. https://akcesedukacja.pl/baza-wiedzy/blog/na-czym-polega-myslenie-komputacyjne.

Przytomska-Pietrzak, Alicja. „Programowanie czas zacząć!". *Biuletyn Nauczycieli Bibliotekarzy* 2017/6: 64–72. http://bnb.oeiizk.waw.pl/6-2017/12_przytomska-pietrzak.pdf.

Raźniak, Aleksandra. „Zakodowane opowiadania, czyli jak wykorzystywać kodowanie w metodzie narracyjnej". *Języki obce w szkole* 2019/1: 29–34. http://jows.pl/artykuly/zakodowane-opowiadania-czyli-jak-wykorzystywac-kodowanie-w-metodzie-narracyjnej.

Rojewska, Katarzyna. *Nauka programowania dla dzieci – gry edukacyjne*. No Fluff Jobs Blog. https://nofluffjobs.com/blog/nauka-programowania-dla-dzieci.

Rynek pracy, a umiejętności. Wyniki badań. Pearson Central Europe. http://umiejetnosci2030.pl/.

Siegmund Janet, Christian Kästner, Sven Apel, Chris Parnin, Anja Bethmann, Thomas Leich, Gunter Saake, André Brechmann. "Understanding Understanding Source Code with Functional Magnetic Resonance Imaging" W: *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. Association for Computing Machinery, New York, NY, USA, 378–389. DOI: https://doi.org/10.1145/2568225.2568252.

Stańdo, Jacek, Monika Spławska-Murmyło, *Aktywna realizacja zagadnień z zakresu robotyki*. Zestaw 8, Zeszyt 3. Warszawa: ORE, 2017.

Stańdo, Jacek, Monika Spławska-Murmyło, *Aktywna realizacja zagadnień z zakresu robotyki*. Zestaw 7, Zeszyt 3. Warszawa: ORE, 2017.

Syrocka, Aleksandra. *Zestaw LEGO Education WeDo 2.0. Kompleksowa recenzja 2021*. serwis RoboCamp. https://www.robocamp.pl/pl/blog/lego-education-wedo2-recenzja.

Sysło, Maciej M. „Na ratunek uczącym się matematyki w szkołach. Jak moglibyśmy się uczyć1, III". *Przegląd Pedagogiczny* 1/2019: 269–281. DOI: https://doi.org/10.34767/PP.2019.01.19.

Wasilewska-Kamińska, Ewa. „Kształcenie myślenia krytycznego – perspektywa historyczna". W: Taż, *Myślenie krytyczne jako cel kształcenia. Na przykładzie systemów edukacyjnych USA i Kanady*. Warszawa: Uniwersytet Warszawski, 2016. DOI: https://doi.org/10.31338/uw.9788323525639.

Wing, Jeannette Marie. "Research Notebook: Computational thinking – what and why?". *The Link,* The magazine of Carnegie Mellon University's School of Computer Science Magazine. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.

Yadav, Aman, Chris Stephenson, Hai Hong. "Computational Thinking for Teacher Education". *Communications of the ACM*, Volume 60, Issue 4, April 2017: 55–62. DOI: https://doi.org/10.1145/2994591.

Zabawy w programowanie offline – publikacja. serwis Koduj.gov.pl. https://www.gov.pl/web/koduj/zabawy-w-programowanie-offline--publikacja.

Zatoński, Dariusz. *Raport: Kompetencje Zawodowe Przyszłości 2020 (Institute for the Future)*. serwis Alogic.pl. https://alogic.pl/blog/raport-kompetencje-zawodowe-przyszlosci-2020-institute-for-the-future.