Programujemy już dziesiątki lat...

dr Jan A. WIERZBICKI

W ostatnich kilku latach można zauważyć silny trend związany z rozpowszechnianiem nauki programowania. Ta sytuacja skłoniła mnie do kilku refleksji. Nasuwa się naturalne pytanie, czy programowanie to jakaś nowość, czy dopiero teraz zaczynamy uczyć programowania? Te pytania można rozszerzyć do pytania ogólniejszego, dotyczącego informatyki.

Często na wielu kursach i zajęciach, jakie prowadzę ze studentami, uczniami i nauczycielami zadaję pytanie: Ile lat ma informatyka? Prawie zawsze udzielane odpowiedzi to: dwadzieścia lat, trzydzieści lat, około czterdziestu lat itp. Nikt nawet nie pomyśli o większej liczbie. Informatyka wiązana jest przez ogół osób z bezpośrednim użyciem komputerów i technologii. Czy słusznie? Moim zdaniem – zdecydowanie nie! Dlatego, że informatyka to nauka ścisła, korzeniami sięgająca starożytności. W wielu zapiskach starogreckich, starorzymskich mamy do czynienia z jednym z wyróżników informatyki, jakim jest algorytm. Od algorytmów opisujących na przykład proste sposoby obliczania do klasycznych algorytmów, takich jak algorytm Euklidesa czy sito Eratostenesa.

Informatyka jest związana z przetwarzaniem różnych danych i informacji. Komputery i obecna technologia są narzędziami opierającymi się na informatyce. Informatyki nie można, co niestety bardzo często się dzieje, mylić z technologiami informacyjno-komunikacyjnymi (TIK). Technologie informacyjno-komunikacyjne to „produkty" informatyki do wykorzystania zarówno w pracy zawodowej, jak i w życiu codziennym.

Jeśli informatyka jest tak starą nauką, to czy programowanie jest też tak stare?

Musimy znowu zadać pytanie – co to jest programowanie?

Programując, wydajemy komputerowi lub innemu urządzeniu stosowne polecenia, które ma on wykonać. Aby skutecznie wykonać zadane polecania, muszą być przemyślane i wykonane w odpowiedniej kolejności. Ten sposób pracy to nic innego jak algorytm.

Algorytmy, tak jak już wspomniałem, znane są od starożytności, a właściwie od początku istnienia ludzkości, bowiem nawet ludy pierwotne miały swoje metody polowań czy upraw.

Sam algorytm możemy tworzyć i testować bez komputera. Programowanie samo w sobie wymaga już zastosowania jakiegoś urządzenia, w którym opracowany program ma działać. Programy mogą wyglądać różnie. Sięgając w przeszłość – na przykład mechaniczne maszyny liczące też miały wbudowany program, tyle że oparty na mechanizmie różnych zębatek.

Po tym krótkim przeglądzie możemy już stwierdzić, że programowanie nie jest elementem nowym. Może więc jest czymś nowym w obecnej dydaktyce? Rzeczywiście, w nauczanym realnie materiale na lekcjach, nawet tych, które w nazwie miały informatykę, programowanie nie było wprowadzane dość powszechnie. Raczej uczono TIK.

Można zauważyć, że w bardzo wielu przypadkach osoby świetne w pracy z TIK mają małe doświadczenie w programowaniu. Wiążą się z tym problemy z myśleniem abstrakcyjnym oraz znajomością podstaw matematyki czy innych dziedzin. Narysowanie kwadratu, które wymaga podania jego właściwości, takich jak liczba boków i miara kąta wewnętrznego to zadanie czasami prawie niewykonalne przez ucznia. Dlatego jest tak ważne, aby od samego początku edukacji uczyć elementów programowania, które mogą i zazwyczaj wykorzystują wiele pojęć z różnych przedmiotów. Programowanie uczy logicznego myślenia, kreatywności, co jest niezwykle istotne w kształceniu każdego ucznia.

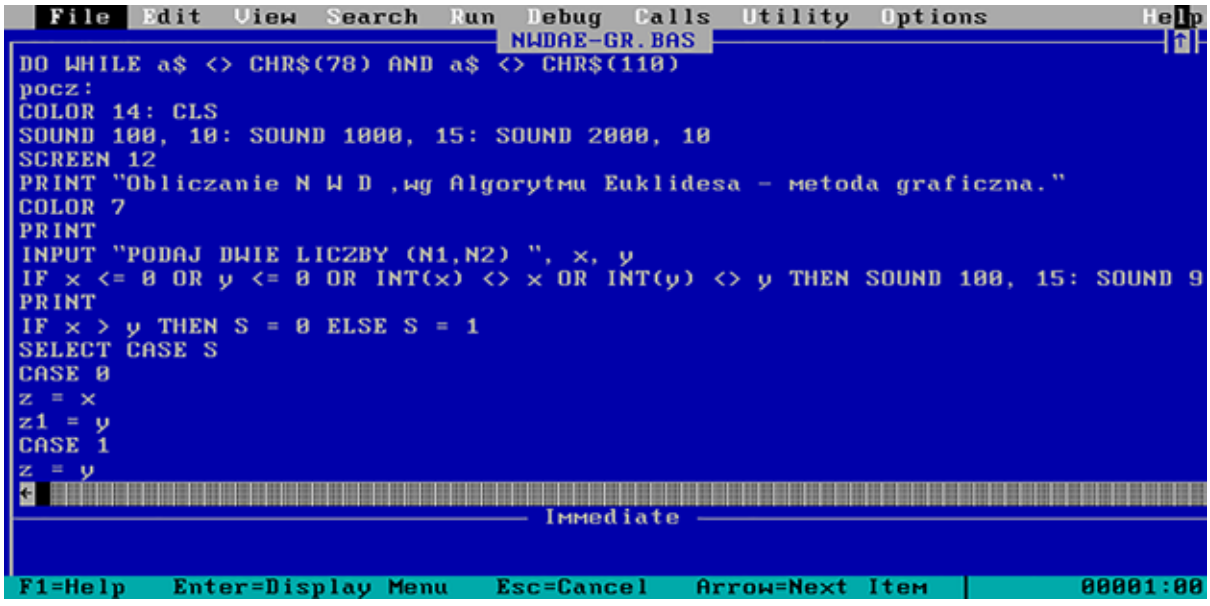
Opiszę teraz moje bezpośrednie doświadczenie z programowaniem wskazujące na to, że czasami potrzeba bardzo mało, aby nauczyć się bardzo wiele. Mój pierwszy komputer, który miałem w połowie lat 80. ubiegłego wieku, to był Commodore 16 (nie popularny wówczas Commodore 64). Komputer ten nie był zatem zbyt popularny, w szczególności w Polsce, więc praktycznie nie było do niego żadnego oprogramowania, posiadał tylko wbudowany interpreter języka programowania Basic. Można zadać pytanie, czy w takiej sytuacji czegoś się nauczyłem, pracując na tym komputerze?

Odpowiedź brzmi – bardzo dużo! Wszystko, co chciało się mieć, trzeba było samemu zaprogramować. Programowanie było nierozerwalnie związane z nauką i ugruntowywaniem wiedzy z różnych przedmiotów szkolnych. Budując program, uczymy się symulacji procesów i pojęć z różnych dziedzin. Programując nawet zabawkę-grę, uczymy się na przykład geometrii czy trygonometrii.

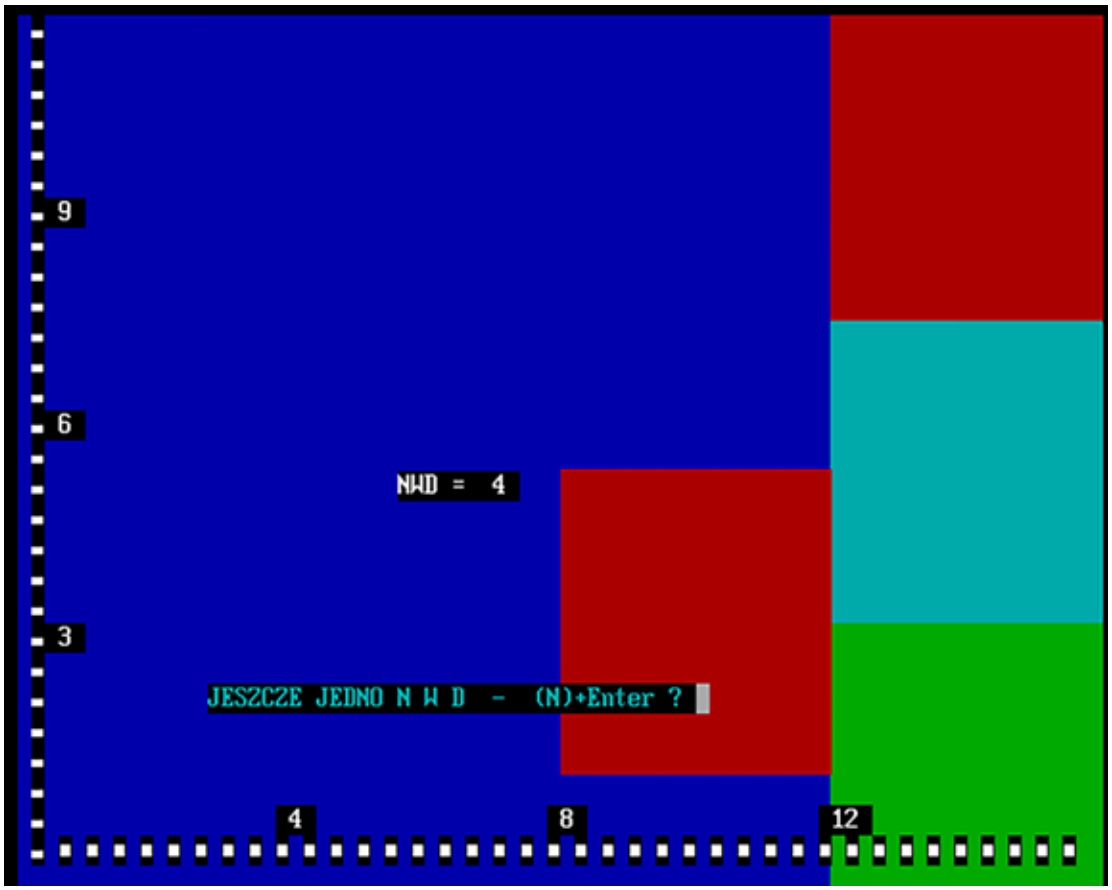
Napisałem mnóstwo programów – od gier do baz danych – co pozwoliło mi się nauczyć i ugruntować bardzo wiele zagadnień. Frajdę miałem przy tworzeniu programu, np. gry, a samo granie nie było już takie atrakcyjne.

Kolejne programy dydaktyczne napisane już na innych komputerach przedstawiały i obrazowały tradycyjne algorytmy, np. algorytm Euklidesa, badanie wykresów funkcji, metody sortowania, badanie ciągów, wizualizacje pojęć symetrii, jednokładności, działania na wektorach. Programy analizowały też teksty literackie, szukając palindromu lub anagramu.

Dzisiaj programy napisane pod dostępnym ponad 20 lat temu systemem operacyjnym DOS da się nawet uruchomić na najnowszych systemach operacyjnych stosując oprogramowanie do tworzenia maszyn wirtualnych (rys. 1, 2).



Rysunek 1. Fragment kodu programu algorytmu Euklidesa w języku Basic.



Rysunek 2. Interpretacja graficzna algorytmu Euklidesa – efekt działania programu w języku Basic.

Opisane przykłady świadczą, że nauka informatyki zawsze obejmowała jako jeden z kluczowych elementów naukę programowania. Jak można łatwo zauważyć, wiele pojęć przywoływanych w ramach uczenia algorytmiki i programowania jest ponadczasowych. Algorytmy i programy napisane kilkadziesiąt lat temu nadal obowiązują i będą obowiązywać w przyszłości, wskazane jako konieczne dla kształcenia uczniów.

Analiza i zrozumienie klasycznych algorytmów daje uczniom niezwykłe korzyści nie tylko w zakresie nauki informatyki, ale też ich ogólnego rozwoju. Uczy analizy i rozwiązywania problemów, kreatywności, systematyczności pracy oraz samokształcenia.

Obecnie naukę programowania można już wdrażać na początkowych etapach edukacyjnych. Opiera się ona w dużej mierze na idei geometrii żółwia, na której oparty był język programowania Logo – wspaniałe narzędzie dydaktyczne.

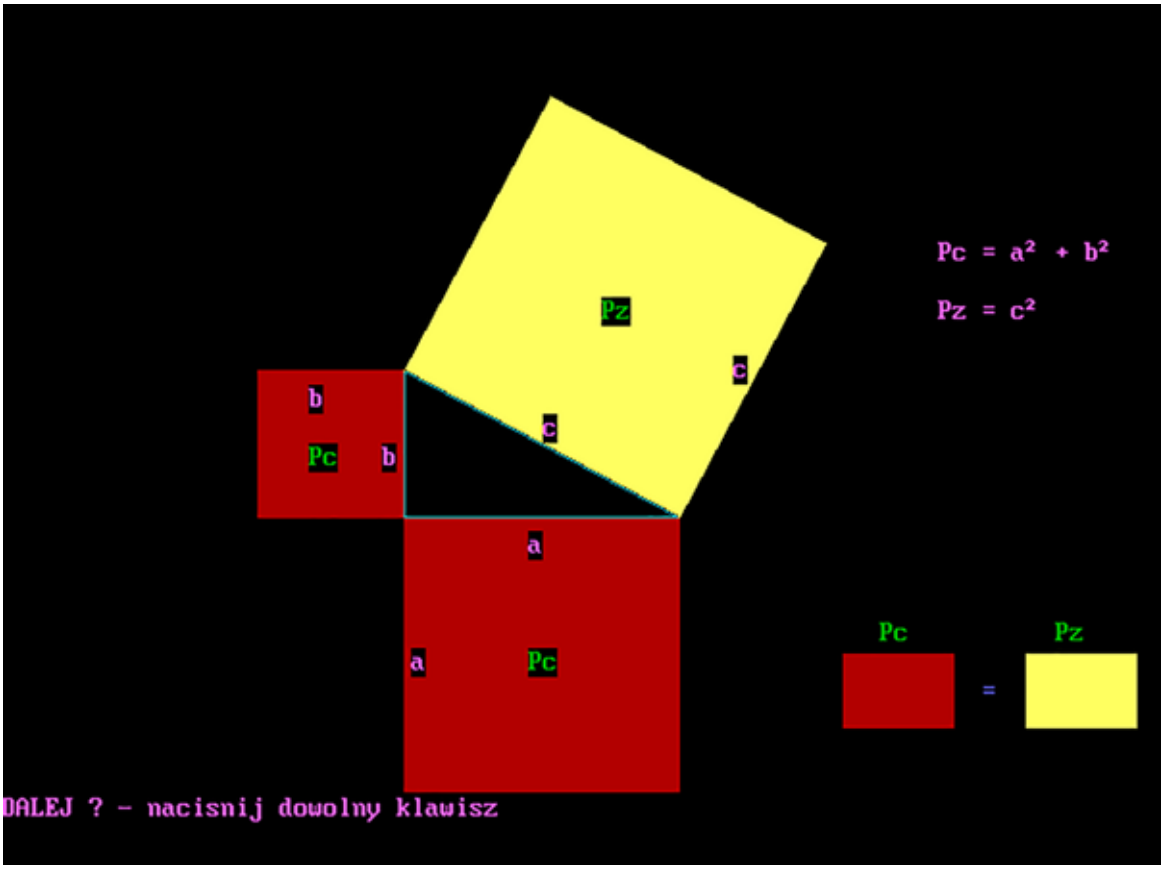
Język ten został stworzony przez Seymoura Paperta na potrzeby nauczania. Papert doskonale rozumiał, czym jest istota zastosowań informatyki w nauczaniu. Odwrócił popularne powiedzenie – „nauczanie wspomagane komputerowo" na „dziecko programuje komputer". Był to ważnym przełom, bowiem stwierdzenie „nauczanie wspomagane komputerowo" sugeruje, że chcemy programować dzieci i innych uczących się, a to jest absurd.

Tworzenie historii obrazkowych czy grafiki, to w tle nauka różnych pojęć z wielu przedmiotów szkolnych, na przykład matematyki czy języka polskiego. Chcąc coś stworzyć, uczymy się. Jak to określił Papert – „Uczenie przez tworzenie", chęć tworzenia wymusza naukę. Poruszając obiektem, uczeń musi znać podstawowe pojęcia geometrii, jeśli w programie ma zostać wypowiedziany jakiś wierszyk, uczeń musi go poznać i poprawnie wpisać. Tło historyjki obrazkowej może wymagać przedstawienia odpowiedniej scenerii związanej z czasem historycznym czy miejscem geograficznym. Mamy więc naukę historii, geografii, przyrody.

Jak widać, pomysłów na pisanie programów może być bardzo wiele, w zależności od tego, czego chcemy nauczyć ucznia. Zaczynamy programować na zasadzie zabawy, nie przytaczamy bezpośrednio definicji pojęć informatycznych, ale je stosujemy w tle, poprzez pewne wymuszenie spowodowane chęcią rozwiązania danego problemu. Ważne jest, aby uczeń starał się samodzielnie rozwiązywać zadania, testował rozwiązania i poprawiał ewentualne błędy. Te stwierdzenia są jawnie zapisane w podstawie programowanej przedmiotu informatyka już w szkole podstawowej.

Zagadnienie programowania jest istotne, umożliwia tworzenie czegoś nowego od podstaw i pełnię realizacji własnych wizji. Niestety rozwój technologii, najróżniejszych narzucanych programów i serwisów powoduje, że mało osób programuje i uczy się wartościowych elementów programowania. Prościej jest napisać różne bardziej lub mnie mądre wpisy na forach internetowych, niż programować. Obawiam się, czy paradoksalnie z użyciem technologii nie cofamy się do lat 50., 60. ubiegłego wieku. W owym czasie były też komputery, ale ich rozwojem, programowaniem zajmowały się tylko nieliczne grupy specjalistów. Ogół społeczeństw nie znał elementów informatyki i jej pozytywnego wpływu na rozwój myślenia, rozwiązywania problemów oraz nauki. W latach 80. fascynacja komputerami wymuszała poniekąd zajmowanie się programowaniem. Niestety, dalszy rozwój TIK i narzucanych form serwisów oraz aplikacji komputerowych nie promował globalnie idei programowania, co było wielką szkodą dla edukacji każdego człowieka. Na szczęście trend ten się teraz zmienia, czego dowodem są też zapisy w nowej podstawie programowanej przedmiotu informatyka zarówno dla szkoły podstawowej, jak i ponadpodstawowej. Zachęcając dzieci do programowania, nie mamy na celu, aby stały się one w przyszłości programistami, lecz dbamy o ich ogólny rozwój intelektualny. Programowanie uczy różnej wiedzy i wielu umiejętności, co jest wartościowe dla każdego, niezależnie od tego, jaką w przyszłości będzie miał profesję.

dr Jan Aleksander WIERZBICKI jest kierownikiem ds. nauki w Ośrodku Edukacji Informatycznej i Zastosowań Komputerów w Warszawie.



Rysunek 3. Efekt działania programu napisanego w pierwszej połowie lat 90., dowodzącego twierdzenia Pitagorasa.