

Programowanie jako metoda kształcenia

Dawniej sprawa była jasna – maszyny nie myślały i nie rozumowały, działania oparte na rozumowaniu były wyłączną domeną człowieka (pomijając instynktowne działania zwierząt). Aby podjąć decyzję opartą na rozumowaniu, człowiek musiał mieć wiedzę, aktualne informacje i musiał umieć rozumować – dedukować i indukować, kojarzyć fakty, wnioskować. Od czasu wynalezienia komputera, w końcu też maszyny tylko elektronicznej, a nie mechanicznej, sprawa nie jest już taka prosta. Komputer bowiem też ma pamięć, a w niej informacje – często gigantyczne zasoby informacji, w dodatku na bieżąco uaktualniane – i potrafi te informacje przetwarzać, czyli kojarzyć fakty i na ich podstawie wnioskować. I tutaj dochodzimy do istoty tego, o czym jest ten felieton, a mianowicie o umiejętności programowania. Komputer oczywiście nie myśli, komputer tylko wykonuje programy – myśli natomiast programista, który te programy pisze. Programowanie jest fascynującą działalnością dostępną tylko współczesnemu człowiekowi. Zachowując wszelkie proporcje, przypomina powoływanie do istnienia pewnego bytu, który się inteligentnie zachowuje, w dodatku całkowicie zgodnie z wolą programisty. Pragnę zaznaczyć, że mówię tu o człowieku, który tworzy algorytm według własnego pomysłu i wyraża go, czyli koduje, w pewnym języku programowania, a nie o człowieku, który dostaje gotowy algorytm przygotowany przez kogoś innego i jedynie go koduje. Oczywiście, żeby biegle kodować, trzeba opanować zasady programowania i sam język. To trochę tak, jak z grą na fortepianie – zanim zacznie się uprawiać działalność artystyczną będącą źródłem wielkiej satysfakcji, najpierw trzeba poćwiczyć gamy i palcówki.

Z edukacyjnego punktu widzenia najważniejsze w nauce programowania jest to, że kształtuje ono zdolność do algorytmicznego myślenia. Jeśli będzie spełniony podany przeze mnie warunek, to komputerze zrób to, co ci każę, w przeciwnym razie zrób coś innego, co ci każę. Programista po pierwsze musi dokładnie wiedzieć, co komputer ma zrobić. Nie może pominąć żadnej czynności, choćby najdrobniejszej, bo wówczas wynik nie będzie taki, jakiego oczekiwał on sam i użytkownicy jego programu. Co jednak ważniejsze, musi rozpatrzyć wszystkie możliwe warunki, czyli sytuacje, które mogą zajść przy rozwiązywaniu problemu, który oprogramowuje. Musi uruchomić swoją wyobraźnię i przewidzieć wszystkie, podkreślam wszystkie możliwe sytuacje, a dla każdej powiedzieć co zrobić. Komputer, na którym programista testuje swój nowo napisany program, jest jak lustro, w którym odbija się jego zdolność do logicznego myślenia i wyobraźnia. Jest też jak ręce rzeźbiarza, które w miarę pracy tę zdolność do logicznego myślenia i wyobraźnię kształtują. Dlatego każde dziecko w szkole powinno uczyć się programować. Niekoniecznie po to, aby w przyszłości zostać zawodowym programistą komputerów, choć to bardzo dobry zawód, tylko po to, aby nauczyć się algorytmicznego podejścia do rozwiązywania problemów, zarówno w życiu

osobistym, jak i zawodowym. Ludzie, którzy myślą algorytmicznie, skonfrontowani z jakimś problemem nie uciekają od niego, nie panikują i nie narzekają, tylko zabierają się za jego rozwiązanie. Analizują możliwe warianty, sprawdzają warunki, pod jakimi te warianty mogą się zrealizować, i wybierają najlepszy. Ludzie, którzy tak myślą i tak działają są w społeczeństwie bardzo cenieni.

Nauka programowania może też kształtować umiejętności uczniów i studentów w zakresie zarządzania. Mam tu na myśli nauczanie programowania współbieżnego. Programista musi wówczas zapanować nad komputerem lub zbiorem komputerów dołączonych do sieci, na których jest współbieżnie wykonywanych wiele programów. Programy te komunikują się między sobą wysyłając do siebie sygnały i komunikaty, aby osiągnąć wspólny cel. Zadaniem programisty jest sprawienie, aby każdy potrzebny sygnał był wysłany, aby żaden sygnał nie był zignorowany lub pominięty, aby każdy trafił do właściwego odbiorcy, aby wszystkie komunikaty były jednoznacznie zrozumiałe przez wszystkich odbiorców i właściwie zinterpretowane. Programista, który zmierzy się z problemami komunikacji i synchronizacji w środowisku programowym i nauczy się je rozwiązywać, będzie znacznie lepiej rozumiał analogiczne problemy w odniesieniu do zarządzania zespołami ludzkimi, choć te są oczywiście znacznie bardziej złożone choćby ze względu na aspekty emocjonalne, których w przypadku programowania nie ma, a których w odniesieniu do ludzi nie można lekceważyć. Jednak od rozwiązania podstawowych problemów komunikacji i synchronizacji w zespołach ludzkich w decydującej mierze zależy ich efektywność.

Bezдушny komputer nie wybacza programiście żadnego błędu, więc popycha go ku doskonałości. Młody programista zbliży się do tej doskonałości, jeśli się nie zniechęci – to jednak jest już rola nauczyciela programowania, a nie komputera. Bardzo ważny jest taki dobór tematów do przykładowych programów, które dają w efekcie atrakcyjny wynik. Lepiej, jeśli w wyniku programu napisanego przez ucznia w szkole pojawi się animacja obiektu graficznego na ekranie, niż pierwiastki równania kwadratowego. Programowanie powinno być w systemie edukacyjnym widziane jako narzędzie do wyrabiania pewnych ważnych umiejętności, a nie wyłącznie jako czynność techniczna. Gdyby to ode mnie zależało, to wprowadziłbym obowiązkowe nauczanie programowania dla studentów prawa. W myśl zasady – jeśli nie umiesz logicznie myśleć, to nie możesz być prawnikiem. Jeśli twierdzisz, że umiesz, to udowodnij to programując, bo to jest najlepszy test. Jestem przekonany, że gdyby prawnicy przeszli przez taką formację, to w polskim prawie nie byłoby tylu luk, sprzeczności i niespójności.