

# Programowanie w szkole podstawowej



## PODSTAWA PROGRAMOWA

14 lutego 2017 r. Minister Edukacji Narodowej, Anna Zalewska, podpisała rozporządzenie dotyczące podstawy programowej m.in. wychowania przedszkolnego oraz kształcenia ogólnego dla szkoły podstawowej. Tym samym zapisy tej podstawy zaczną obowiązywać nauczycieli i uczniów klas I, IV i VII już od 1 września 2017 roku. Każdego następnego roku objęte nią zostaną kolejne klasy. Od dawna wiemy także, że likwidacji ulegną gimnazja, a w szkole podstawowej dzieci spędzą osiem lat.

Najważniejszych kierunkami zmian, którymi powinni zainteresować się nauczyciele dotychczasowych zajęć komputerowych w szkole podstawowej i informatyki w gimnazjum, są:

- spiralny układ treści
- korelacja przedmiotowa
- TIK w podstawie programowej innych przedmiotów
- wprowadzenie nauki programowania
- praca zespołowa i realizacja projektów edukacyjnych.

Spiralny układ treści gwarantuje stopniowe wzbogacanie wiedzy i umiejętności, powtarzanie i utrwalanie na kolejnych etapach nauczania oraz ich rozszerzanie. Dlatego cele kształcenia - wymagania ogólne - są takie same dla wszystkich etapów edukacyjnych, a treści nauczania - wymagania szczegółowe - skutecznie je rozszerzają. Nowością jest wprowadzenie TIK w podstawie programowej innych przedmiotów, np. geografii, chemii czy biologii. Dotychczas nie było takich zapisów i wspieranie zajęć przedmiotowych nowoczesnymi technologiami wyglądało różnie. Czy ma to znaczenie dla nauczycieli informatyki?

Sądę, że tak. Można liczyć na *kształcenie kompetencji i umiejętności cyfrowych uczniów w ramach realizacji innych przedmiotów*<sup>1</sup>. Przy takim podejściu zdecydowanie lepiej będzie można wykorzystać korelację przedmiotową (np. matematyczno-informatyczną, przyrodniczo-informatyczną, techniczno-informatyczną) oraz realizować projekty edukacyjne. Dzięki takiemu podejściu uczeń nie będzie miał także wątpliwości, że informatyka wspiera wiele dziedzin, które bardzo często bez niej nie mogłyby już w ogóle istnieć. Ale to nie jedyna zaleta. Skoro część zadań z TIK będą realizowali nauczyciele innych przedmiotów, to nauczyciele informatyki będą mogli bardziej skoncentrować się na algorytmice i programowaniu, czyli rozwijaniu kompetencji informatycznych uczniów, które stały się już w tym roku szkolnym (2016/2017) jednym z pięciu podstawowych kierunków realizacji polityki oświatowej państwa<sup>2</sup>. Przy okazji tych zmian zastąpiono nazwę przedmiotu *zajęcia komputerowe - edukacją informatyczną* w klasach I-III i *informatyką* w klasach IV-VI. To zdecydowanie lepiej oddaje charakter tych zajęć i przywraca informatyce, jako dziedzinie nauki (computer science), właściwą rangę w szkole. Nie powinno więc nikogo dziwić, że za jedną z najważniejszych umiejętności rozwijanych

<sup>1</sup> Reforma edukacji 2017. Zmiany w kształceniu ogólnym, specjalnym i zawodowym oraz w obszarze wychowania i profilaktyki, <https://www.ore.edu.pl/nowa-podstawa-programowa/> (zob: ORE REFORMA EDUKACJI 2017. PREZENTACJA OGÓLNA. Stan prawny na\_05.04.2017.pdf)

<sup>2</sup>[https://men.gov.pl/jakosc-edukacji/nadzor-pedagogiczny/podstawowe-kierunki-realizacji-polityki-oswiatowej-panstwa-w-roku-szkolnym-2016\\_2017.html](https://men.gov.pl/jakosc-edukacji/nadzor-pedagogiczny/podstawowe-kierunki-realizacji-polityki-oswiatowej-panstwa-w-roku-szkolnym-2016_2017.html)

w ramach kształcenia ogólnego uznano *kreatywne rozwiązywanie problemów z różnych dziedzin ze świadomym wykorzystaniem metod i narzędzi wywodzących się z informatyki, w tym programowanie*<sup>1</sup>.

Nowa podstawa programowa narzuca obowiązek programowania od klasy pierwszej do klasy ósmej szkoły podstawowej. Gdybyśmy chcieli ją porównać z ustępującą podstawą, to zauważylibyśmy, że zmiany są ogromne. W poprzedniej nie było zapisów uwzględniających myślenie algorytmiczne i programowanie. Wprawdzie w punkcie czwartym zajęć komputerowych dla klas IV-VI wprowadzono *Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera*<sup>2</sup>, ale w treściach nauczania nie rozbudowano w sposób jasny i wystarczający tych wymagań. Zapis, który miał wówczas ukierunkować nauczyciela brzmiał: *Uczeń: za pomocą ciągu poleceń tworzy proste motywy lub steruje obiektem na ekranie*. Takie uzupełnienie przytoczonego celu kształcenia otwierało drogę do wielu niejasnych interpretacji. Do tego wszystkiego w zalecanych warunkach i sposobach realizacji zajęć komputerowych nie było żadnych wyjaśnień ani wskazówek na ten temat. Trudno zapis rozpatrywać w obszarze programowania także dlatego, bo przed właściwym programowaniem muszą pojawić się zagadnienia związane z podejściem algorytmicznym przy rozwiązywaniu problemów. A takich nie było w ogóle.

Dopiero na III etapie edukacyjnym ustępującej podstawy programowej (gimnazjalnym) cel kształcenia wymagań ogólnych ujmuje stosowanie podejścia algorytmicznego przy rozwiązywaniu problemów. I rzeczywiście ten cel jest dobrze odzwierciedlony oraz rozbudowany w treściach nauczania wymagań szczegółowych<sup>3</sup>. Ale zapisów związanych z programowaniem nadal nie było. To wcale nie oznacza, że uczniowie nie mogli programować.

Część nauczycieli miała szansę wprowadzenia programowania we wcześniejszych latach na podstawie autorskiego programu. Dużo pomysłów w tym zakresie dostarczały wydawnictwa, które w podręcznikach wychodziły poza ustępującą podstawę programową i proponowały ciekawe pomysły. Do tego wszystkiego szkoły mogły przystąpić do pilotażu programowa-

nia i realizować go już od 1 września 2016 r.<sup>4</sup> Niezależnie od tego, jaką decyzję nauczyciel podjął, od następnego roku szkolnego 2017/2018 mamy w tym zakresie bardzo istotne zmiany.

Nowa podstawa programowa na każdym etapie edukacyjnym szkoły podstawowej zakłada w celach kształcenia:

1. *Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.*
2. *Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów (...)*<sup>5</sup>

Nieprzypadkowo te dwa cele (wśród przytoczanych tam pięciu) znalazły się na pierwszym i drugim miejscu. Tym samym autorzy podstawy chcieli zapewne zwrócić uwagę na wagę tych zagadnień i konieczność realizowania w pierwszej kolejności. Oba cele mają też bogate odzwierciedlenie w treściach nauczania wymagań szczegółowych. Nauczyciel otrzymuje także cenne wskazówki o warunkach i sposobach ich realizacji. Autorzy podstawy docenili również współpracę w grupach, udział w projektach zespołowych<sup>6</sup> oraz ich zarządzanie. Uczeń wdrożony zostanie do krytycznego myślenia, argumentowania i wnioskowania oraz dobrej organizacji pracy i precyzyjnego prezentowania pomysłów. Będzie uczył się aktywnego słuchania, skutecznego komunikowania się, dokonywania samooceny. Można powiedzieć, że ten cel jest dopełnieniem wcześniej przytoczanych, bo we współczesnym świecie zespołowe rozwiązywanie problemów, także podczas programowania, jest nieuniknione. Nikt dzisiaj nie pisze programów sam, a tworzenie niektórych systemów informatycznych wymaga kilku tysięcy osób podzielonych na zespoły<sup>7</sup>. O tym trzeba mówić dzieciom i dostosowywać pracę grupową do stawianych uczniom problemów.

<sup>1</sup> Rozporządzenie Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej wychowania przedszkolnego oraz podstawy programowej kształcenia ogólnego dla szkoły podstawowej, w tym dla uczniów z niepełnosprawnością intelektualną w stopniu umiarkowanym lub znacznym, kształcenia ogólnego dla branżowej szkoły I stopnia, kształcenia ogólnego dla szkoły specjalnej przysposabiającej do pracy oraz kształcenia ogólnego dla szkoły policealnej, s. 12

<sup>2</sup> Rozporządzenie Ministra Edukacji Narodowej z dnia 23 grudnia 2008 r. w sprawie podstawy programowej wychowania przedszkolnego oraz kształcenia ogólnego w poszczególnych typach szkół, s. 250

<sup>3</sup> tamże, s. 448-449

<sup>4</sup> Do 15 września 2016 r., chęć udziału w pilotażowym wdrożeniu programowania w edukacji formalnej w oparciu o innowacje pedagogiczne w szkołach, zgłosiły 1 592 szkoły. (źródło: <https://programowanie.men.gov.pl/1-592-szkoly-zglosily-chec-udzialu-w-pilotazowym-wdrozeniu-programowania/>) To niecałe 6% wszystkich szkół podstawowych, gimnazjów i ponadgimnazjalnych, które mamy w Polsce (zob. GUS, Oświata i wychowanie w roku szkolnym 2015/2016, Warszawa 2016, <http://stat.gov.pl/obszary-tematyczne/edukacja/edukacja/oswiata-i-wychowanie-w-roku-szkolnym-20152016,1,11.html>)

<sup>5</sup> Rozporządzenie Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej..., op. cit., s. 175

<sup>6</sup> W warunkach i sposobach realizacji czytamy również, że *W trakcie prac nad projektami (indywidualnymi lub zespołowymi) uczniowie powinni mieć również możliwość korzystania z komputerów lub innych urządzeń cyfrowych, w zależności od potrzeb wynikających z charakteru zajęć, realizowanych celów i tematów*, tamże, s. 180

<sup>7</sup> Na przykład 2,5 tys. osób pracowało przy tworzeniu Windows

<sup>7</sup> Zob.: <http://gadzetomania.pl/48868,25-tys-osob-pracuje-przy-tworzeniu-windows-seven>



na ekranie<sup>1</sup>. Warto na chwilę zatrzymać się nad tym zapisem. Autorzy podstawy kładą nacisk na **samodzielnosc w programowaniu**. Nie bez znaczenia jest więc zapis mówiący, że uczeń programuje według **własnych pomysłów**. Daje to dzieciom szansę twórczego myślenia i programowania już na pierwszym etapie edukacyjnym. To jest dla nich zupełnie nowa sytuacja umożliwiająca nie tylko wykonywanie poleceń, których rezultat zaprogramował ktoś inny, ale tworzenie własnych światów i wyrażania siebie w twórczym programowaniu.

Na kolejnym etapie edukacyjnym (IV-VIII), szczególnie w klasach IV-VI uczniowie rozwijają podejście algorytmiczne przy rozwiązywaniu problemów, ale robią to, wykorzystując wiedzę z innych przedmiotów. Zgodnie z podstawą programową może być to, np. *liczenie średniej, znajdowanie elementu w zbiorze nieuporządkowanym lub uporządkowanym, znalezienie elementu najmniejszego i największego oraz wyróżniają podstawowe kroki w algorytmicznym rozwiązywaniu problemu*<sup>2</sup>. Na tym etapie uczeń bardziej świadomie podchodzi do rozwiązywania problemów, dlatego informatyka powinna mieć dla niego charakter formalny. W klasach IV-VI uczniowie powinni więc posługiwać się pojęciami informatycznymi (np. algorytm, sekwencja, iteracja). Dobrze byłoby, aby uczeń po znalezieniu rozwiązania, w drodze logicznego myślenia, potrafił określić dla niego algorytm i wyróżnić podstawowe kroki: *określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu*<sup>3</sup>. Proste algorytmy, zgodnie z podstawą programową, uczeń powinien zapisać w wizualnym języku programowania, wykorzystując przy tym polecenia sekwencyjne, warunkowe i iteracyjne oraz zdarzeń<sup>4</sup>. Wówczas będzie mógł zobaczyć swoje rozwiązanie w działaniu.

W klasach IV-VI możemy wykorzystywać wspomniane już wcześniej środowiska Scratcha, Baltiego, Blockly Games czy All Can Code. Tyle tylko, że uczeń powinien z nimi pracować na wyższym poziomie, niż w klasach młodszych. Należy zachęcać go do samodzielnych działań związanych z rozwiązywaniem problemów, stosowaniem podejścia algorytmicznego: np. programowania własnych komiksów, prostych gier, interaktywnych prezentacji. Uczeń powinien mieć przy tym możliwość testowania swojego rozwiązania na każdym etapie projektowania i modyfikowania kodu źródłowego. Zgodnie z podstawą programową *Uczeń: projektuje, tworzy i zapisuje w*

*wizualnym języku programowania prosty program sterujący (...) obiektem na ekranie komputera*<sup>5</sup>. Wizualne języki programowania świetnie się do tego nadają. Dobrze byłoby, aby potrafił też zaprogramowane projekty zaprezentować rówieśnikom i wyjaśnić ich działanie. Mogłoby to mieć aktywizujący wpływ na całą klasę i jednocześnie wymagałoby od niego dużego zaangażowania oraz dodatkowych działań, także wykorzystania aplikacji do prezentowania treści.

Uczniowie klas VII-VIII, którzy realizowali zajęcia komputerowe w ramach ustępującej podstawy programowej i nie rozwiązywali problemów wykorzystując podejście algorytmiczne, muszą teraz aktywnie się w ten proces włączyć. Ponieważ nie mieli wcześniej okazji spotkać się również z programowaniem, dlatego powinni przejść całość tak, jak uczniowie nowej podstawy programowej klas IV-VI. Tyle tylko, że w krótszym czasie, bo w dwóch, a nie trzech latach. Jest to możliwe do zrealizowania, bo są bardziej dojrzałymi, zdecydowanie na wyższym poziomie logicznego myślenia i gotowi do rozwiązywania problemów poprzez realizowane zajęcia z innych przedmiotów, np. matematykę. Praca z wizualnymi językami programowania nie powinna sprawiać im kłopotu. Wyższe poziomy w Scratchu, Blockly Games (np. labiryntu, ptaka, żółwia, animacji czy stawu) czy też praca w trybie zaawansowanego programowania klasycznego w Baltie powinny spełnić oczekiwania nawet najbardziej wybrednych uczniów. Dla tych, którym układanie skryptów z klocków będzie niewystarczającym wyzwaniem, możemy zaproponować proste polecenia w języku CoffeeScript polecanym przez zespół ekspertów Code Monkey. Ciekawym rozwiązaniem jest także Code Comb, platforma nauki programowania poprzez gry. Uczniowie piszą prawdziwe kody w języku tekstowym Java Script lub Python, sterując w ten sposób bohaterem w grze i pokonując stawiane przed nim wyzwania. W ten sam sposób mogą też nauczyć się języka HTML, którego znajomość jest wymagana zgodnie z nową podstawą programową. Uczniowie są silnie zmotywowani i zaangażowani w proces układania skryptów, lepiej zapamiętują i koncentrują się nad rozwiązywanym problemem. Gry są dostosowane do poziomu wiedzy i wieku ucznia, który, podróżując wraz ze swoimi bohaterami, ma poczucie sprawstwa tego, co programuje.

Uczniowie klas VII-VIII, którzy wcześniej zrealizowali zajęcia z informatyki w klasach IV-VI w ramach nowej podstawy programowej (8-letniej szkoły podstawowej) mogą pójść krok dalej. Ale takich uczniów będziemy mieli dopiero za trzy lata. Wówczas będą mieli już za sobą solidne podstawy rozwiązywania problemów z wykorzystywaniem podejścia algorytmicznego, przez trzy lata mogli poznać też różnorodne sposoby pracy w wizualnych językach

<sup>1</sup> Rozporządzenie Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej..., op. cit., s. 44

<sup>2</sup> tamże, s. 175-176

<sup>3</sup> tamże, s. 176

<sup>4</sup> tamże, s. 176

<sup>5</sup> tamże, s. 176

programowania. Dlatego algorytmy i programowanie powinny wejść na wyższy, bardziej zaawansowany poziom. Uczeń może programować rozwiązania algorytmiczne wybranych problemów. Przy czym w programowaniu powinien skoncentrować się już na językach tekstowych. Jeśli spotyka się z tymi językami po raz pierwszy, to warto go tam wprowadzić poprzez zabawę, na przykład wcześniej wspomniane Code Monkey oraz Code Comb. Dla urozmaicenia można też zaproponować Taxi Code (Silent Teacher). To bezpłatna nauka Java Script od podstaw w postaci 85 krótkich wyzwań, które wyświetlane są na ekranie do uzupełnienia przez użytkownika. W sam raz dla początkujących, żeby oswoić się z językiem, składnią, poleceniami, wprowadzaniem zmiennych, funkcji i podstawowych obliczeń. A wszystko to bezpłatnie, bez zakładania konta.

Na tym etapie edukacyjnym warto rozważyć wprowadzenie języka programowania Python i zdecydowanie rozwijać umiejętności programowania zespołowego i pracy projektowej.

Jeśli będziemy chcieli kontynuować pracę z językami wizualnymi, to jest na rynku kilka ciekawych, bezpłatnych propozycji, którymi warto się zainteresować. Jedną z nich jest Blockly. Dzięki zaprojektowanym przez Google bloczkom możemy łagodnie przejść na wyższy poziom programowania od języka wizualnego do tekstowego. Na podstawie wcześniej ułożonego z klocków kodu w wizualnym środowisku narzędzie generuje zapis w postaci tekstowej odpowiadający językowi: JavaScript, Python, PHP, Lua lub Dart. Uczeń może więc poznać zapis tekstowy wybranego języka wówczas, kiedy utworzy coś wcześniej w języku wizualnym. Może też zapis tekstowy skopiować i przenieść do swojego projektu, w inne miejsce.

Ciekawym rozwiązaniem jest także MIT App Inventor. Dzięki aplikacji dostępnej online uczniowie mogą tworzyć użyteczne projekty na urządzenia mobilne z systemem Android. Tu również będą układali skrypty z bloczków na wyższym poziomie zaawansowania. Jestem pewien, że dzięki środowisku proponowanemu przez zespół Massachusetts Institute of Technology uczniowie mogą zaprojektować ciekawe aplikacje i do tego działające na ich urządzeniach mobilnych. Podobnie jest w App Lab, które proponuje code.org oraz w wielu innych tego typu narzędziach, które można znaleźć w sieci.

Kolejną propozycją z przytoczonej wyżej grupy jest Gameblox. Jest to edytor gier, który również używa języka programowania opartego na bloczkach. Dzięki nim możemy tworzyć i udostępniać projekty własnych gier. Gameblox jest bezpłatny i nie wymaga pobierania na komputer lokalny. Z zaprogramowanymi grami możemy bawić się na witrynie dowolnego urządzenia, w tym także mobilnego. Jestem pewien, że powyższe propozycje spełnią oczekiwania

nawet najbardziej wybrednych uczniów i dostarczą im wielu ciekawych wrażeń.

Zainteresowania informatyczne uczniów mogą być również rozwijane podczas dodatkowych zajęć edukacyjnych i kółek zainteresowań w szkole oraz w ramach zajęć pozaszkolnych.

W klasach VII-VIII na zajęciach z informatyki powinniśmy pomagać w wyborze dalszych kierunków kształcenia, szczególnie związanych z informatyką. Warto wskazywać i pielęgnować świadomość atrakcyjności zawodów, np. programisty czy grafika komputerowego.

## O CZYM WARTO PAMIĘTAĆ?

### To tylko propozycja...

Przytoczone przeze mnie wcześniej środowiska, platformy i języki należy traktować jedynie jako propozycje różnych sposobów programowania. Gdybyśmy bowiem spojrzeli na te rozwiązania z perspektywy Khan Academy, to zauważylibyśmy, że tam podstawy Java Script proponuje się już uczniom powyżej ósmego roku życia. To oznacza, że można byłoby zacząć wcześniej wprowadzać język tekstowy. W Khan Academy jest to osvajanie się z nim poprzez proste rysowanie i wypełnianie figur geometrycznych oraz umieszczanie ich w odpowiednim miejscu (przestrzeni). Jednak w przypadku naszego ucznia barierą mogą być polecenia w języku angielskim. A może się mylę? W każdej klasie będzie to na pewno wyglądało inaczej. Uczniowie przecież prezentują zróżnicowany poziom wiedzy i umiejętności. Nauczyciel musi podjąć samodzielnie decyzję, kiedy i jaki język wprowadza. Podstawa programowa pozostawia mu swobodny wybór i niczego nie narzuca.

### O różnorodności...

Są języki wizualne i tekstowe, z którymi można pracować w szkole podstawowej przez wiele lat. Jeśli chodzi o te pierwsze, to nawet przez osiem (np. Scratch, Baltie). Nie warto jednak przywiązywać się do jednego środowiska czy języka. Trzeba uczniów zachęcać do pracy z wieloma i pokazywać ich różnorodność w rozwiązywaniu problemów i projektowaniu własnych pomysłów. Jeśli stworzymy im takie warunki nauki, to później nie będą dla nich zaskoczeniem inne środowiska, z którymi mogą się spotkać na wyższych etapach kształcenia, a potem w pracy zawodowej.

### O programowaniu małymi krokami...

Nie znam żadnego programisty, który od razu napisałby poprawny kod, program. Natomiast znam wielu, którzy go zmieniają, wracają do wcześniejszych założeń, modyfikują. Odnosząc to do środowiska szkolnego, trzeba pamiętać, że będzie podobnie. W moim przekonaniu z dziećmi należy postępować bardzo ostrożnie. Nie wszyscy będą myśleli i radzili sobie tak samo. Nie wszyscy będą w przyszłości pro

gramistami. Ważne jest abyśmy ich nie zniechęcili. Dobrze byłoby, żebyśmy uczyli ich programowania metodą małych kroków, rozkładali problem na czynniki pierwsze, dużo poświęcali czasu na tłumaczenie poszczególnych etapów działania wcześniej wykonanego zadania. Pozwalajmy im także popełniać błędy, nad którymi będzie można podyskutować i zachęcajmy uczniów do zwracania uwagi na działanie układowego skryptu. Jeśli damy solidne podstawy, pokażemy różnorodność rozwiązań i będziemy z uczniami na ten temat rozmawiać, to jestem pewien, że za kilka lat możemy mieć naprawdę dobrych programistów.

### **DLACZEGO WARTO PROGRAMOWAĆ UCZNIÓW? DOKĄD ZMIERZAMY?**

Dlaczego tak intensywnie (od klasy I do VIII) wprowadzono naukę programowania? Przyczyn jest kilka. Szkoła podstawowa to najlepszy czas na naukę podstaw programowania. Dzieci są wówczas najbardziej „plastyczne”. Można z nich „ulepić” młodych programistów, łatwo zmotywować do działań, wskazać efekty programowania i jego wpływ na otaczającą ich rzeczywistość, także zachęcić do cyfrowego wysiłku poza szkołą, co w efekcie może przynieść uczniowi większą pewność w samodzielnym kierowaniu swoim procesem uczenia się. Programowanie jako trzeci język przyszłości może też skutecznie zahamować cyfrowy „analfabetyzm”, to jest przede wszystkim „konsumowanie Internetu” przez młodych ludzi i do tego nadaje sens, bo wymaga ciągłej nauki. Jest trochę jak gra komputerowa, w której uczymy się od początku, aż uzyskamy poziom mistrza. Bardziej ekscytująca staje się wtedy, gdy napotykamy trudności, bo wówczas musimy myśleć i skupić całą uwagę na rozwiązaniu problemu. Podobnie jak w programowaniu.

Jest jeszcze jeden powód, dostarczany przez pracodawców, którzy nieustannie szukają specjalistów w tym zakresie. *Deficyt na rynku stale się pogłębia.*

*Szacuje się, że obecnie w kraju brakuje 30–50 tys. programistów, zaś do 2020 roku w Unii Europejskiej luka na rynku osiągnie wielkość nawet miliona miejsc pracy. Obecnie problem ze znalezieniem wykwalifikowanych programistów ma 40 proc. pracodawców<sup>1</sup>. Nie może nas to dziwić. Każdy, kto chociaż trochę interesuje się nowoczesnymi technologiami, zdaje sobie sprawę, że za kilka lat wszystko będzie programowalne. I nie trudno sobie to wyobrazić. Niedawno kupiłem pilota do prezentacji, który zdalnie uruchamia filmy, a poprzez poruszanie nim w dół lub w górę zmienia natężenie emitowanego dźwięku. Do tego wszystkiego powiększa lub podświetla zdalnie wyświetlane na ekranie obiekty. Łączy się przy tym z oprogramowaniem zainstalowanym na komputerze. Ktoś musiał to zaprogramować. Pewnie podobnie będzie z wszystkimi urządzeniami, a nawet przedmiotami codziennego użytku, np. długopisami:-) W zasadzie z częścią już tak jest. Poza tym bardzo szybko rozwija się obecnie robotyka, w niedalekiej perspektywie planuje się tworzenie inteligentnych miast (SmartCity), *bardziej energooszczędnych i przyjaznych środowisku, a równocześnie tańszych w utrzymaniu<sup>2</sup>*. I to wszystko nie ma się dziać gdzieś daleko, ale niebawem w Polsce. Nasz słabo jeszcze rozwinięty rynek jest bardzo spragniony nowoczesnych technologii. Warto więc inwestować w ludzi, którzy będą je tworzyli: w cyfrowych mędrców (nauczycieli) i twórców (uczniów) tak, abyśmy w przyszłości widzieli w nich władców technologii, a nie jej konsumentów.*

<sup>1</sup> *Polska firma stworzyła specjalnego robota do nauki programowania. Trafi do polskich szkół*, <https://innowacje.newseria.pl/news/polska-firma-stworzyła,p1956610620>

<sup>2</sup> *Inteligentne miasta dobrą perspektywą dla polskiego rynku. Nowe technologie przyjmą się w Polsce szybciej niż na Zachodzie Europy*, <https://innowacje.newseria.pl/news/inteligentne-miasta-dobra,p1910942477>