

Dlaczego warto uczyć się programowania?



Katarzyna Gagan, nauczyciel konsultant ds.
informatyki
Centrum Edukacji Nauczycieli w Białymstoku
Zdjęcie: www.flickr.com

*„Nauka pisania programów to gimnastyka dla mózgu.
Pozwala wypracować umiejętność efektywnego myślenia
o rzeczach niezwiązanych z informatyką”*

Bill Gates.

Rozwój nowych technologii zapewnia wydajniejsze metody pracy, otwiera nowe perspektywy działalności człowieka. IT dotyka różnych dziedzin współczesnego życia i przenika do zupełnie nowych obszarów. Powoduje wzrost zapotrzebowania na kompetencje związane z programowaniem oraz umiejętność tworzenia nowych rozwiązań za pomocą kodu, jak również zdolności przetwarzania, przesyłania i analizowania danych.[1] *Na rynku pracy „na zawodowy sukces mogą liczyć pracownicy wszechstronni i otwarci na nowoczesne rozwiązania, gotowi wykonywać pracę elastycznie”.* [2] A do kompetencji przyszłości zaliczane są: uczenie się i wykorzystywania wiedzy z wielu dziedzin – interdyscyplinarność, praca w międzynarodowym środowisku, praca w wirtualnych zespołach, praca w „szumie informacyjnym”, programowanie.[3]

Uważa się, że język programowania będzie alfabetem profesjonalistów i podstawową kompetencją XXI wieku. To stwierdzenie odnosi się nie tylko do specjalistów ICT, ale do każdego. Stąd mówi się o potrzebie nauki programowania już w szkołach podstawowych.

Maciej Sysło - prof. Wydział Matematyki i Informatyki Uniwersytetu Mikołaja Kopernika powiedział: *„Dzisiaj wiele zawodów niezwiązanych z ICT wymaga umiejętności programistycznych. Na przykład biolog, badając genotyp, programuje komputer do wykonania tej czynności”.* Programista i bloger Sacha Greif tłumaczy, że programowanie staje się ważną umiejętnością z perspektywy zwykłego człowieka. *„Nie chodzi o to, że trzeba od razu stać się gwiazdą IT. Zyskujesz zdolność tworzenia stron internetowych albo – jeśli musisz w pracy przy komputerze wykonywać powtarzalne i nudne czynności – możliwość napisania programu, który robi to za ciebie”.* Oznacza to, że skonstruowane skrypty mogą wykonywać rutynowe zadania typu: porządkowania dokumentów, plików czy wyszukiwania informacji, zliczanie danych itp.

„Wkrótce może się okazać, że podstawy poziom programowania, może być

wymaganiem standardowym jakim dzisiaj np. jest obsługa pakietu Office”- to słowa dr Łukasza Arendta pracownika Uniwersytetu Łódzkiego .[4]

Współcześni uczniowie - cyfrowi tubylcy – preferują media interaktywne z Internetem (i wyszukiwarką Google) na czele, doskonale się nimi posługują, ale mają problemy z dotarciem do wartościowych informacji i ich interpretacją. Preferują hipertekstowy dostęp do informacji. Oznacza to zmianę sposobu myślenia z dotychczasowego linearnego na wielowątkowy. Są oni przy tym niezdolni do głębszej refleksji, nie potrafią wyciągać ogólnych wniosków czy przyjąć szerszego punktu widzenia. Zapamiętują mnóstwo informacji, ale nie potrafią ich interpretować ani zrobić z nich użytku – w rezultacie coraz mniej ludzi jest zdolnych do tak dziś cenionej kreatywności. [5] Uczniowie „uczą się niesystematycznie, zazwyczaj wtedy, kiedy muszą. Nie czerpią radości z poznawania. Obserwuje się, że mają też kłopoty ze rozumieniem pojęć, wnioskowaniem, planowaniem i podejmowaniem decyzji. Słabo wypadają w testach sprawdzających umiejętność twórczego myślenia.[6] Do ograniczeń, zalicza się też między innymi zanik umiejętności przeprowadzenia logicznego wywodu.

Uczniowie dorastają w dobie cyfryzacji, Sprostanie zmianom społecznym, kulturowym i technicznym, jest dużym wyzwaniem dla szkoły, która musi szukać rozwiązań organizacyjnych, dydaktycznych i dostosować model nauczania do rzeczywistych potrzeb i oczekiwań społecznych.[7] Zadania szkoły powinny uwzględniać nowe kompetencje cyfrowe współczesnego ucznia, rozwijać umiejętności rozumowania, myślenia strategicznego, argumentowania, wnioskowania czy dokonywania interpretacji. Niezmiernie ważne jest także przygotowanie uczniów do rozwiązywania problemów z różnych dziedzin życia/gospodarki/nauki przy świadomym wykorzystaniu technik i metod jakie oferuje informatyka.

Edukacja informatyczna to już nie tylko nauka obsługi pakietu Office. Coraz większy nacisk już w klasach najmłodszych, kładzie się właśnie na umiejętności programistyczne i kształtowanie postaw z nimi związanych. Są to: umiejętność analitycznego i abstrakcyjnego myślenia, planowania, kreatywności, zdolności do realizacji innowacji, optymalizacji działań, skutecznej komunikacji, pracy w zespole. Programowanie to przede wszystkim świetna zabawa, rozwiązywanie zagadek i złożonych problemów. O nauce programowania coraz częściej mówi się jako o działaniu wspierającym współpracę i pracę zespołową. Programowanie nie ogranicza się tylko do pisania skryptów w określonym języku programistycznym, a co za tym idzie nauka programowania nie powinna ograniczać się do nauki danego języka i tłumaczenia algorytmów na ten język. Obejmuje analizę konkretnego problemu, wypracowanie rozwiązania go w postaci algorytmu, dostosowanie algorytmu do możliwości jego wdrożenia (np. parametrów komputera), przetłumaczenie algorytmu na odpowiedni język programistyczny (kodowanie). Bezpośrednio związanymi z programowaniem czynnościami jest testowanie oraz eliminowanie błędów ze skryptu. [8]

Aby programować, należy nauczyć się myśleć abstrakcyjnie, odejść od wcześniej

poznanych schematów i zawsze szukać lepszego (bardziej wydajnego) rozwiązania zgodnie z mottem: *do it better*.^[9]

Rozwiązywanie prostych zadań, zagadek i łamigłówek logiczno-algorytmicznych, układanie obrazków, tekstów, instrukcji w logicznym porządku, tworzenie poleceń dla określonego planu działania, to umiejętności, które uczniowie nabywają już na pierwszym etapie kształcenia. Przykłady i analiza algorytmów z życia codziennego pomagają zrozumieć uczniom instrukcje, algorytmy z powtórzeniami czy algorytmy realizujące warunek. Jest to doskonale przygotowanie do późniejszego konstruowania skryptów w wizualnych językach programowania.

Rozwijanie myślenia algorytmicznego, wspomagane wizualizacją lub symulacją działań algorytmicznych nie powinno być oderwane od innych edukacji pierwszego etapu kształcenia. Taki sposób nauki programowania można realizować wykorzystując np. matę edukacyjną. Klocki dołączone do maty: strzałki, symbole obrotów, skoków, zmiany wielkości, start i stop, świetnie wprowadzają do aplikacji typu: Scratch Junior, Kodable czy Lightbot. Kształtowanie umiejętności logicznego myślenia u uczniów można również wspierać poprzez rozwiązywanie łamigłówek programistycznych jakimi są np. : Konkurs Bóbr, czy inicjatywa Godzina Kodowania.

Kolejny etap nauki programowania to programowanie w językach wizualnych. Popularnym środowiskiem programistycznym jest Scratch. Jest to intuicyjny język pozwalający w sposób wizualny tworzyć gry, pokazy multimedialne, historyjki itp. Niewątpliwą zaletą Scratcha jest błyskawiczny efekt, jaki widzi uczeń, tworząc swój program. Ma to duże znaczenie przy nauce najmłodszych, którzy oczekują szybkich rezultatów swojej pracy. Uczniowie mają ochotę i motywację do dalszej pracy^[10]. Powiązanie programowania z robotyką i konstruowaniem urządzeń to kolejny wyłaniający się trend w edukacji informatycznej. Połączenie elementów nauki programowania ze sterowaniem robotami uatrakcyjnią uczenie się i podnosi motywację, pokazując fizyczny efekt kodowania, a przede wszystkim - rozwija pomysłowość uczniów.^[11]

Umiejętności programistyczne nabyte we wczesnym etapie edukacji powinno wdrażać do programowania w tekstowych językach programowania.

Nauka programowania uczniów pierwszego etapu kształcenia wymaga dobrego przygotowania nauczycieli. Wzmocnienia ich kompetencji cyfrowych i wyposażenia w praktyczną wiedzę o tym, jak wprowadzać dzieci w świat nowych technologii i programowania – język przyszłości.

[1] Piotr Waszczuk, *Zrozumienie nowych technologii coraz bardziej w cenie*,

<http://itwiz.pl/zrozumienie-nowych-technologii-coraz-bardziej-cenie/> [dostęp 11.03.2017]

[2] - Trendy rynku pracy w 2017 roku- <http://infostrow.pl/biznes/trendy-rynku-pracy-w-2017-roku/cid-76593-a> [dostęp 11.03.2017]

2017-10-09/010,70550,8 [dostęp 11.03.2017]

[3] Marcin Świejkowski, *Kompetencje przyszłości. 5 umiejętności niezbędnych w 2020 roku* - <http://youngpro.pl/artykuly/baza-wiedzy/kompetencje-przyszlosci-5-umiejtnosci-niezbednych-w-2020-roku> [dostęp 10.03.2017]

[4] Marcin Świejkowski Tamże [dostęp 10.03.2017]

[5] J. Morbitzer, *Ekspertyza dotycząca zmian uczenia się osób urodzonych po 1990 r.*, Warszawa, 2014, s. 6

[6] W. Kołodziejczyk: *Edukacja 2.0 – wyzwaniem dla współczesnej szkoły*
<http://edukacjaprzyszlosci.blogspot.com/2010/02/edukacja-20-wyzwaniem-dla-wspoczesnej.html>

[dostęp 11.03.2017].

[7] A. Piecuch, *Uczeń w cyfrowej szkole*, Nowe Horyzonty Edukacji – 1(8)/2014, s. 21

[8] Program Mistrzowie Kodowania. Jak uczą programowania na świecie.
http://www.ceo.org.pl/sites/default/files/news-files/dobre_praktyki_programowania_z_zagranicy.pdf, s. 5; [dostęp 13.03.2017]

[9] Szymon Siarkiewicz; *Programowanie – Pierwsze Kroki*
<http://szymonsiarkiewicz.pl/artykuly/programowanie/programowanie-pierwsze-kroki/>
[dostęp 10.03.2017]

[10] M. Filiciak, K. Sijko, A.Tarkowski „*Nauka programowania w szkołach*” s. 29
http://ngoteka.pl/bitstream/handle/item/262/Nauka-programowania-w-szkolach_raport.pdf?sequence=1[dostęp 13.03.2017}

[11] Program Mistrzowie Kodowania. op.cit., s. 8