PL ISSN 0137 — 8929
Nr indeksu 38 142

wiedza i życie

3 1985

W początkach techniki komputerowej podstawowym sposobem porozumiewania się człowieka z komputerem był tzw. język wewnętrzny maszyny. Tworzyły go — zapisywane zwykle cyfrowo, w ósemkowym systemie liczbowym — kody odpowiadające sygnałom, jakie musi otrzymać układ sterowania maszyny by wykonywać potrzebne czynności. Nie warto wymieniać wszystkich powodów, dla których ten sposób programowania uważano jednak za niekorzystny. Można wskazywać na pracochłonność i małą czytelność programów, można podnosić argument, że język wewnętrzny każdego komputera jest inny i to utrudnia wymianę programów pomiędzy ośrodkami posługującymi się różnymi maszynami, można też twierdzić, że główną przyczyną niedogodności jest duże prawdopodobieństwo popełnienia błędu w programie pisanym w języku wewnętrznym. Dla mnie jednak najbardziej przekonywujący jest motyw psychologiczny: otóż było wprost nieprzyzwoite, by człowiek — pan i twórca komputera musiał uczyć się jego języka, jeśli chce mu przekazać swoją wolę. Niech raczej ta maszyna uczy się naszego języka!

Nie ma na to dowodu, gdyż informatycy to ludzie poważni i lubią racjonalizować powody wszelkich działań, ale gotów jestem się założyć, że powody budowy licznych języków programowania były właśnie takie: przywrócenie właściwego podziału ról pomiędzy człowiekiem, a jego dziełem.

Jak było, tak było — liczy się efekt. Obecnie przy kontaktach z komputerem, przy stawianiu mu zadań i przy programowaniu — wykorzystuje się języki specjalne, nazywane językami algorytmicznymi. Pojawia się jednak pytanie, dlaczego stosuje się te sztuczne, wydumane języki algorytmiczne, a nie najzwyczajniej w świecie język naturalny. Otóż trudno w to uwierzyć, ale z punktu widzenia potrzeb programowania komputerów język naturalny

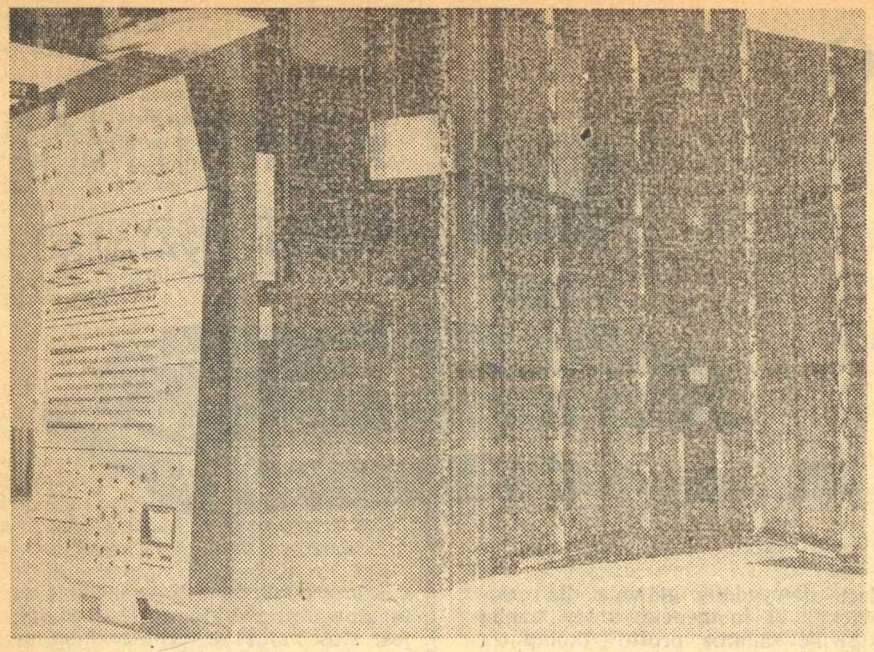
Cz. V

O KOMPUTERACH PRAWIE WSZYSTKO

jest niewygodny głównie dla człowieka! W momencie, kiedy trzeba stawiać zadania prosto (komputer nie jest tak inteligentny i domyślny jak człowiek!) precyzyjnie i konkretnie — język naturalny okazuje się zbyt mało ścisły, formułowane w nim polecenia są niejednoznaczne, a nade wszystko — jest on rozwlekły: wypowiedzenie najprostszego polecenia wymaga tak wielu słów, a każde słowo to na ogół wiele liter, że w sumie trzeba wprowadzać do maszyny ogromne ilości informacji. Bardziej opłaca się więc stworzyć sztuczny język — uboższy od naturalnego, ale pozwalający na precyzyjne wypowiadanie poleceń dla komputera.

Oczywiście, takiego języka trzeba się specjalnie uczyć — ale jest to cena, którą płacimy za wygodę użytkowania komputera. Cena niezbyt wygórowana — dodajmy zaraz — gdyż twórcy języków programowania maszyn cyfrowych dołożyli starań, aby nauczenie się języka nie było dla człowieka nadmiernie uciążliwe. W istocie większość języków algorytmicznych przewiduje użycie kilkunastu zaledwie słów (zwykle są to słowa angielskie, gdyż tradycją jest stosowanie w technice komputerowej elementów języka angielskiego), a także elementów notacji matematycznej (znaki działań, nawiasy, znaki równości i nierówności itp.), co ma swoje głębokie uzasadnienie. Skoro komputer jest maszyną liczącą, to naturalne jest użycie zapisów matematycznych przy stawianiu mu zadań do rozwiązania, gdyż w tej postaci zadania te są zwykle formułowane.

Języków algorytmicznych w sumie powstało sporo można nawet wyrazić obawę, że jest ich zbyt dużo: ta mnogość utrudnia wszak życie wszystkim, którzy chcą się tych języków nauczyć i ich używać. Równocześnie duża liczba różnych języków dowodzi jasno że żaden z nich nie jest naprawdę dobry i uniwersalny: gdyby taki dobry język powstał, zbyteczne by się stały automatycznie wszystkie inne! Tymczasem komputerowa „Wieża Babel" wciąż rośnie — nie ma prawie miesiąca, aby w prasie specjalistycznej nie pojawiło się doniesienie o kolejnym,



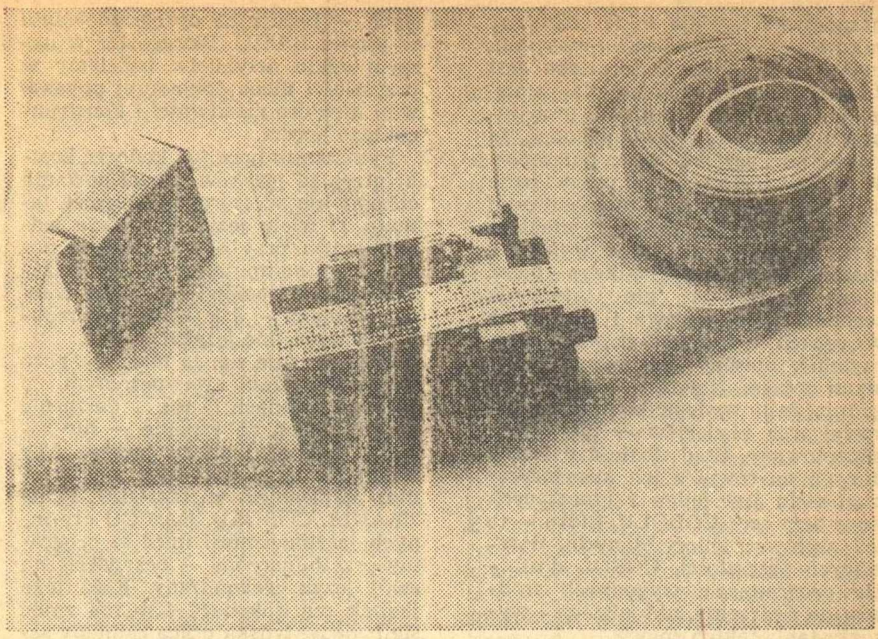
Komputer. Czy on się ma uczyć naszego języka, czy my musimy poznać jego język?

oczywiście rewelacyjnym, znakomitym, oryginalnym — języku programowania. Zgroza! Jeśli nie zwariują od tego informatycy, to pękną od nadmiaru komputery!

Oczywiście te biadania to żarty: każdy z powstających języków ma swoje uzasadnienie, zbiór zadań rozwiązywanych przy jego pomocy, grupę użytkowników pragnących się nim posługiwać i komputery zdolne go rozumieć i używać. Mimo tych uzasadnień istotne znaczenie ma tylko kilka języków i im właśnie chciałbym poświęcić parę słów.

Największe znaczenie w historii informatyki odegrał język ALGOL. Był on przystosowany do tworzenia programów typu obliczeniowego, matematycznych, naukowych — stąd krąg jego rzeczywistych użytkowników był ograniczony W dodatku komputery mało wydajnie wykorzystywały programy pisane w ALGOLU — czas obliczeń był duży, o wiele dłuższy niż czas obliczeń tego samego programu napisanego wprost w języku wewnętrznym komputera. Z tych powodów praktycy programowania odnosili się do ALGOLU nieufnie, zachwycał on natomiast teoretyków i naukowców — zresztą do dziś używa się ALGOLU do zapisu programów w publikacjach naukowych — nawet jeśli w rzeczywistości oprogramowanie realizuje się w innych językach. Decydują o tym zalety ALGOLU: duża czytelność pisanych w nim programów oraz ich „elegancja" — a względy estetyczne odgrywają w informatyce większą rolę, niż są to skłonni przyznać sami informatycy.

ALGOL powstał w wyniku wyrafinowanego procesu intelektualnego; utworzyła go, po licznych teoretycznych dyskusjach grupa naukowców pracujących pod auspicjami IFIP (międzynarodowej organizacji zaj-



O tym, czy program zawarty na taśmie będzie zrozumiały dla człowieka, decyduje język, w jakim go zapisano.

mującej się problematyką informatyki). Jego wielki konkurent i — co tu ukrywać — zwycięzca: język FORTRAN powstał na tym tle całkowicie odmiennie. Żartobliwie można określić, że pochodzi on z „nieprawego łoża"; jego koncepcja powstała jako skompletowanie i usystematyzowanie różnych „sztuczek" jakimi programiści ułatwiali sobie pracę. Patronował temu przedsięwzięciu koncern IBM i być może dlatego FORTRAN zyskał tak wielką popularność. Jako jego użytkownik muszę dostrzegać także i inne powody: jest to język naprawdę wygodny. Programy napisane w tym języku liczą się szybko, jest on dostępny dla większości komputerów — cóż z tego zatem, że nie jest tak elegancki, jak ALGOL?

FORTRANEM posługują się na świecie programiści rozwiązujący najrozmaitsze zadania, jego struktura predestynuje go do podobnego kręgu zastosowań, jak ALGOL; obliczenia numeryczne, zastosowania naukowe, prace inżynierskie. Tymczasem główny obszar zastosowań komputerowych to prace biurowe. Obliczenie parametrów jądra atomowego, ruchu gwiazd w Galaktyce lub konstrukcji rakiety kosmicznej to zadania jednorazowe, a rachunki za telefon, faktury sklepowe, ewidencja obrotów bankowych — potrzebne są codziennie... Dlatego ogromnie ważne są te narzędzia — w tym także języki programowania przystosowane do obsługi zadań ekonomicznych.

Językiem o takich właśnie własnościach jest COBOL. W stosunku do FORTRANU i ALGOLU jest on niewątpliwie bardziej „rozgadany". Program w języku COBOL przypomina tekst instrukcji napisanej w języku naturalnym (angielskim, chociaż norma dopuszcza też odmiany

narodowe i istnieją takie odmiany jak niemiecka, francuska i rosyjska), czytelnej dla każdego człowieka. Jest to celowe: taki czytelny dla każdego program budzi zaufanie urzędników i jest mniej prawdopodobne, że będzie on liczył nie to, co potrzebne. W COBOLU dużo uwagi przywiązuje się do eleganckiej, dopracowanej formy wydruków, będących efektem pracy maszyny. To także wynika z zastosowania: obsługiwanego przez biuro interesanta nie interesuje technika komputerowa i chce mieć w rękach dokument dopracowany, czytelny, opisany niezbędnymi tekstami objaśniającymi — słowem wygodny. Jeśli wydruk komputera nie spełni tych kryteriów — nie będzie akceptowany. Stąd dbałość o formę danych na równi z ich merytoryczną treścią jest jedną z rzucających się w oczy cech COBOLU. Biuro — to też archiwa: zbiory danych dotyczących przeszłości. COBOL dostarcza programiście wygodnych środków do tworzenia, aktualizacji i wykorzystania takich zbiórw.

Język BASIC powstał początkowo jako narzędzie do programowania prostych i szybko wykonywanych obliczeń Podstawową jego cechą była prostota użytkowania, był to ponadto z założenia język konwersacyjny, to znaczy programista mógł podczas jednej „sesji" przy monitorze komputera: napisać program, uruchomić go, sprawdzić uzyskane wyniki, dokonać modyfikacji programu, ponownie uruchomić, przeprowadzić obliczenia dla różnych danych, wykonać obliczenia według fragmentu, a nie całego programu — wedle woli, wedle potrzeb, wedle fantazji. Przy takiej pracy mniej ważna jest sprawność obliczeniowa i dlatego pierwsze wersje języka BASIC były pod tym względem niedopracowane. Tymczasem praktyka pokazała, że prostota BASIC przeważa nad walorami innych języków, że narzędzie zbudowane do doraźnych celów zyskało szerokie zastosowanie, że na koniec wszystkie typy i odmiany tak zwanych komputerów osobistych — bazują na oprogramowaniu w języku BASIC. Dlatego też w naszym cyklu artykułów będziemy w przyszłości nieco dokładniej pokazywali niektóre możliwości i konstrukcje tego języka.

Tymczasem pora już kończyć krótki przegląd języków programowania chociaż nie wyczerpałem jeszcze tematu. O wielu językach będzie jednak mowa przy okazji różnych zastosowań komputera. Pozostaje zatem tylko do omówienia nader ważny problem — jak komputer rozumie te wszystkie języki?

Otóż okazuje się, że maszyna poza swoim językiem wewnętrznym, o którego wadach tak obszernie wspominałem na wstępie — nie rozumie żadnego innego języka. Aby komputer mógł korzystać z programów napisanych w ALGOLU, FORTRANIE, COBOLU, BASICU czy innych językach musi mieć te programy przetłumaczone — oczywiście na swój język wewnętrzny. Któż tego jednak ma dokonać? Kto ma zająć się benedyktyńską pracą zamiany setek instrukcji programu w jednym języku na ten sam program — ale w innym (wewnętrznym) języku? Odpowiedź jest oczywista, a jednak zaskakująca: sam komputer! Na pierwszy rzut oka jest w tym sprzeczność: jak może tłumaczyć, skoro nie rozumie? A jeśli rozumie, to po co tłumaczenie?

Musimy wrócić raz jeszcze do podstawowej prawdy: komputer jest tylko „opakowaniem" dla programów. Jeśli potrafimy — a potrafimy dziś bardzo dobrze — napisać program tłumaczący, to ten program będzie tłumaczył inne programy. Taki program tłumaczący nazywa się translatorem. Jeśli maszyna ma w swoim wyposażeniu translator określonego języka, to może się nim posługiwać: komputer będzie miał to, co lubi; swoje kody wewnętrzne. Proste? Z pozoru tak, ale proszę sobie wyobrazić, jak trudno jest napisać dobry translator!

RYSZARD TADEUSIEWICZ