

REKOMENDACJE W SPRAWIE UCZENIA/SIĘ PROGRAMOWANIA I ROZWIJANIA KOMPETENCJI CYFROWYCH

Niniejsze rekomendacje i postulaty są efektem wieloletnich doświadczeń kilku organizacji pozarządowych i innych instytucji, zajmujących się wprowadzaniem nowych technologii, a w szczególności nauki programowania, do edukacji szkolnej i pozaszkolnej. Rekomendacje powstały w związku z zapowiedziami Ministra Edukacji Narodowej Anny Zalewskiej, Ministra Cyfryzacji Anny Streżyńskiej oraz Ministra Nauki i Szkolnictwa Wyższego Jarosława Gowina dotyczącymi wprowadzenia nauki programowania w polskich szkołach oraz propozycjami zmian w podstawie programowej kształcenia informatycznego Rady d.s. Informatyzacji przy MEN. Organizacje pozarządowe, instytucje publiczne i prywatne, a także nauczyciele i nauczycielki, aktywne od lat w promowaniu i nauczaniu programowania oceniają pozytywnie zapowiedzi władz i deklarują współpracę w tworzeniu nowoczesnego modelu edukacji cyfrowej w Polsce. Jesteśmy gotowi dzielić się swoimi doświadczeniami w przygotowywaniu nauczycieli i szkół do tej zmiany, rozwiązaniami systemowymi i dydaktycznymi, a także materiałami edukacyjnymi, które wypracowane zostały i sprawdzone w ramach naszych projektów i działań.

I. Dlaczego? O potrzebie uczenia programowania i kompetencji cyfrowych

Uczenie programowania wpisuje się w strategiczne cele związane z rozwojem kapitału ludzkiego i społecznego Polski, wyrównywaniem szans osób z różnych środowisk i regionów oraz budowaniem gospodarki konkurencyjnej, opartej na wiedzy i zrównoważonej. Kształcenie w tym kierunku jest jednym z warunków rozwoju innowacji technologicznych, biznesowych oraz społecznych i umożliwia rywalizację z krajami, z których musimy obecnie importować bardziej zaawansowane technologie. Umiejętność programowania i myślenie komputacyjne, czyli umiejętność kreatywnego rozwiązywania problemów z różnych dziedzin metodami wywodzącymi się z informatyki, są ważne nie tylko ze względu na rosnące zapotrzebowanie na rynku pracy w Polsce, Europie i na świecie (jeśli nie zmienią się obecne trendy, w 2020 r. w Unii Europejskiej zabraknie ok. 800 tys. specjalistów ICT). Myślenie komputacyjne wiąże się także z innymi kluczowymi kompetencjami XXI wieku, takimi jak rozpoznawanie i rozwiązywanie problemów, wyciąganie wniosków, logiczne, krytyczne a zarazem twórcze myślenie. Niezbędne są one do skutecznego uczestnictwa w dynamicznie zmieniającym się środowisku życia zawodowego i społecznego.

Warto zauważyć, że umiejętność programowania to tylko część szeroko rozumianych kompetencji cyfrowych i medialnych. Powszechna obecność technologii komunikacyjnych (zarówno tradycyjnych mediów masowych, jak i mediów elektronicznych) sprawia, że jakość życia współczesnych społeczeństw w coraz wyższym stopniu zależy od zdolności rozumienia i korzystania z informacji, które docierają do nas za pośrednictwem mediów. Od tych kompetencji zależą na przykład: stopień korzystania przez obywateli z usług elektronicznych (w tym usług e-administracji), dostępność edukacji przez całe życie, bezpieczeństwo osobiste i publiczne czy konkurencyjna przewaga. Technologie komunikacyjne kreują nowe możliwości rozwoju społecznego i indywidualnego, ale równocześnie rodzą trudne wyzwania związane

z wykluczeniem cyfrowym, bezpieczeństwem w sieci czy nową dynamiką procesów politycznych i demokratycznych. Dlatego program wdrażania kompetencji cyfrowych powinien obejmować także szeroki zakres kompetencji medialnych.

II. Co zrobić? Jakie kroki należy pilnie podjąć?

1. **Programowanie powinno być wprowadzane do edukacji formalnej jak najwcześniej, od pierwszych klas szkół podstawowych, a nawet już w przedszkolu.** Zgodnie z propozycjami Rady d.s. Informatyzacji Edukacji przy Ministrze Edukacji Narodowej zmiany wymaga podstawa programowa przedmiotów informatycznych. Po odpowiednich modyfikacjach może być jednocześnie wdrażana we wszystkich klasach, szkołach i na wszystkich etapach edukacyjnych, od edukacji wczesnoszkolnej aż do matury. "Umiejętności nabyte podczas programowania są również przydatne na zajęciach z innych przedmiotów, jak i później w różnych zawodach, niekoniecznie informatycznych. Umożliwiają uczniom przejście z pozycji cyfrowego konsumenta na pozycję cyfrowego twórcy oraz przyjęcie roli osoby władającej technologią, a nie tylko poddającej się jej. Nauczanie programowania w szkole nie jest więc jedynie kształceniem programistów" - czytamy w "Mapie drogowej" przygotowanej przez Radę.
2. Równolegle należy tworzyć - w skali ogólnopolskiej i lokalnej - możliwości rozwijania umiejętności kodowania i programowania **w ramach edukacji poza formalnej i nieformalnej, zarówno dla młodych ludzi, jak i nauczycieli oraz wszystkich chętnych.** Dzięki temu osoby zainteresowane tym w zakresie wykraczającym poza program szkolny, mogą uczyć się w sposób bardziej elastyczny i z wykorzystaniem wyposażenia, które może być trudne do zapewnienia w szkole, a zarazem jest użyteczne dla szerszego grona odbiorców (m.in. roboty czy drukarki 3D - np. Chicago Public Library wypożycza roboty szkołom i osobom indywidualnym i organizuje warsztaty z ich programowania oraz druku 3D). Rolę pośredniczącą pomiędzy edukacją formalną i pozaszkolną mogą tu pełnić m.in. współpracujące ze sobą biblioteki szkolne i publiczne. Uwaga: pojęcie "kodowanie" oznacza w naszym ujęciu „programowanie dla każdego”, czyli takie programowanie, w którym wykorzystuje się ograniczony zestaw algorytmów i struktur danych.
3. **Niezbędne jest wspieranie i wykorzystywanie zróżnicowanych modeli wprowadzania podstaw programowania dla różnych grup wiekowych,** uwzględniając również poziom zaawansowania poszczególnych grup. W szkołach podstawowych fundamentów programowania (np. kodowania w języku Scratch albo Kodu) mogą uczyć odpowiednio przygotowani nauczyciele różnych przedmiotów. W klasach I-III mogą to być nauczycielki edukacji wczesnoszkolnej, w klasach IV-VI nauczyciele/lki informatyki i odpowiednio przygotowani nauczyciele innych przedmiotów. Kształcenie umiejętności kodowania i myślenia komputacyjnego w ogólności może, a nawet powinno odbywać się także w trakcie zajęć bez użycia komputera (np. tworzenie prostych schematów blokowych, algorytmów rozwiązań). **W edukacji wczesnoszkolnej zalecane jest nauczanie kodowania przez zabawę,** z wykorzystaniem różnorodnych środków dydaktycznych, w tym gier planszowych czy samojezdnymi zabawkami programowalnymi, tak by kodowanie nie odbywało się wyłącznie z użyciem tabletów czy komputerów. Modelowanie w trakcie gry i zabawy zachowań

polegających na rozwiązywaniu prostych problemów sprawia, że od najmłodszych lat dzieci w naturalny sposób zaczynają funkcjonować w środowisku algorytmicznym. Równocześnie należy rozszerzyć naukę programowania o zajęcia z wykorzystaniem robotów czy mikrokomputerów (np. Arduino czy RaspberryPi). Obecność robotów na zajęciach technicznych i/lub informatycznych powoduje wyraźny wzrost zainteresowania dzieci w różnym wieku nauką programowania, zwłaszcza jeśli można np. 2-3 razy w roku wprowadzić jakiś nowy sprzęt. Ważnym czynnikiem motywującym jest współuczestnictwo uczennic i uczniów w określaniu celów edukacyjnych, a nawet sposobów ich realizacji, tak by uwzględniać potrzeby i pasje młodych ludzi. Przemyślane stosowanie zróżnicowanych form pracy, w tym zwłaszcza metod aktywizujących i zadań ukazujących praktyczne zastosowania umiejętności programistycznych i komputacyjnych ma kluczowe znaczenie (np. programowanie gier komputerowych i testowanie ich na dedykowanym sprzęcie, kodowanie w Minecraft, tworzenie aplikacji na urządzenia mobilne).

4. Na wyższych etapach kształcenia **większość zajęć powinna być prowadzona przez nauczycieli informatyki na lekcjach tego przedmiotu**, zwłaszcza po wprowadzeniu zmian do podstawy programowej. Jednak zarówno w gimnazjach, jak i w szkołach ponadgimnazjalnych mogą i powinny zdarzać się odstępstwa od tej reguły - np. zajęcia z programowania prowadzone w ramach kółek zainteresowań, kół naukowych czy w formie projektów edukacyjnych, zachęcających do kreatywnego rozwiązywania problemów z wielu różnych dziedzin (np. matematycznych, technicznych czy społecznych) metodami wywodzącymi się z informatyki. Kodowanie powinno zatem wyjść poza lekcje informatyki i zajęcia pozalekcyjne - jego elementy powinny znaleźć się na innych przedmiotach, nie tylko ścisłych, ale także np. języków obcych czy wiedzy o społeczeństwie (w formie np. budowania quizów, prostych gier). Kluczowe jest **zachęcanie nauczycieli innych przedmiotów do nauki podstaw programowania**.
5. Zadaniem o szczególnej wadze jest wprowadzenie na szeroką skalę nauczania podstawowych kompetencji programistycznych w **kształceniu i szkoleniu zawodowym**, i to nie tylko w szkołach o profilu informatycznym, ale także np. architektonicznym i budowlanym, mechanicznym, ekonomicznym czy nawet odzieżowym. Uczenie w takich szkołach podstaw kodowania i korzystania z narzędzi cyfrowych może pomóc zmodernizować edukację zawodową w Polsce.
6. Na wszystkich etapach kształcenia ważna jest **współpraca nauczycieli różnych przedmiotów** - tak, aby zajęcia z programowania łączone były z rozwiązywaniem prawdziwych problemów z innych dziedzin nauki i życia. W pracy z uczniami należy **unikać zadań wymagających pracy odtwórczej**, która zniechęca do myślenia i szukania własnych, kreatywnych rozwiązań. Problemy i pytania stawiane uczniom powinny mieć w miarę możliwości charakter otwarty. Nauka programowania musi obejmować proces "kompletny" - od pomysłu i wymagań funkcjonalnych, przez testowanie, po instrukcję dla użytkownika, itp. Stwarza to możliwość i konieczność zorganizowania pracy w grupie, lepiej odzwierciedla biznesową rzeczywistość oraz ukazuje wagę samego pomysłu, który stanowi podstawę udanych start-upów. Duże znaczenie mają **projekty interdyscyplinarne z technologiami informacyjnymi "w tle", angażujące cały "ekosystem" szkoły**, z uwzględnieniem rodziców, lokalnych instytucji, w tym bibliotek i mediów.

7. Jeśli chodzi o edukację uczennic i uczniów, którzy chcą dalej kształcić się w tym kierunku, niezbędne wydaje się **zapewnienie większego wyboru języka programowania na maturze** - lista powinna być otwarta, tak by nauczyciele i uczniowie nie byli skazani na jeden język (pierwszym krokiem mogłoby być np. wprowadzenie możliwości zdawania matury w języku Python). Jednym z ważniejszych czynników tradycyjnie zniechęcających młodych ludzi do uczenia się programowania na wyższych etapach kształcenia jest niewystarczająca znajomość matematyki. Tymczasem - choć dobrzy matematycy często lepiej radzą sobie z programowaniem - to wiele programów i obszarów ich zastosowań nie wymaga wysokich kompetencji matematycznych (np. Java Script, HTML, arkusze stylów - jak CSS czy programy do projektowania graficznych interfejsów użytkownika). Należy zatem kierować się ogólną zasadą, że najważniejsze, by nie zniechęcić młodych ludzi, zanim zdążą zapoznać się i oswoić z programowaniem na „bazowym” poziomie, nie wszyscy z nich muszą przecież zostać informatykami. **Przełamanie oporu przed wejściem w świat programowania jest ważniejsze niż nauczenie konkretnych narzędzi i algorytmów.** W sieci można znaleźć przystępne opisy dostępnych narzędzi i samouczki do nauki różnych języków - trzeba jednak jak najwcześniej pomóc młodym ludziom przekroczyć barierę, wynikającą z lęku i poczucia, że „z tym sobie nie poradzę”.
8. **Konieczne jest wprowadzenie zmian w podstawie programowej zajęć komputerowych (szkoła podstawowa) i informatyki (gimnazjum i szkoła ponadgimnazjalna), ale także innych przedmiotów, na których rozwija się kompetencje informacyjne oraz medialne.** Są one tu rozumiane tu jako zdolność uczniów do m.in. wyszukiwania, selekcji, przetwarzania i dzielenia się informacjami i bezpiecznego korzystania z sieci (patrz s.1). W realiach polskiej podstawy programowej zadania te rozdzielone są pomiędzy nauczycieli różnych przedmiotów (m.in. polonistów, bibliotekarzy), nie należy jednak pomijać tego aspektu w podstawie programowej kształcenia informatycznego. Rozwijanie kompetencji medialnych także w ramach nauczania informatyki pozwoli na ich całościowe kształtowanie - „rozsianie” elementów edukacji medialnej po siatce różnych przedmiotów prowadzi do ich pomijania lub powierzchownego traktowania. Co więcej, logika pracy współczesnych programistów - i specjalistów w wielu innych dziedzinach - zakłada przecież nieustanne korzystanie z szeroko rozumianych kompetencji informacyjnych. I tak, w przypadku jakiegoś problemu, pierwszym krokiem jest zwykle skorzystanie z wyszukiwarki internetowej i sprawdzenie, czy inny programista już go nie rozwiązał i nie opublikował efektu na forum typu np. Stack Overflow. Warto wprowadzić podobny model pracy na zajęciach z informatyki.
9. Edukacja cyfrowa to nie tylko programowanie, ale także **projektowanie 3D, grafika komputerowa, zaawansowany skład tekstu, animacja, montaż filmów, edycja zdjęć, tworzenie muzyki, stron internetowych, programowanie robotów, analiza danych** i wiele innych zastosowań. Kładąc zbyt duży nacisk na programowanie w językach uniwersalnych i rozwiązywanie abstrakcyjnych problemów algorytmicznych możemy zrazić część uczniów, stwarzając fałszywe wrażenie, że chodzi tylko o „suchą” algorytmikę, podczas kiedy naprawdę idzie o to, by pokazać, że komputery i języki programowania to narzędzia pomagające rozwiązywać problemy z różnych dziedzin życia. Może się to wydarzyć na różne sposoby - przez napisanie algorytmu wyszukującego najkrótszą drogę, zaprojektowanie mieszkania z użyciem odpowiedniego programu, przygotowanie montażu zdjęć z wakacji lub stworzenie

edukacyjnej gry komputerowej. Edukacja cyfrowa obejmuje także przygotowanie do pracy “w chmurze” - współdzielenia dokumentów; prezentowania wyników pracy zespołowej, notowanie instrukcji programistycznych w wirtualnych notatnikach, używanie narzędzi do wideokonferencji, itp.

10. W podstawie programowej należy podkreślić znaczenie **współpracy między kodyjącymi uczniami oraz edukacyjnych projektów programistycznych**. Projekty te powinny być dłuższe (zwłaszcza w gimnazjach i szkołach ponadgimnazjalnych) i wykraczać poza 1-2 godziny lekcyjne, bazując nie tylko na pracy uczniów na zajęciach, ale również zaangażowaniu poza lekcją i szkołą. Mogą one także włączać zewnętrznych ekspertów-praktyków, tak rzadko dziś wykorzystywanych w szkołach. Problem braku współpracy między nauczycielami różnych przedmiotów wynika przy tym nie tyle z niewłaściwych zapisów podstawy programowej, ale raczej przestarzałej organizacji pracy oraz źle ustawionych priorytetów w procesie szkolnej edukacji, gdzie nie ceni się dostatecznie pracy zespołowej ani między uczniami, ani nauczycielami. A przecież współdziałanie to jedna z tych kompetencji XXI wieku, której znaczenie - dla programistów, ale i większości zawodów - podkreślają wszyscy pracodawcy i analitycy rynku pracy.

III. Kto? Jak przygotowywać i wspierać nauczycieli?

1. Wprowadzając modyfikacje podstawy programowej należy równocześnie **pozostawić dużą autonomię nauczycielom, tak by mogli wybrać najbardziej odpowiedni model pracy z uczniami oraz sposób kształcenia/rozwijania swoich umiejętności** (studia podyplomowe, szkolenia doskonalące, kursy online, samokształcenie). Jednocześnie trzeba przygotować - lub wskazać wśród istniejących materiałów - **przykładowe, referencyjne programy nauczania i scenariusze zajęć**. Ważne, aby programów tych było kilka i były zróżnicowane - nauczycielom/lkom łatwiej będzie zrozumieć, na czym polega ich autonomia i jak mogą z niej skorzystać, by elastycznie reagować na sytuację w konkretnej szkole (np. sprzęt, łącze internetowe, poziom edukacyjny), klasie, a nawet odpowiadać na potrzeby wybranej grupy uczniów. Należy wykorzystać istniejące już sprawdzone w szkołach programy, scenariusze zajęć i pomoce dydaktyczne, opracowane m.in. przez sygnatariuszy rekomendacji, którzy równocześnie deklarują **gotowość przygotowania kolejnych materiałów edukacyjnych na wolnych licencjach**.
2. Powinny powstać **różne modele nauczania programowania dla dzieci z różnymi niepełnosprawnościami, z wykorzystaniem technologii asystujących**, takich jak programy powiększające i czytające. Program nauczania musi uwzględniać elementy związane z projektowaniem uniwersalnym, tak aby uczący się od początku edukacji rozumieli potrzebę tworzenia produktów dostępnych dla wszystkich niezależnie od rodzaju niepełnosprawności, wieku, używanego sprzętu czy oprogramowania. Nie oznacza to oczywiście, że narzędzia o ograniczonej dostępności (jak np. niezwykle popularny język programowania MIT Scratch) nie mogą być używane w szkole, nauczyciele powinni jednak być informowani o ograniczeniach różnych platform oraz rozwiązaniach alternatywnych, które pozwolą im uczyć bez wykluczania.

3. Niedostateczna ilość sprzętu komputerowego (i jego jakość czy wiek) lub brak szybkiego łącza internetowego to nie jedyne problemy utrudniające rozwijanie kompetencji programistycznych i cyfrowych. Wiele szkół dysponuje dostatecznym wyposażeniem, jednak nie jest w stanie lub nie potrafi efektywnie korzystać ze sprzętu. Główna przeszkoda to **brak wystarczającej wiedzy i umiejętności wśród nauczycieli** oraz - zgłaszana przez część szkół - zbyt mała liczba godzin zajęć komputerowych. Wprowadzenie dodatkowych zajęć z puli godzin finansowanych przez organ prowadzący szkołę (tzw. "godziny jst") lub prowadzenie zajęć w formie "innowacji pedagogicznych" jest uzależnione od dobrej woli władz samorządowych, co powoduje zróżnicowanie poziomu edukacji w różnych miejscowościach i szkołach.
4. **Pilnym wyzwaniem jest zatem stworzenie elastycznego i wielowariantowego systemu kształcenia i doskonalenia nauczycieli informatyki oraz zainteresowanych edukacją cyfrową nauczycieli różnych przedmiotów**, zarówno z zakresu kompetencji programistycznych, jak i dydaktycznych, niezbędnych do skutecznego prowadzenia zajęć. Taki system musi odpowiadać na potrzeby kilku różnych kategorii potencjalnych „klientów” – w tym przyszłych nauczycieli, osób już uczących informatyki, ale chcących rozwinąć swe kompetencje programistyczne, a także nauczycieli innych przedmiotów, w tym edukacji wczesnoszkolnej, którzy są gotowi podjąć to wyzwanie. Należy we współpracy z uczelniami wyższymi i placówkami doskonalenia wypracować standardy przygotowania nauczycieli, przyjmując jednak możliwie otwarte zasady walidacji i/lub certyfikacji, tak by umożliwić jak najszerszej grupie nauczycieli rozwój, ewaluację i potwierdzanie własnych kompetencji. Proces szkoleniowy powinien uwzględniać takie elementy jak: szkolenia stacjonarne; e-learning oraz e-coaching; wsparcie merytoryczne i dydaktyczne w ramach sieci nauczycieli-praktyków, zarówno w formie społeczności wirtualnych, jak i w realu (np. zespołu nauczycieli z jednej szkoły, jednej gminy czy powiatu). Szkolenia i sieci wsparcia mogą być prowadzone przez różne podmioty - wyższe uczelnie, placówki doskonalenia i inne instytucje edukacyjne, mające doświadczenie w tym zakresie i odpowiednie zasoby kadrowe.
5. **Szczególną rolę w rozwijaniu kompetencji i motywacji może odegrać społeczność nauczycieli i nauczycielek programowania** – takie społeczności już powstają m. in. wokół projektów realizowanych przy wsparciu Ministerstwa Cyfryzacji - w tym Koduj z Klasą i Ogólnopolska Akademia Programowania, w projektach Mistrzowie Kodowania, Godzina Kodowania czy Link do przyszłości, w grupie Superbelfrzy, wokół Polskiego Towarzystwa Informatycznego oraz w innych środowiskach, skupionych pod egidą Lidera Cyfryzacji w Szerokim Porozumieniu na rzecz Umiejętności Cyfrowych jako platformy współpracy z Ministerstwem Cyfryzacji i administracją publiczną. Z jednej strony takie wspólnoty powstające wokół konkretnych projektów czy języków programowania mają dużą siłę oddziaływania, z drugiej – warto, aby uczestnicy poszczególnych projektów mieli **możliwość wymiany doświadczeń i zyskali poczucie wspólnoty entuzjastów programowania, a równocześnie współautorstwa przemian polskiej edukacji cyfrowej**.

IV. Jak promować programowanie i kompetencje cyfrowe?

1. Ważnym elementem, uwzględnianym dotąd w zbyt małym stopniu w działaniach prowadzonych ze środków publicznych jest **promocja nauki programowania i kompetencji cyfrowych** - na szczęście zupełnie inaczej wygląda to w projektach prowadzonych przez podmioty komercyjne. Sygnały płynące ze szkół pokazują, że jednym z istotnych problemów jest niedocenienie działań szkoły przez organ prowadzący i kuratorium. Z drugiej strony - zarówno szkoły, jak i konkretni nauczyciele często nie potrafią skutecznie informować o tym, jak i czego uczą, co ma zarówno negatywne skutki finansowe (np. przy staraniach o dofinansowanie), jak i wizerunkowe. Należy z jednej strony ich w tym wspierać, a ponadto należy przeprowadzić ogólnopolską kampanię społeczną, której celem będzie przekonanie osób niezaangażowanych lub wręcz sceptycznych wobec nauki programowania. Europejski Tydzień Kodowania czy Godzina Kodowania stanowią dobre źródło sprawdzonych metod prowadzenia takich kampanii. Do propagowania nowego podejścia do uczenia kompetencji cyfrowych warto włączyć znane i ważne osoby (tak, jak robią to Barack Obama w Stanach Zjednoczonych czy David Cameron w Wielkiej Brytanii). Podobnie w Polsce – **Prezydent, Premier, Minister Edukacji Narodowej i Minister Cyfryzacji, wybitni programiści i programistki, a nawet celebryci** mogliby pokazać, że nauka programowania to jeden z fundamentów innowacyjności i rozwoju gospodarki oraz kapitału ludzkiego i społecznego Polski.

Pod powyższymi Rekomendacjami podpisują się autorzy i współpracownicy reprezentujący takie organizacje lub inicjatywy jak:

Fundacja Centrum Edukacji Obywatelskiej

Fundacja Szkołą z Klasą

Fundacja Nowoczesna Polska

Fundacja Rozwoju Społeczeństwa Informacyjnego

Fundacja CoderDojo

Fundacja Orange

Instytut Spraw Publicznych

Stowarzyszenie Miasta w Internecie

Fundacja Edukacja dla Przyszłości

Superbelfrzy RP

Szerokie Porozumienie na Rzecz Umiejętności Cyfrowych w Polsce

Centrum Cyfrowe Projekt Polska

Parlament Studentów Rzeczypospolitej Polskiej