

# Uczmy logicznego myślenia

**O pomysłach na sensowne nauczanie informatyki w szkołach z Adamem Jurkiewiczem, autorem książki „Python 3” (recenzja na str. 50) i nauczycielem tego języka w liceum i technikum, rozmawia Tomasz Kulisiewicz.**

**Adam Jurkiewicz**

zdobywca wyróżnienia Szerokiego Porozumienia na Rzecz Umiejętności Cyfrowych w Polsce w latach: 2017, 2020 i 2021, członek zarządu Szkolnej Sekcji Informatyki przy Polskim Towarzystwie Informatycznym. Autor książki „Python 3. Projekty dla początkujących i pasjonatów”, wydanej przez Helion Edukacja, kursów języka Python dla projektu OSE IT-Szkoła – w zakresie szkoły podstawowej i szkoły ponadpodstawowej oraz współautor treści z języka Python w projekcie E-Podręczniki do kształcenia ogólnego dla klas ponadgimnazjalnych. Programista Python, administrator systemów UNIX/Linux, nauczyciel informatyki w pewnym liceum. Prywatnie – miłośnik szant, stateczny mąż, ojciec i dziadek, zwiariowany nauczyciel młodzieży, jeśli tylko ma okazję.



■ **Tomasz Kulisiewicz:** Z grona 274 tys. maturzystów w 2020 r. informatykę rozszerzoną zdawało tylko niecałe 8,8 tys., podczas gdy matematykę rozszerzoną ok. 74 tys., fizykę rozszerzoną ok. 20 tys. Dlaczego informatyka na maturze – mimo braku informatyków na rynku i ich wręcz legendarnie wysokich zarobków – jest tak mało popularna?

■ **Adam Jurkiewicz:** Decyzje maturzystów są racjonalne. Egzamin z informatyki nie jest łatwy, a w zasadzie nic nie daje w staraniach o przyjęcie na studia, nawet na wydziały informatyczne. Brutalnie można powiedzieć, że nie opłaca się jej zdawać na maturze. Co z tym zrobić? A może niczego nie robić? Może tych kilka tysięcy maturzystów zdających informatykę rozszerzoną bez problemów dostaje się na studia informatyczne, a egzamin zdawali dla własnej satysfakcji i sprawdzenia się...

■ **Mamy bardzo niski udział dziewcząt na wydziałach informatycznych. Na uczelniach wyższych kobiety stanowią większość, nawet na uczelniach technicznych jeszcze nie jest źle, ale wśród studentów i absolwentów wydziałów informatycznych kobiet jest zaledwie ok. 15%. Dlaczego? Czy da się przełamać tę tendencję?**

■ Jeszcze do czwartej klasy szkoły podstawowej dziewczynki dają sobie radę z myśleniem algorytmicznym i „pre-informatyką”, często lepiej od chłopców. Potem zaczynają działać stereotypy środowiskowe i nastawienia rodzinne: dziewczynom wmawia się, że są lepsze w kuchni, a w szkole – w przedmiotach humanistycznych czy artystycznych. Obawiam się, że w najbliższych kilku latach to się nie zmieni, ale może później. Na LinkedIn widzę coraz więcej kobiet

w takich obszarach, jak webdevelopment czy projektowanie interfejsów, do czego potrzebne są zdolności plastyczne czy ogólnie coś, co tradycyjnie kojarzy się z kobiecymi umiejętnościami (np. rozróżnianie barw). Nie wiem, czy istnieją jakieś uwarunkowania psychofizyczne, może warto to badać. Wydaje mi się też, że kobiety lepiej odnajdują się w środowisku, lepiej współpracują w zespołach, dlatego więcej ich pojawi się w młodym pokoleniu ściśle współpracującym ze sobą w start-upach. Programowanie w tradycyjnym stylu jest chyba jedynym obszarem, w którym może działać „męski” model pracy samotnego nerda wklepującego po nocach kod do komputera.

■ **Co Pana zdaniem zmienia tworzenie aplikacji narzędziami low-code/no-code? Za kilka lat będzie wokół nas 5 czy 10 razy tyle urządzeń IoT, ale na pewno nie będzie potrzeba do ich oprogramowania tyle samo razy więcej programistów. Jaka przyszłość czeka zawód informatyka?**

■ Oczywiście narzędzia low-code/no-code umożliwiają działom biznesowym czy merytorycznym firm, urzędów i organizacji tworzenie aplikacji do wykonywania konkretnych zadań. Po odpowiednim przeszkoleniu można takie aplikacje składać z gotowych elementów i prawidłowo je parametryzować. Ktoś musi jednak zaprojektować, wykonać i przetestować takie narzędzia – choć i w tym obszarze już widać postępy automatyzacji. Na pewno idziemy w kierunku zmniejszenia udziału pracowitego wklepywania kodu, zamiast tego potrzeba będzie dużo więcej inwencji w opracowywaniu algorytmów, metod oraz narzędzi i bibliotek. Co do przyszłości zawodu: kiedy pojawiły się statki parowe, szkutnicy i budowniczy pięknych żaglowców

czarno widzieli przyszłość zawodu. Jednak wraz z parowcami pojawiło się zapotrzebowanie na potrafiących je budować i naprawiać. Do kucia czy wypasu koni już nie trzeba tylu ludzi, ale kiedy pojawiły się konie mechaniczne, pojawił się zawód mechanika samochodowego. Telewizja miała spowodować bezrobocie wśród radiowców i filmowców, potem Internet miał wyrzucić na bruk specjalistów od telewizji. Oczywiście, były specjalności, które znikwały, ale na ich miejsce pojawiały się dziesiątki nowych. Będziemy mieli do czynienia z zupełnie nowymi specjalnościami informatycznymi, których istnienia dziś nawet nie potrafimy sobie wyobrazić. Wystarczy spojrzeć, co się dzieje w uczeniu maszynowym i sztucznej inteligencji: pojawiają się trenerzy czy „pasterze” robotów. O przyszłość informatyków się nie obawiam. Zarządzać samokonfigurującymi się w zależności od warunków, wymagań itp. segmentami sieci 5G ręcznie się nie da, do tego potrzeba naprawdę sporo ML i AI, ale ktoś takie rozwiązania musi stworzyć, ktoś – kontrolować, czego i jak się uczą, dobierać dane do trenowania programów, ktoś musi umieć tunin-gować takie systemy.

■ **To w takim razie warto postawić pytanie: czego uczyć w szkole – myślenia algorytmicznego czy programowania? I kto ma tego uczyć?**

■ Na pewno trzeba wszystkich uczyć myślenia algorytmicznego, choć uważałbym na terminologię, bo nawet termin „algorytmika” z trudem przebija się wśród użytkowników TikToka i sam uważam, żeby nie przesadzać z taką terminologią wśród uczniów. Jak zwał, tak zwał – najważniejsze jest to, żeby uczyć wszystkich logicznego myślenia, również dlatego, żeby dorosłym ludziom nie dawało się wmówić wszystkich bzdur i kłamstw. Uczenie logicznego myślenia w dzisiejszej szkole jest trudne, a z punktu widzenia niektórych polityków – wręcz niekorzystne. Z drugiej strony jest też obecnie moda na uczenie wszystkich programowania. Nie wszystkim jest potrzebna umiejętność kodowania, ale wszystkim z pewnością przyda się zdolność logicznego myślenia. Dlatego to programowanie na poziomie do czwartej klasy szkoły podstawowej jest cenne, bo uczy właśnie myślenia logicznego, algorytmicznego, w dodatku na życiowych przykładach, np. porządkowania jakiś elementów otoczenia. Wchodzące trochę później środowisko Scratcha też jest przydatne jako wprowadzające różne abstrakcje w poglądowy i łatwy sposób.

Problem zaczyna się później, w klasach VII-VIII. Pomysł, żeby w tych klasach wprowadzić już tekstowy język programowania był dobry, pod warunkiem że będą potrafili tego uczyć odpowiednio wyszkoleni nauczyciele, a było z tym – delikatnie mówiąc – średnio. Dobrze o tym wiem, bo właśnie w tamtych latach uczyłem nauczycieli Pythona i zdawałem sobie sprawę, że trzeba im pomagać, bo dotąd na lekcjach informatyki uczyli głównie posługiwania się nieśmiertelnym programem do rysowania, czego zresztą mieli dość już i nauczyciele, i uczniowie.

Skąd wziąć nauczycieli do uczenia nowoczesnej informatyki? Nie pchają się do szkoły drzwiami i oknami. Trzeba by ich odpowiednio wynagradzać... Nie widać chęci do finansowego docenienia nauczycieli informatyki, tym bardziej, że od razu pojawi się pytanie: a dlaczego tylko nauczycieli informatyki? A co z pozostałymi?

Bardzo nam potrzeba uczenia wszystkich – i nauczycieli, i uczniów – stosowania narzędzi i rozwiązań informatycznych w różnych dziedzinach: w uczeniu polskiego, historii, biologii itp. Począwszy od prawidłowego korzystania z edytora tekstu, żeby coś napisać i żeby to dobrze wyglądało, miało prawidłową strukturę, dało się przetwarzać, nie mówiąc już o odczycie narzędziami dla osób niewidomych czy niedowidzących. W wielu szkołach ciągle jeszcze idzie się na lekcje informatyki do oddzielnej sali z komputerami, zamykanej kratą z dwoma kłódkami, bo jest tam kosztowny sprzęt, i walczy się ze smartfonami zamiast próbować ich wykorzystania w edukacji. Trochę to zmieniła pandemia, w społecznych inicjatywach powstawały serwisy takie, jak np. lekcjewsieci.pl, do których nauczyciele różnych przedmiotów wstawiają treści edukacyjne jako swoje pomysły na lekcje, wykorzystując powszechnie dostępne narzędzia informatyczne.

■ **Od kilku lat zajmuje się Pan uczeniem Pythona, od niedawna w szkole. Nie wdając się w dyskusję „o wyższości świąt Wielkanocy nad świętami Bożego Narodzenia” – dlaczego Python i skąd uczenie w szkole?**

■ Dobrze się złożyło, że właśnie wtedy, kiedy trzeba było zacząć uczyć tekstowego języka programowania, w podręcznikach pojawił się Python. To był dobry wybór, bo w odróżnieniu choćby od C++ Python ma „niski próg wejścia”. A więc krótką odpowiedzią na pierwsze pytanie jest: bo Python jest łatwy, w dodatku łatwo się czyta programy – uczniowie przeważnie dają sobie radę z jęz. angielskim na tym poziomie i nie czują zniechęcenia zbyt wysokimi wymaganiami. Wprawdzie spotkałem się ze strony nauczycieli z pytaniami, dlaczego nie spolszczyć Pythona, by wszystkim było jeszcze łatwiej – ale pomińmy to milczeniem.

Na drugie pytanie odpowiedź też jest krótka: lubię uczyć ludzi, dzielić się swoją wiedzą. Poza tym poprosiła mnie o prowadzenie zajęć dyrektorka liceum, którą kiedyś uczyłem na kursach dla nauczycieli. Na szczęście na życie zarabiam w innych moich obszarach działalności, więc mogłem sobie pozwolić na wsparcie szkoły w bliskiej mi dziedzinie. Po latach prowadzenia szkoleń i kursów informatycznych dla dorosłych, w tym dla nauczycieli, skorzystałem z tej okazji, żeby wzbogacić swoje doświadczenie zawodowe. Uczę w liceum i technikum. Są to dla mnie bardzo ciekawe i ożywcze doświadczenia, dają wiele materiałów do przemyśleń.

Eksperymentowałem w szkole z różnymi sztuczkami, na przykład pokazując na początku roku moim uczniom znany podręcznik „Black Hat Python”<sup>1</sup> – podręcznik hakowania, ale

tego, które nazywane jest *ethical hacking*, w odróżnieniu od przestępczego włamywania się do systemów – i pytając, kto chce zostać hakerem albo kto chce się nauczyć pisać boty do komunikatora Discord. Na początku prawie wszyscy z 300 uczniów zgłosili się tłumnie, myśląc, że za chwilę będą się potrafili włamywać do banków czy na strony rządowe. Zgodnie z założeniami kursu, do końca dotrwali tylko najwytrwalsi, którzy dali sobie radę ze sporym nakładem pracy, a przede wszystkim myślenia.

” *A tak przy okazji – od dwudziestu lat walczę z zupełnie zniekształconym znaczeniem słowa „haker”, oryginalnie opisującym kogoś, kto ma dużą wiedzę informatyczną i chce się nią dzielić. Choć to walka bardzo trudna, to jednak jeszcze się nie poddaję i na różnych konferencjach dla nauczycieli nie tylko przedstawiam się jako „haker edukacji”, ale staram się to samo określenie podsuwać nauczycielom.*

Wracając do podstawy programowej – uważam, że czas byłoby zmienić podstawę dla liceum, która moim zdaniem ma za silny przechył w stronę pakietu Office. Dwadzieścia lat temu, kiedy wszyscy pracownicy biurowi musieli się nauczyć korzystania z komputerów, na pewno miało to sens, ale czasy się zmieniają. Uważam, że warto zajmować się w szkole TeXem. Ci, którzy się zetknęli z oprogramowaniem składu zbudowanym na systemie TEX wiedzą, że nie jest to zwykły program DTP, ale rozbudowany system programowania w języku znaczników. LaTeX i ogólnie TEX jest świetnym przykładem uczenia nie tylko składu, lecz także nauki specjalistycznego języka programowania, natomiast historia rozwoju tego systemu – przykładem tworzenia rozwiązań otwartoźródłowych: źródła od początku są w domenie publicznej. Pokazując LaTeXa swoim uczniom, mówię: zanim się przerazicie, że to takie skomplikowane, powinniście zdać sobie sprawę, że jeśli wybieracie się na dobre studia, na których będziecie musieli coś porządnie napisać, to na pewno się z nim spotkacie. Zapoznają ich także z takimi edytorami, jak HackMD, edytor online bardzo popularnego, prostego i uniwersalnego języka znaczników Markdown, działającego w różnych środowiskach systemowych, stosowanego też w systemach zarządzania treścią.

■ **Czy szkoła i uczelnia mogą nadążyć za bardzo szybkim rozwojem metod i zastosowań informatyki?**

■ Nie wdając się w rozważania na temat dzisiejszej polityki edukacyjnej i kierunków działań resortu edukacji

i ograniczając do dziedziny, którą się bezpośrednio zajmuję: w szkole mamy do czynienia z kilkuletnim cyklem tworzenia i zatwierdzania podstawy programowej oraz podręczników – to w szybko rozwijających się dziedzinach stanowi dużą przeszkodę. Jeśli w fizyce stała grawitacja jest ciągle taka sama, choć i tam się wiele zmienia, to z informatyką mamy poważny problem wynikający z ogromnego tempa zmian. Problemem są też nauczyciele, którzy mają przekazywać tak dynamicznie zmieniającą się wiedzę. Czy szybkie zmiany da się przeprowadzić z obecną kadrą, czy też trzeba poczekać na wymianę pokoleniową? Z wymianą pokoleniową jest duży kłopot, bo jak wspomnieliśmy, nie widać tłumów młodych ludzi zgłaszających się do zawodu. Sytuacja zaczyna się robić podobna jak z pielęgniarkami: za kilka lat pójdą na emeryturę wszystkie, które nie są na tyle młode, żeby wyjechać do Niemiec, Szwecji czy Norwegii... W dodatku informatycy nie muszą się nawet ruszać z domu, żeby pracować na drugim końcu świata za duże pieniądze. Kto będzie uczył nauczycieli informatyki i ogólnie – informatyków – na uczelniach, na których młodzi nie chcą zostawać, a nawet kończyć drugiego stopnia studiów informatycznych? Ten problem wykracza poza ramy naszej rozmowy, powiem tylko, że od walki z wyimaginowanymi wrogami czy ideologiami ani PKB, ani Human Development Index czy inne wskaźniki rozwoju nam nie wzrosną. Na pewno potrzebna jest ogólnokrajowa dyskusja na temat modelu rozwoju kraju i rozwoju edukacji – a nie tylko rozwoju samej branży informatycznej. Trzeba przemyśleć, do kąd chcemy zmierzać.

Wracając do uczenia informatyki w szkołach: chętnie podyskutowałbym z resortem edukacji i z ekspertami nad „pływającą podstawą programową” informatyki czy nawet „dynamicznie samokonfigurującymi się programami nauczania”, automatycznie dostosowującymi się do warunków i potrzeb, jak te sieci 5G, o których wcześniej wspomnieliśmy, „tuningowanymi” przez nauczyciela dobierającego elementy w zależności od umiejętności – i uczniów, i jego samego. Technicznie jest to możliwe do realizacji na przykład na platformach elektronicznych, do tworzenia takich programów nauczania można zastosować osiągnięcia ML i AI. Trzeba pokonać opór ludzi, którzy ciągle jeszcze nie mogą się przyzwyczaić do tego, że dziś i jutro jedyną stałą rzeczą wokół nas jest i będzie ciągła zmiana. Na razie takiego modelu uczenia, w którym nauczyciel sam sobie konfiguruje podstawę programową, nie da się zrealizować w dzisiejszym schemacie sprawdalności zewnętrznej. Ogólnie realizacja takiego pomysłu w bismarckowskim modelu szkoły powszechnej jest niemożliwa, bo wymaga radykalnej zmiany sposobu myślenia o celach i metodach edukacji. To jest prawdziwe wyzwanie – i miejmy nadzieję, że zostanie kiedyś podjęte.

<sup>1</sup> Autorstwa J. Seitz. Polskie wydanie „Black Hat Python. Język Python dla hakerów i pentesterów” (e-book, Helion, Gliwice, 2015).