

RYSZARD TADEUSIEWICZ

Akademia Górniczo-Hutnicza im. S. Staszica w Krakowie

Nauczanie podstaw informatyki na studiach wyższych

Programowanie maszyn cyfrowych staje się podstawowym zajęciem coraz liczniejszej grupy ludzi różnych specjalności. Komputery tak dalece przeniknęły do nauki, techniki i gospodarki, że wykorzystywanie ich i programowanie stały się koniecznością. Nie zawsze zresztą koniecznością rzeczywistą; często mamy tu do czynienia z wtórnym zafascynowaniem, kiedy to specjalista, jedynie chwilowo zajmujący się informatyką dla rozwiązania pewnego konkretnego zadania, daje się wciągnąć w nie mającą końca pogoń za mirażem, jakim jest doskonały program komputerowy. Narzędzie zaczyna przesłaniać cel, aż wreszcie samo staje się celem. Ośrodki obliczeniowe (głównie uczelniane, ale obawiam się, że nie tylko) stają się niemal wyłącznym polem działania licznych „naukowców hybrydowych”: mechaników-informatyków, lekarzy-informatyków, językoznawców-informatyków itp. Zamiast myśleć — czeka się na wyniki, zamiast konstruować stanowiska pomiarowe — buduje się programy, zamiast wykonywać doświadczenia — szuka się błędów. Komputery codziennie zadrukowują setki stron papieru, pożerają setki tysięcy kart dziurkowanych, obliczają i podają do wiadomości miliony liczb, wykonują setki miliardów operacji matematycznych.

Czy zawsze te obliczenia mają sens? Czy wszystkie z trudem obliczone i tysiącami drukowane liczby stają się podstawą głębokich przemyśleń? Jak często wydruki z komputera służą jako parawan dla przedmiotowej ignorancji i braku koncepcji? Ile spośród licznych błyskotliwych

zastosowań komputera ma charakter istotnej potrzeby, a ile wynika z mody na pseudonowoczesność?

Na żadne z tych pytań na pewno nie znajdziemy odpowiedzi. Niemniej konieczne wydaje się ich postawienie właśnie w kontekście problemu nauczania informatyki, gdyż to, czego, a także jak nauczamy studentów ma zasadniczy wpływ na ich późniejszą postawę wobec możliwości i problemów, jakim źródłem jest komputer. Niewątpliwie ogromna ilość osiągnięć współczesnej nauki i techniki jest bezpośrednim następstwem szerokiego wykorzystania komputerów, dlatego musimy upowszechnić wiedzę informatyczną wśród nieprofesjonalistów. Niewątpliwie także komputer stanowi dla każdego wykształconego człowieka źródło nieustannego wyzwania, porównywalnego jedynie z intelektualną atrakcyjnością szachów, stąd łatwość ulegania wskazanej wyżej fascynacji, ze wszystkimi jej ujemnymi następstwami.

Czy istnieje rozwiązanie wskazanego problemu?

Być może nie. Jest wielce prawdopodobne, że wynalazek komputera będzie miał podobne skutki, jak wynalazek druku, który obok ogólnie znanych kulturotwórczych funkcji spowodował zalew informacji pisanej, z którą obecnie nie potrafimy się uporać i który powoduje, że na przykład ten artykuł prawie na pewno nie dotrze do tych czytelników, do których jest w intencji autora adresowany. Być może spośród milionów liczb produkowanych przez komputery będziemy jeszcze mniej wiedzieli o otaczającej nas rzeczywistości, którą odzwyczailimy się postrzegać własnymi zmysłami, bez pomocy maszyny. Być może.

Zanim jednak nie jest jeszcze za późno możemy przynajmniej starać się spodziewanemu złu zaradzić. Najbardziej celowe wydaje się tu działanie w zakresie nauczania podstaw informatyki, szczególnie w odniesieniu do nauczania tego przedmiotu na kierunkach studiów nie związanych bezpośrednio z elektroniczną techniką obliczeniową.

Stan nauczania podstaw informatyki. Trudności nauczania podstaw informatyki wynikają generalnie z trzech źródeł:

- braku jednolitej definicji zakresu przedmiotu;
- niedostatku środków technicznych;
- niskich kwalifikacji kadry nauczającej.

Jestem przekonany, że w stosunku do każdego z tych

źródeł można toczyć zacięte spory. Powołując się na istniejące i zatwierdzone programy studiów można wykazać, że dokładnie wiadomo, czego nauczać; argumentując liczbą godzin przeznaczonych w uczelnianych ośrodkach obliczeniowych na cele dydaktyczne (w szczególności na liczenie programów studenckich) można odrzucić jako niepoważne przypuszczenie o brakach sprzętowych; na koniec mając ogromnie liczną i utytułowaną kadrę nie możemy wątpić w najwyższy możliwy poziom nauczycieli.

Tematyka nauczania. Zadając większej liczbie ludzi zajmujących się techniką komputerową pytanie: czym jest informatyka?, przekonamy się, że większość z nich pytanie to wprawi w zakłopotanie, a pozostali będą podawali różne określenia, najchętniej przechodząc do konkretów w rodzaju: programowanie, budowa maszyn cyfrowych, konstrukcja systemów informacyjnych itp. Analiza literaturowa tegoż zagadnienia dostarczy nam szybko kilkunastu przynajmniej definicji informatyki. Właśnie — kilkunastu, a nie jednej, precyzyjnej. Nie przeszkadza nam (to jednak pisać i mówić o informatyce tak, jakby to była tak samo konkretna dziedzina, jak elektrotechnika czy wytrzymałość materiałów.

Naturalnie, omówiony stan ma swoje przyczyny, w szczególności w tym, że informatyka liczy sobie na serio niewiele więcej niż 20 lat. Chcąc jednak nauczać, musimy mieć sprecyzowany pogląd na zakres nauczanego przedmiotu, w związku z czym w programach studiów, a także w publikowanych podręcznikach wybierane są pewne zagadnienia z zakresu programowania, struktury i organizacji logicznej maszyn cyfrowych, teorii informacji, teorii kodów, technologii projektowania systemów informatycznych i innych, pokrewnych zagadnień. Wyraża się przy tym mniej lub bardziej jawną nadzieję, że suma tych przedmiotów stanowi jakąś reprezentację informatyki. Nadzieja ta jest jednak zawodna, szczególnie w przypadku, kiedy nauczanie informatyki jest prowadzone na kierunkach studiów, na których przedmiot ten ma znaczenie pomocnicze, uzupełniając wiedzę zawodową przyszłego inżyniera metalurga, mechanika, budowlanca itp. Liczba wiadomości praktycznych i teoretycznych zawartych w nieprecyzyjnie rozumianym pojęciu informatyki stoi w tak rażącej dysproporcji do liczby godzin, jaką można poświę-

cić na jej nauczanie, że wykładany program staje się żalosnym kompromisem pomiędzy szerokimi zamiarami a realiami — takimi jakie są. Najczęstszy obraz tego kompromisu sprowadza się do nauczania (bardzo pospiesznego i nieporządnego) programowania w jednym z dostępnych języków wyższego rzędu. Zwykle językiem tym jest *FORTRAN* (bo najszerzej znany, prosty i efektywny), w bardziej wyrefinowanych układach sprzętowych — na przykład *JEAN* (bo konwersacyjny i łatwo go uczyć), niekiedy *BASIC* (dostępny na minikomputerach), gdzieś tam *Algol* (bo elegancki), aktualnie *Pascal* (bo modny).

Jestem zdania, że niezależnie od tego, jakiego języka uczymy, popełniamy błąd, mszczący się potem zatłoczonymi ośrodkami obliczeniowymi, zalewem kiepskich programów i wspomnianym na wstępie modelem „specjalistów hybrydowych”. Można postawić tezę, że dlatego tak wielu ludzi obecnie programuje, że jedynie nieliczni potrafią naprawdę wykorzystywać komputery. Powszechnie stosowany system nauczania informatyki zbyt wiele uwagi poświęca programowaniu, zbyt zaś mało wykorzystywaniu oprogramowania bibliotecznego, umiejętności precyzyjnego formułowania problemów w postaci algorytmów (zalecanych ewentualnie później do oprogramowania specjalistom), prawie w ogóle nie mówi się o metodyce informatycznej analizy problemów z uwzględnieniem charakteru danych wejściowych i pożądanej formy wyników.

Nie potrafię naturalnie podać optymalnego modelu tematyki nauczania informatyki, jestem jednak głęboko przekonany, że konieczna jest szeroka dyskusja nad takim modelem. W szczególności jestem zdania, że mając do dyspozycji jeden semestr zajęć „Elektronicznej techniki obliczeniowej” jest niedopuszczalną rozrzutnością poświęcać go w całości na zmagania z kolejnymi instrukcjami i konstrukcjami języka *FORTRAN*, przy całkowitym praktycznie pominięciu wszelkich innych problemów informatyki. Znacznie bardziej celowe jest zapoznanie studenta z własnościami maszyny i metodologią programowania za pomocą maksymalnie prostego języka programowania, dysponującego bogatą diagnostyką i na tyle prostą składnią, aby trudności „ortografii” języka programowania nie dodawały się do trudności, jakie nieuchronnie każdy zapoznający się z programowaniem musi napotkać: trudności zderzenia sta-

tycznej postaci zapisu programu z dynamiczną akcją maszyny, wykonywaną w wyniku realizacji programu. Wstępną propozycją takiego języka jest opisany w pracy (1) język *MINI*, bazujący na języku polskim, oparty na dziewięciu zaledwie prostych składniowo formatach instrukcji, a dający możliwości obliczeniowe porównywalne z *BASIC-iem*¹. Czas zaoszczędzony dzięki zastosowaniu prostego języka programowania może i powinien być wykorzystany dla nauczania adeptów informatyki sposobu posługiwania się maszyną i jej bibliotecznym oprogramowaniem, stawiania problemów pod kątem i na miarę techniki obliczeniowej, planowania użycia maszyny, na koniec wreszcie — rozumienia ograniczonej techniki informatycznej.

Środki techniczne, wykorzystywane w nauczaniu Informatyki. W większości wyższych uczelni podstawowym środkiem technicznym wykorzystywanym w nauczaniu informatyki jest duża maszyna cyfrowa, z reguły pracująca w reżimie wsadowym, na której między innymi wykonywane są programy studenckie. Ponieważ programów studenckich jest dużo (zazwyczaj, i słusznie, żąda się, aby każdy student napisał i uruchomił przynajmniej jeden program), a jednocześnie są to programy dość krótkie (czas ich wykonania dodatkowo skracany jest wskutek nieuchronnie występujących błędów), przeto stosuje się zwykle obsługę programów z wykorzystaniem wsadowego systemu operacyjnego, podnoszącego sprawność przetwarzania. Wszystkie te środki są konieczne ze względu na potrzebę ograniczenia i tak wysokich kosztów obsługi programów dydaktycznych. Powodują jednak owe środki, że student jest odsunięty od komputera bardzo daleko, mając do czynienia praktycznie wyłącznie z plikami kart zawierającymi program i dane oraz z wydrukami zawierającymi wyniki lub (częściej) diagnostykę błędów. Można wskazywać na korzystne strony takiej sytuacji, polegające na przystosowaniu przyszłego użytkownika komputera do warunków, w jakich zazwyczaj przyjdzie mu działać po studiach, jednak proces osvajania z techniką obliczeniową, a następnie proces wytwarzania korzystnych przyzwyczajeń w korzystaniu z komputera

¹ Translatory tego języka dla maszyn Odra 1300 i Cyber 72 udostępniła się nieodpłatnie każdemu zainteresowanemu.

mogą być wydatnie przyspieszone w wyniku możliwości korzystania z maszyny w reżimie konwersacyjnym.

Stwierdzenie tego faktu skłania do ogólniejszej refleksji. Otóż generalnie komputer w uczelni wykorzystywany jest w trzech kierunkach: do prac naukowych, prac administracyjnych (przetwarzanie danych dla potrzeb uczelni) oraz do dydaktyki. Te trzy kierunki zastosowań różnią się zasadniczo wymaganiami, jakie z nich wynikają odnośnie do konfiguracji systemu komputerowego, tymczasem z reguły obsługiwać je musi jeden system cyfrowy, gdyż koszty sprzętu informatycznego są nadal niezwykle wysokie. Charakterystyczne jest przy tym, że w tej sytuacji większość uczelni posiada sprzęt przystosowany głównie do prac naukowych (duża i szybka jednostka centralna ze skromnym wyposażeniem w urządzenia wejścia-wyjścia, zwłaszcza interakcyjne i ograniczoną liczbę pamięci masowych). Jest to sytuacja nieprawidłowa, gdyż analiza wykorzystania czasu maszyny dowodzi, że dydaktyka i zadania studenckie zajmują więcej czasu, niż prace naukowe, przy czym te ostatnie także, przynajmniej w części, mogą być wykonywane na maszynie dostosowanej swą architekturą do zadań dydaktycznych.

Wydaje się, że pilną potrzebą chwili jest dokładne przeanalizowanie wymagań i potrzeb, jakie dydaktyka stawia sprzętowi komputerowemu i podjęcie starań, aby kolejne instalowane w ośrodkach akademickich systemy cyfrowe spełniały te wymagania. Naturalnie, nie potrafię zaproponować tu optymalnego modelu struktury systemu cyfrowego przeznaczonego do celów dydaktycznych, wydaje się jednak możliwe wskazanie kilku przynajmniej cech pożądanых dla takiego systemu. Po pierwsze, konieczne jest wyposażenie umożliwiające wielodostępną interakcyjną pracę licznych grup studenckich. Ze względu na to, że początkujący programiści popełniają wyjątkowo dużo błędów, a także dlatego, że celem pisanych i uruchamianych przez studentów programów jest nauka, a nie uzyskanie określonych wyników obliczeń, których wydruk ma być wykorzystany w charakterze dokumentu, celowe jest szerokie stosowanie monitorów ekranowych zamiast urządzeń drukujących. Rozwiązanie takie, jakkolwiek sprzętowo droższe, wydaje się być ekonomicznie uzasadnione po obliczeniu kosztów niepotrzebnie zużywanego papieru, zadrukowywa-

nego wydrukami licznych błędnych kompilacji oraz błędnych wykonań studenckich programów. Problem liczby końcówek postulowanego systemu musi być rozważany indywidualnie, z uwzględnieniem programu studiów, liczby studentów i możliwości lokalowych uczelni. Niecelowe jest bowiem grupowanie większej liczby końcówek w jednym pomieszczeniu, jako że praca z komputerem bardziej niż jakakolwiek inna forma działalności umysłowej wymaga skupienia, łatwego do osiągnięcia w odosobnionej kabinie i trudnego w sali grupującej kilkanaście osób.

Na podkreślenie zasługuje fakt, że pomimo potrzebnego wielodostępnego konwersacyjnego trybu pracy maszyna cyfrowa obsługująca dydaktykę uczelni nie musi być zbyt szybka ani nie musi posiadać wyrafinowanego systemu operacyjnego, gdyż pomimo licznych końcówek tempo wprowadzania informacji nie jest zbyt duże, jako że początkujący studenci redagują swoje programy znacznie wolniej i z większą liczbą poprawek, niż doświadczeni programiści. Podobnie, czas reakcji systemu, jakkolwiek pożądanym najkrótszy, może być wydłużony w stosunku do czasu wymaganego w systemach profesjonalnych.

Po drugie, celowe jest wyposażenie dydaktycznego systemu komputerowego w specjalizowane oprogramowanie służące usprawnieniu dydaktyki, ze szczególnym uwzględnieniem możliwości przekazania maszynie czynności kontrolnych. Nie dotykając tu ogromnego i stale aktualnego tematu komputerowego wspomagania nauczania innych przedmiotów, musimy stwierdzić, że w zakresie nauczania podstaw programowania istnieją ogromne możliwości podniesienia sprawności nauczania, przy równoczesnym zmniejszeniu pracochłonności. Niektóre z tych możliwości wiążą się z faktem, że w nauczaniu programowania zawsze następuje okres, kiedy student musi sprawdzić swoje umiejętności przez samodzielne napisanie i uruchomienie programu na zadany temat. Zadania stawiane studentom w tym zakresie dotyczą zwykle prostych zagadnień, egzemplifikujących poszczególne problemy i techniki programowania, a wynik otrzymany z programu studenckiego służy jedynie kontroli, czy student potrafi zadanie rozwiązać. Wydaje się w tej sytuacji możliwe i celowe powierzenie kontroli maszynie, która może być wyposażona w zbiór (jak największy i ustawicznie przez grono nauczające poszerzany)

zadań wraz z rozwiązaniami pozwalającymi na weryfikację poprawności programu studenckiego. Oczywiście, budowa takiego zbioru „etiud” programowych, a zwłaszcza zaprojektowanie skutecznych testów badających poprawność i optymalność (na przykład czas obliczeń lub liczba instrukcji) programów studenckich wymaga sporego wysiłku i pomysłowości. Zaletą takiego systemu jest jednak pozostawanie wypróbowanych zadań w „banku danych” komputera i stałe poszerzanie ich zasobu w miarę eksploatacji systemu, a także łatwiejsze ujednoczenie programu i wymagań dla grup studenckich prowadzonych przez różnych asystentów. Ewentualna wada, polegająca na powtarzaniu się zadań w kolejnych latach studiów i wynikająca stąd możliwość prosperowania „giełdy” studenckiej, jest możliwa do przewyciężenia drogą poszerzania zasobów „banku zadań”; wada ta jest zresztą aktualna także i w obecnym systemie prowadzenia zajęć.

Trzecią, bardziej odległą potrzebą w zakresie sprzętu informatycznego służącego dydaktyce podstaw informatyki jest potrzeba urządzeń specjalistycznych, przeznaczonych wyłącznie do celów dydaktycznych. Jako dydaktyk, prowadząc zajęcia z zakresu elektronicznej techniki obliczeniowej, odczuwam niejednokrotnie zakłopotanie wywołane konfliktem pomiędzy potrzebą poglądowego przedstawienia określonych konstrukcji programowych lub postaci komputerowej diagnostyki błędów (do czego typowo obecnie używa się tablicy lub rzutnika pisma) a kameralnością oryginalnych dokumentów, produkowanych przez komputerowe urządzenia wyjściowe. Wydruk drukarki lub obraz na ekranie typowego monitora mogą wygodnie oglądać 2-3 osoby. Przy grupie powyżej sześciu osób jest to praktycznie niemożliwe, zwłaszcza jeśli dydaktyk musi wydruk komentować, wskazując jego poszczególne elementy.

Aktualnie trudność tę usiłuje się rozwiązywać kilkoma sposobami, z których każdy ma istotne wady. Najczęściej obieraną drogą jest dążenie do zmniejszania liczebności grup studenckich. Jest to rozwiązanie z wielu punktów widzenia korzystne, pozwala między innymi zbliżyć się do ideału, jakim jest zindywidualizowane nauczanie praktyczne każdego studenta, a także ułatwia operowanie grupą studencką bezpośrednio w ośrodku obliczeniowym, w razie prowadzenia zajęć bezpośrednio przy maszynie. Jest to jednak rów-

nocześnie rozwiązanie bardzo kosztowne, gdyż wymaga (w przeliczeniu na tę samą liczbę studentów) znacznie większej liczby godzin pracy nauczycieli, a także kłopotliwe w warunkach ograniczonej liczby sal ćwiczeniowych. Sytuacja mogłaby jednak ulec radykalnej poprawie, gdyby możliwe było zainstalowanie w salach ćwiczeniowych specjalizowanych urządzeń wejściowo-wyjściowych, przyłączonych *on-line* do użytkowanej maszyny cyfrowej. Specjalizacja wspomnianych urządzeń powinna polegać na wykorzystaniu wielkogabarytowych wyświetlaczy, zastępujących konwencjonalną tablicę. Wyświetlacz taki, oparty na technice diod *LED*, ciekłych kryształów czy wreszcie — dostatecznie jasny monitor kineskopowy i układ optyczny umożliwiający rzutowanie obrazu na ekran, może oddać nieocenione usługi i pomimo swoich kosztów — może być opłacalny z uwagi na obniżenie kosztów eksploatacji. Jest to jednak rozwiązanie, jak wspomniano na wstępie, bardzo odległe w czasie.

Kadry nauczające informatyki. Rozpowszechnione jest mniemanie, że silną stroną aktualnie realizowanych systemów nauczania informatyki, przy wszystkich sprzętowych i programowych niedogodnościach wymienionych wyżej, są kadry nauczające: kompetentne, ofiarne i fachowe. Podejmując ten temat w niniejszym, z założenia polemicznym artykule, zdaję sobie doskonale sprawę z jego trudności i z kontrowersyjności wszelkich sądów, jakie tu zostaną przytoczone. Nikt bowiem, a już najmniej niżej podpisany, nie może nie zauważać ofiarności i fachowości nauczycieli akademickich zajmujących się nauczaniem informatyki. Jeśli jednak jest tak dobrze, to dlaczego jest tak źle? Wydaje się bowiem, że za przytoczony na wstępie niniejszego artykułu obraz nieprawidłowego korzystania z techniki obliczeniowej przez „specjalistów hybrydowych” ponoszą odpowiedzialność także nauczający, a może nawet głównie oni.

Wydaje się (jakkolwiek jest to oczywiście sąd subiektywny), że nauczyciele (przynajmniej niektórzy) są w niewłaściwy sposób fachowi i zana d t o ofiarni. Aby poszerzyć tę tezę należy zastanowić się, jakie kwalifikacje powodują, że określona osoba cieszy się zasłużonym uznaniem jako wybitny informatyk. Myślę, że nie popełnię dużego błędu, jeśli wskażę trzy grupy tego typu specjalistów.

Pierwsi z nich to teoretycy. Są oni autorami oryginalnych prac z zakresu dowodzenia poprawności programów,

organizacji procesów współbieżnych czy teorii translacji. Prace te, cieszące się zasłużonym uznaniem i reprezentujące bardzo wysoki poziom naukowy, są tak teoretyczne i zanurzone w tak zaawansowanej matematyce, że pomiędzy nimi a codziennym stosowaniem maszyny matematycznej jest przepaść. W istocie, jakkolwiek „nie ma nic bardziej praktycznego, niż dobra teoria”, to jednak wybitny teoretyk nie musi *eo ipso* być dobrym praktykiem, jeszcze mniej może mieć wspólnego z umiejętnością nauczania podstaw praktycznego wykorzystywania techniki obliczeniowej.

Drugą grupę tworzą utalentowani programiści. Niewątpliwie, jak w wielu innych dziedzinach ludzkiej działalności, tak i w programowaniu komputerów obok wiedzy oraz rzemiosła konieczna jest pewna wrodzona predyspozycja, powodująca, że ten sam program, na ten sam temat, jeden specjalista napisze szybko i sprawnie, a inny wypracuje mozolnym trudem, przy czym ostateczny efekt w drugim przypadku będzie wielokrotnie gorszy niż w pierwszym. Niewątpliwie, pisanie błyskotliwych programów jest sztuką, a ci, którzy ją posiadli, cieszą się zasłużonym szacunkiem i uznaniem środowiska. Czy oznacza to jednak, że stanowią oni optymalny model nauczyciela? Myślę, że można z tą opinią polemizować. Programista, nawet wybitny talent, będzie stawiał programowanie i rozliczne „sztuki”, jakimi można w zaskakujący sposób usprawnić proces obliczeń, ponad wszelkie inne zajęcia z zakresu techniki obliczeniowej. Będzie rozbudzał ambicje, dla których zaspokajania nie ma miejsca w codziennej pracy użytkownika środków informatyki, będzie stwarzał sytuację, w której uczeń przystępując do rozwiązania zadania będzie w głównej mierze poszukiwał własnego programu, zamiast korzystać z oprogramowania bibliotecznego, przesunie środek ciężkości z problemu „po co komputer stosować”, na problem „jak komputer programować” — problem, który może i powinien być wtórnym, mniej ważnym.

Trzecia grupa cieszących się poważaniem informatyków złożona jest z projektantów rozmaitego typu systemów informatycznych; na ogół współcześnie systemów dość złożonych i „dziwaczkich”, jako że systemy typowe, dotyczące rutynowanych zastosowań, zostały dawno opracowane, zbadane i oprogramowane — wystarczy je kupić lub przystosować. Właśnie ten fakt, że systemy typowe, stanowiące

„chleb codzienny” użytkownika komputera są nisko notowane wśród naukowców, powoduje, że także i w dydaktyce zwraca się większą uwagę na ciekawostki rozwiązań nietypowych i trudnych, poświęcając zbyt mało uwagi systemom typowym i ich stosowaniu. Zapomina się przy tym, że to, co dla specjalisty stanowi elementarz, wstęp do właściwej pracy, coś tak oczywistego, że aż niewartego uwagi — to właśnie jest z reguły najbardziej potrzebne początkującym studentom, i to właśnie będzie treścią ich działań po studiach. Tak więc również i ta grupa wybitnych informatyków nie zawsze sprawdza się w dydaktyce, gdyż ich własna fachowość zniekształca proporcje potrzebne w pracy nauczycielskiej.

* * *

Obraz kadry dydaktyków nauczających podstaw informatyki jest celowo przedstawiony w krzywym zwierciadle. Problemy przejasniono i celowo pominięto nader liczne przypadki, kiedy wybitny teoretyk czy praktyk potrafi także być doskonałym dydaktykiem. Można nawet twierdzić, że większość nauczycieli akademickich potrafi znajdować przejście pomiędzy wysokim zaawansowaniem prowadzonych prac naukowych a niezbędną przystępnością i poprawnym zakresem wiadomości przekazywanych studentom. Celem tej dyskusji było jednak wskazanie, że nie można automatycznie przypisywać wysokich kompetencji dydaktycznych na bazie ogólnie znanej wysokiej fachowości naukowej oraz widocznego zaangażowania i ofiarnej pracy na polu dydaktyki. Wiele praktycznych obserwacji potwierdza tezę, że dobre efekty dydaktyczne w zakresie podstaw informatyki łatwiej jest osiągać bez obciążenia, jakim jest w tym przypadku wybitna fachowość w zakresie prac naukowych z zakresu informatyki. Sytuacja jest diametralnie inna przy nauczaniu informatyki na znacznie wyższym poziomie, na specjalistycznych kierunkach studiów i w odniesieniu do wybranych grup studentów, to zagadnienie nie jest jednak w tym artykule omawiane.

Literatura

1. Tadeusiewicz R. — *Język MINI jako propozycja w zakresie nauczania podstaw informatyki*. „Informatyka” 1977, nr 2.

2. INFOGRYF'78 — *Informatyka w dydaktyce*. Materiały konferencji, Kołobrzeg 1978.
 3. Bauer F. L., Goos G. — *Informatyka*. WNT Warszawa 1977.
 4. Tadeusiewicz R. i in. — *Elektroniczna technika obliczeniowa*. Skrypty uczelniane AGH nr 683 (część I), 609 (część II), 639 (część III), 552 (część IV), Kraków 1977 - 1979.
-