

I NFORMATYKA
K OMPUTERY
S YSTEMY



CENA — 50 zł

Dodatek „Żołnierza Wolności” nr 7/1986 ISSN 0860—2794



UWAGA KONKURS!

Foto: Jan Zelman



Mamy kolejny jubileusz, minęło dwadzieścia pięć lat wojskowej informatyki, stąd okazja do sentymentalnych wspomnień. Przyznam się, że z tych pierwszych lat niewiele pamiętam. Kiedy w Wojskowej Akademii Technicznej powstawały konstrukcje pierwszych maszyn liczących przygotowywałem się do konkursowego egzaminu na pierwszy rok studiów w szkole podstawowej. „Jaga” była dla mnie czarnym charakterem z przedszkolnej literatury — dopiero parę dziesiątek lat później dowiedziałem się, że „Jaga” to również udana konstrukcja dwóch młodych naukowców z Katedry Automatyki i Teleszerowania WAT — Jarosińskiego i Gajewskiego.

Kto dziś pamięta BINUZa —

szeregową, jednoadresową maszynę liczącą z arytmetyką binarną i uzupełnieniem stałoprzecinkowym? I jeszcze jedno wspomnienie: dodawanie i odejmowanie trwało 7 sekund, mnożenie 35, a dzielenie aż 42 sekundy! — to parametry przekątnikowej maszyny cyfrowej skonstruowanej przez kapitana Stanisława Jarosińskiego w 1959 roku w WAT. Na tej maszynie zawodowe ostrogi zdobywali pierwsi informatycy w mundurach. To oczywiście nie wszystko: były jeszcze niezwykle udane konstrukcje maszyn analogowych zespołu, wtedy również kapitana, Józefa Kapicy. Pierwszy był ELWAT-1 przeznaczony do syntezy i analizy układów dynamicznych, później konstrukcję tę wielokrotnie modyfi-

kowano. Powstała również rodzina komputerów — łączących cechy maszyn cyfrowych i analogowych.

Tyle wspomnienia. Jubilatów zaś życzymy jeszcze wielu, wielu udanych pomysłów, a przede wszystkim krótkiej i łatwej ich realizacji.

Nie krzepmy się jednak wspomnieniami, informatyka jest bowiem dyscypliną, która nie znosi bezczynności. To, co wiemy dzisiaj już jutro może nie wystarczać. Zapowiadamy zatem kilka zmian; przede wszystkim od pierwszego numeru przyszłego roku zmieniamy wianę. Nowa szata graficzna, z pewnością ładniejsza, i rozszerzenie tematyki to zapowiedź na najbliższe miesiące — zgodnie z życzeniem Czytelników rejestrować będziemy na naszych łamach wszelkie nowości.

Mamy nadzieję, że poprawi się również kolportaż „IKS-a”, znikną kłopoty z jego zakupem.

To oczywiście nie wszystko, inne zmiany już wewnątrz niniejszego numeru.

WIESŁAW CETERA

Infogryf '86

Planowy, i niestety rzeczywisty, przyjazd pociągu relacji Warszawa — Kołobrzeg godz. 5.05! (bo niby dlaczego pasażerowie mieliby opuszczać wygodny wagon sypialny o 7 lub 8 godzinie...). Mały spacer „Kołobrzeg nocą” i już o godz. 10 dnia 21.10.86 r. rozpoczął się VIII INFOGRYF.

Jest to chyba największe krajowe forum informatyków związanych z teorią i praktyką wdrażania komputerów w przemyśle, i nie tylko.

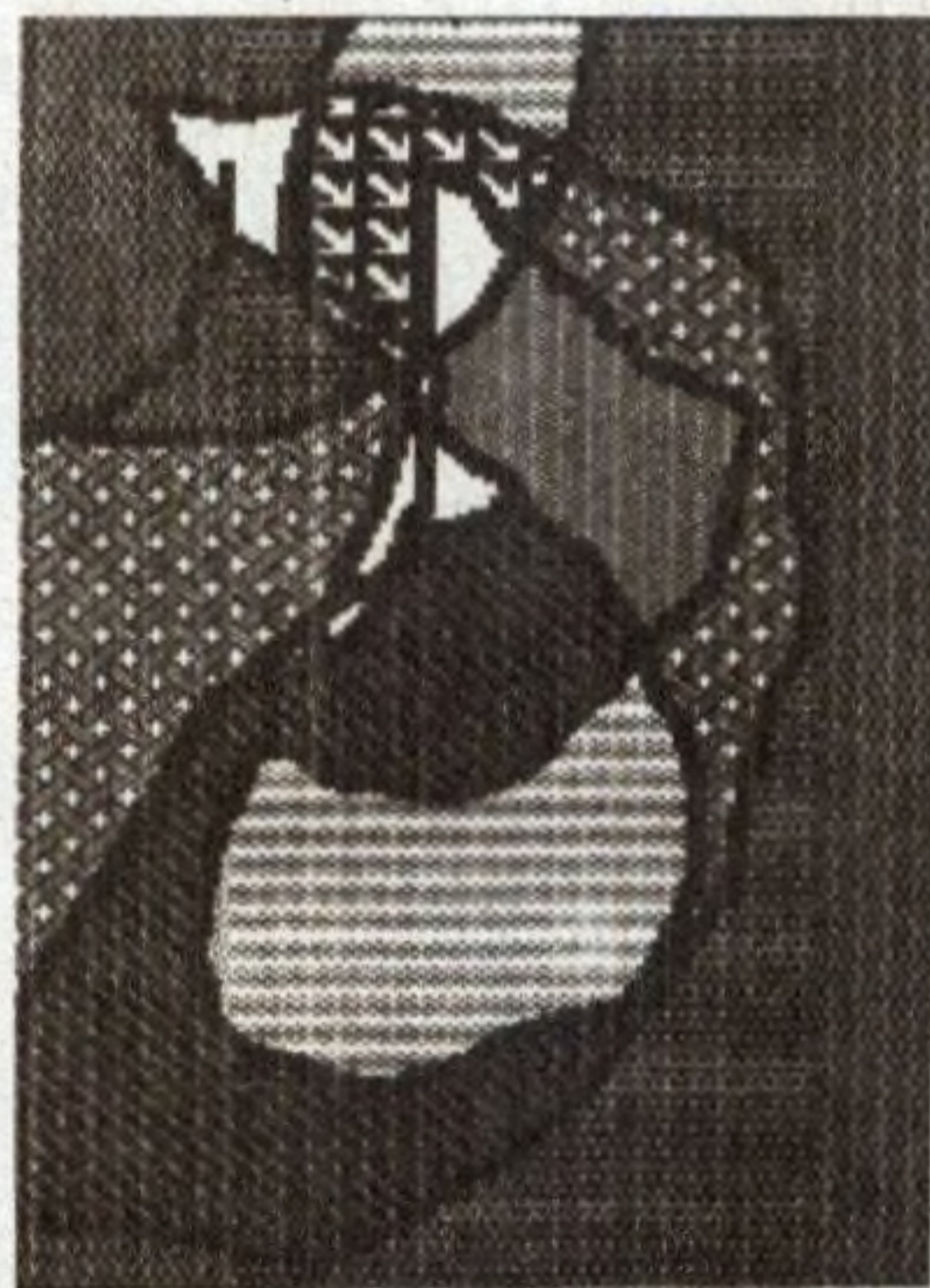
Czternaście lat temu przewodniczący Sekcji Informatyki TNOiK Szczecin — Edward Kolbusz — rozpoczął organizację pierwszego Infogryfu. Nazwę tę zaproponował (na trasie Warszawa — Szczecin), ówczesny dyrektor Zeto Szczecin, Zdzisław Bogdanowicz. Współorganizatorami tej szacownej już dzisiaj imprezy były szczecińskie instytucje: Politechnika, TNOiK i ZETO. Od V Infogryfu do grupy tej dołączyło Zeto Koszalin.

Po raz pierwszy w tym roku zamiast Politechniki, współorganizatorem Konferencji został Uniwersytet Szczeciński. Jest to wynikiem przejścia jej duchowego przywódcy profesora Tadeusza Wierzbickiego wraz z Wydziałem Ekonomicznym z Politechniki do Uniwersytetu Szczecińskiego.

INFOGRYF'86 otworzył prof. Wierzbicki witając serdecznie uczestników konferencji, w tym gości z NRD i Francji. Jednocześnie dał wyraz szczeremu ubolewaniu, że nie może powitać przedstawicieli władz decydujących o kształcie polskiej informatyki, ponieważ nie zaszczyciły swoją obecnością konferencji.

Wykład inauguracyjny, który okazał się felietonem, wygłosił sekretarz naukowy Infogryfu dr hab. Wojciech Olejniczak. Wzbudzał on śmiech, smutek i inne bardzo kontrowersyjne odczucia. Felieton ten dotyczył bowiem EFEKTU SKALI, który, definiując przez przykład, polega na tym, iż nagłe zniknięcie długopisów mogło by doprowadzić do katastrofalnych następstw, natomiast

gdyby to (zniknięcie) dotyczyło komputerów (wszystkich w kraju) negatywne skutki byłyby znacznie mniejsze — nie ta skala?



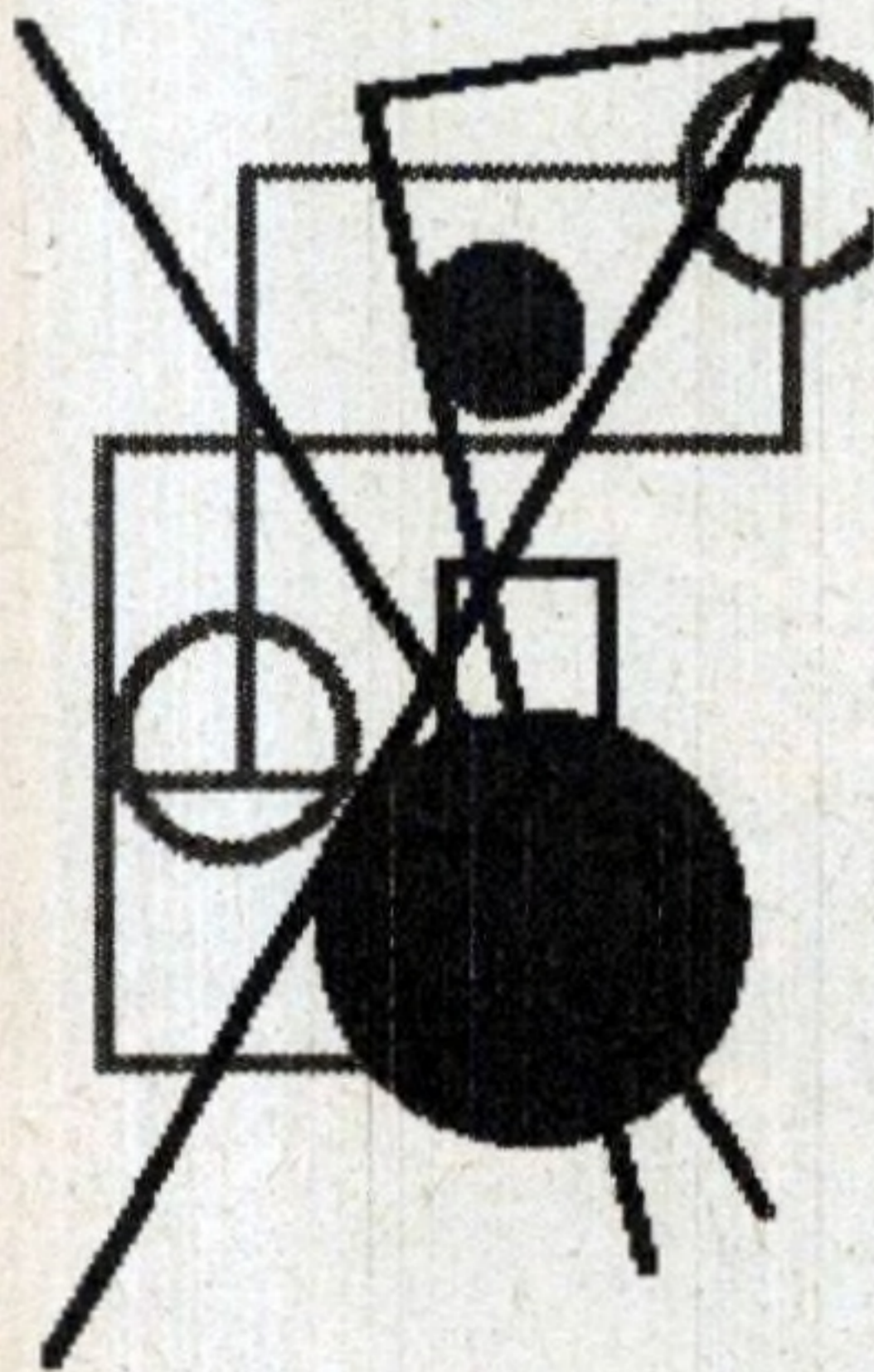
- Obrady toczyły się w trzech sekcjach:
- I. Systemy, metody i techniki tworzenia systemów informatycznych.
 - II. Komputerowe systemy wspomagania decyzji.
 - III. MIKROSKALA'86 — zastosowanie mikrokomputerów w przedsiębiorstwie.

Ponadto odbywały się wykłady WSZECHNICZY.

Równoległy przebieg obrad we wszystkich sekcjach był jednym z istotnych potknięć organizacyjnych Infogryfu. Okazało się bowiem, co można było przewidzieć, że Mikroskala była praktycznie bezkonkurencyjną sekcją. Oznaczało to minimalną frekwencję na pozostałych dwóch salach obrad, gdzie były wygłaszane nie mniej wartościowe referaty, jednak nie tak atrakcyjne (modne jak mikrokomputery).

Tematem jednego z wyróżniających się wykładów były systemy operacyjne mikrokomputerów. Autor — Jan MADEY z UW — w sposób niezmiernie interesujący zaprezentował podstawowe systemy operacyjne CP/M i jego pochodne, MS-DOS i PC-DOS ze szczególnym podkreśleniem architektury zestawu mikro, oraz czynności użytkownika. UNIX został określony przez autora wykładu jako fenomen systemów operacyjnych, podobnie jak Pascal wśród języków programowania.

Bardzo ciekawy był także wykład pani profesor Chislaine DUFOURD z Uniwersyte Louis Pasteur w Strasburgu — „Informatyka dla wszystkich: doświadczenia francuskie”. Wykład dotyczył problemów edukacji w kontekście rozszerzania się sfery zastosowań informatyki z punktu widzenia sprzętu, opro-



Dzięki uprzejmości firmy DSE można ocenić drobny przykład pomocy mikro — wykonanie prezentowanych rycin trwało kilkanaście minut.

gramowania, a także przygotowania kadr nauczycielskich. W trakcie wykładu autorka wprowadziła oryginalny termin — nanokomputery.

Cenne uzupełnienie konferencji stanowią cztery tomy referatów i wykładów (brak nazw firm, autorów, adresów do korespondencji jest istotnym mankamentem).

Podczas INFOGRYFU'86 odbywała się wystawa sprzętu komputerowego. Przez kilka godzin można było oglądać ELWRO 800 — mikrokomputer firmy ELWRO, którego kilkadziesiąt egzemplarzy prawdopodobnie będzie wykonane w najbliższej przyszłości... Natomiast „od ręki” można było kupić prezentowany zestaw, kompatybilny z IBM PC/AT, oferowany przez firmę MIKROKOMPUTERY za 14,9 mln zł. Podobną ofertę złożył COMPUTEX (CS-88PC i CS-286PC, odpowiednio XT i AT). Firma DSE ze Sztokholmu proponowała komputer DSE TURBO 1

(kompat. z XT) i DSE TURBO 2 (kompat. z AT) odpowiednio za 2300 i 3300 dolarów. Reprezentowane były także firmy HIROSS i SELIN proponujące kompletne wyposażenie sal komputerowych (w tym poszukiwana podwójna podłoga!), klimatory, przetwornice. Stoisko swoje miało także COMPUTER STUDIO KAJKOWSCY.

Organizatorzy konferencji twierdzili, iż Infogryf jest taki jak jego uczestnicy. W ocenie wielu osób był chyba trochę zbyt teoretyczny, za co zapewne nie można winić samych referentów. Faktem jest natomiast, że korzyści z osobistych kontaktów przedstawicieli różnych firm, giełda sprzętu i oprogramowania stanowiły wystarczające uzupełnienie INFOGRYFU'86 — był nadal miarodajnym źródłem danych i opinii o stanie informatyki w kraju. Do zobaczenia za dwa lata.

UCZESTNIK

UNIX

Są dwie historie UNIXa. Jedna mówi, iż UNIX jest znany na całym świecie, druga, że nie. Każda z nich ma swych zwolenników, których liczba co roku się zmienia.

Obecnie zakres potencjalnych maszyn tego systemu operacyjnego, takich jak IBM PC AT i Big Blue oferujących UNIX jako swój podstawowy system odnowił związane z nim spekulacje. Czy tylko dlatego, że to co IBM myśli dzisiaj, reszta świata próbuje powiełać jutro?

Jak wygląda UNIX z bliska? Co to jest? Skąd pochodzi? Dlaczego ludzie nad nim pracują? Jakie są jego perspektywy? Czy stanie się standardem systemu operacyjnego następnej generacji mikrokomputerów?

Dla wielu, UNIX jest sposobem życia, religią lub obiektem miłości lub nienawiści. Dla pozostałych jest pewnym systemem operacyjnym — to znaczy programem, który czyni, iż części składowe (hardware) komputera pracują. Bez systemu operacyjnego komputer może służyć tylko jako kosztowny mebel.

System operacyjny jest oprogramowaniem, które mówi jak wprowadzić i wyprowadzić informacje, jak sterować pracą dysków, zbiorów i programów i jak dzielić moc obliczeniową komputera na wielu użytkowników oraz urządzenia zewnętrzne.

W systemach mikroprocesorowych, tylko mała część systemu operacyjnego jest zależna od specyfiki komputera;

jego większa część jest niezależna od sprzętowej konfiguracji maszyny. Oznacza to, że system operacyjny może być łatwo dostosowany do różnych maszyn, których jedyną wspólną charakterystyką jest to, że wykorzystują one ten sam mikroprocesor (CPU). W każdym przypadku tylko specyficzne dla danej maszyny kody powinny być uwzględnione, reszta może pracować bez zmian.

Z innego punktu widzenia oznacza to, że jeśli programiści piszą programy przeznaczone do określonego typu systemu operacyjnego, mają oni duży potencjalny rynek dla swojej produkcji w istocie tworzą go wszystkie maszyny, w których pracuje ten szczególny typ systemu operacyjnego.

Praktycznie mikrokomputery miały dwa znaczące systemy operacyjne. Pierwszym był CP/M (Control Program for Micros) napisany przez Gary Kidalla dla procesorów Z80 i 8080. CP/M stał

dokończenie na str. 5

29 listopada — Dzień Podchorążego



Wszystkim studentom w żołnierskich mundurach z okazji ich święta składamy najlepsze życzenia.

REDAKCJA

Foto: S. Iwan



dokończenie ze str. 3

się nie tylko podstawą firmy **Kidalla** — **Digital Research**, ale standardem dla komputerów 8-bitowych. Rezultatem tego była znaczna liczba programów pracujących pod CP/M.

Kiedy IBM zaprojektował swój PC Digital Research opracować miał system operacyjny dla procesora 8088. Stworzono wiele różnych wersji, ale ostatecznie oprogramowanie wykonała mała firma — **MICROSOFT**. Efektem był **MS DOS** — rozbudowane wersje CP/M.

CP/M i MS DOS są przeznaczone dla indywidualnych użytkowników, niemożliwa jest praca w tak zwanym wielodostępnie. Stan ten był zadowalający do chwili, gdy mikrokomputery miały ograniczony maksymalny adres pamięci równy 640 KB i szybkość przetwarzania nie przekraczającą 4,77 MHz.

Powstanie procesorów o większej mocy obliczeniowej, znacznie przekraczającej potrzeby jednego użytkownika takich jak: INTEL 80286, MOTOROLA 68020 i ZILOG Z8000 oraz olbrzymi spadek cen pamięci, spowodowało pilną potrzebę posiadania systemu, który może obsłużyć wielu użytkowników i realizować wiele zadań jednocześnie.

Wykorzystanie do tego celu dotychczasowych systemów napotykało na wiele trudności. Dwie z nich są szczególnie ważne. Pierwsza — większość dużych systemów operacyjnych jest specyficzna dla danej maszyny. Na przykład IBM OS — 360 pracuje tylko na komputerach z rodziny IBM 360. VMS Digital'a pracuje tylko na procesorach rodziny VMS. Druga trudność także wynika ze specyfiki maszyn, a w zasadzie właścicieli oprogramowania, którzy zazwyczaj niechętnie umożliwiają dopasowanie systemu do podobnej maszyny, ale produkowanej w innej firmie.

UNIX natomiast jest wyjątkiem wśród większości wspomnianych trudności. Po pierwsze jest to system wielodostępny, wieloprogramowy bazujący na olbrzymich doświadczeniach zdobytych na całym świecie. Praktycznie każdy uniwersytet i komputerowy zakład naukowy na Zachodzie ma co najmniej jedną maszynę pracującą pod UNIXem. UNIX jest jednocześnie „stółowym” systemem, **ponieważ w odróżnieniu od innych systemów nie jest napisany w specyficznym dla danej maszyny assemblerze, ale zazwyczaj w wysokopoziomym języku C.**

UNIX z powodzeniem został wykorzystany w dominujących na rynku rodzinach procesorów 16/32-bitowych włączając 68000, 80x86 i Z8000. Nie-

zadowoleni użytkownicy MS DOS chętnie i efektywnie wykorzystują UNIXa.

Właścicielem UNIXa jest korporacja **AT&T**, która dotychczas wcale nie była znana.

W zasadzie UNIX jest nie tylko systemem operacyjnym. Lepszym jego określeniem jest system operacyjny plus komputerowe otoczenie. Zatem, dodatkowo do zabezpieczenia podstawowych zadań pozwalających na pracę w wielodostępnie użytkowników wykorzystujących różne programy UNIX oferuje obsługę zaskakującej liczby urządzeń i możliwości — języki, kompilatory, detektory błędów, dodatki dokumentacyjne, edytory i możliwości przetwarzania tekstów, pocztę elektroniczną, nadzór zbiorów, możliwości dokumentacji itp.

Inne systemy operacyjne posiadają podobne możliwości, ale wyłącznie jako pojedyncze dodatki — jedynie UNIX oferuje je jako standard, integralną część pakietu.

Część własności UNIXa ma na celu ułatwienie pracy programistom. Wiele wersji tego systemu posiada pakiet, który zawiera SOURCE CODE CONTROL SYSTEM (SCCS). Jest to pakiet programów, który automatycznie śledzi i dokumentuje każdą nową lub poprawioną wersję programu, zaznaczając datę, nazwisko autora itp.

Znaczna część pomysłów, rozwiązań może być wykorzystana w innych systemach, a sam UNIX dzięki oryginalnym pomysłom stał się dostępny także nie-programistom.

Można również zauważyć pewien wpływ tego nowego standardu na inne systemy? Na przykład hierarchiczna struktura zbioru w MS DOS v2.0 jest niemal wierną kopią konstrukcji z UNIXa.

Jeśli chcesz zrozumieć UNIXa to wyobraź sobie go jako owoc. W środku, gdzie zazwyczaj bywa pestka, jest jądro UNIXa — podstawowe kody, które pozwalają na pracę systemu. Na zewnątrz pestki jest muszla — interfejs między jądrem, a Tobą.

Jedną z przyczyn, z powodu której ludzie się ekscytują UNIXem jest siła i elastyczność wspomnianej muszli. Ponadto UNIX oferuje ponad 120 samodzielnych komend (instrukcji). Część z nich jest bardzo podobna do innych prostszych systemów. Na przykład „cp” w UNIXie jest podobne do „copy” w MS DOS. Użytkownicy mogą decydować o priorytetach realizowanych przez komputer zadań.

Już w ubiegłym roku sprzedano 100 000 UNIXów, ale MS DOS aż trzy miliony. Jednak proporcję tę należy szacować mając na uwadze, że każdy

UNIX dotyczy około 10 użytkowników (wielodostęp).

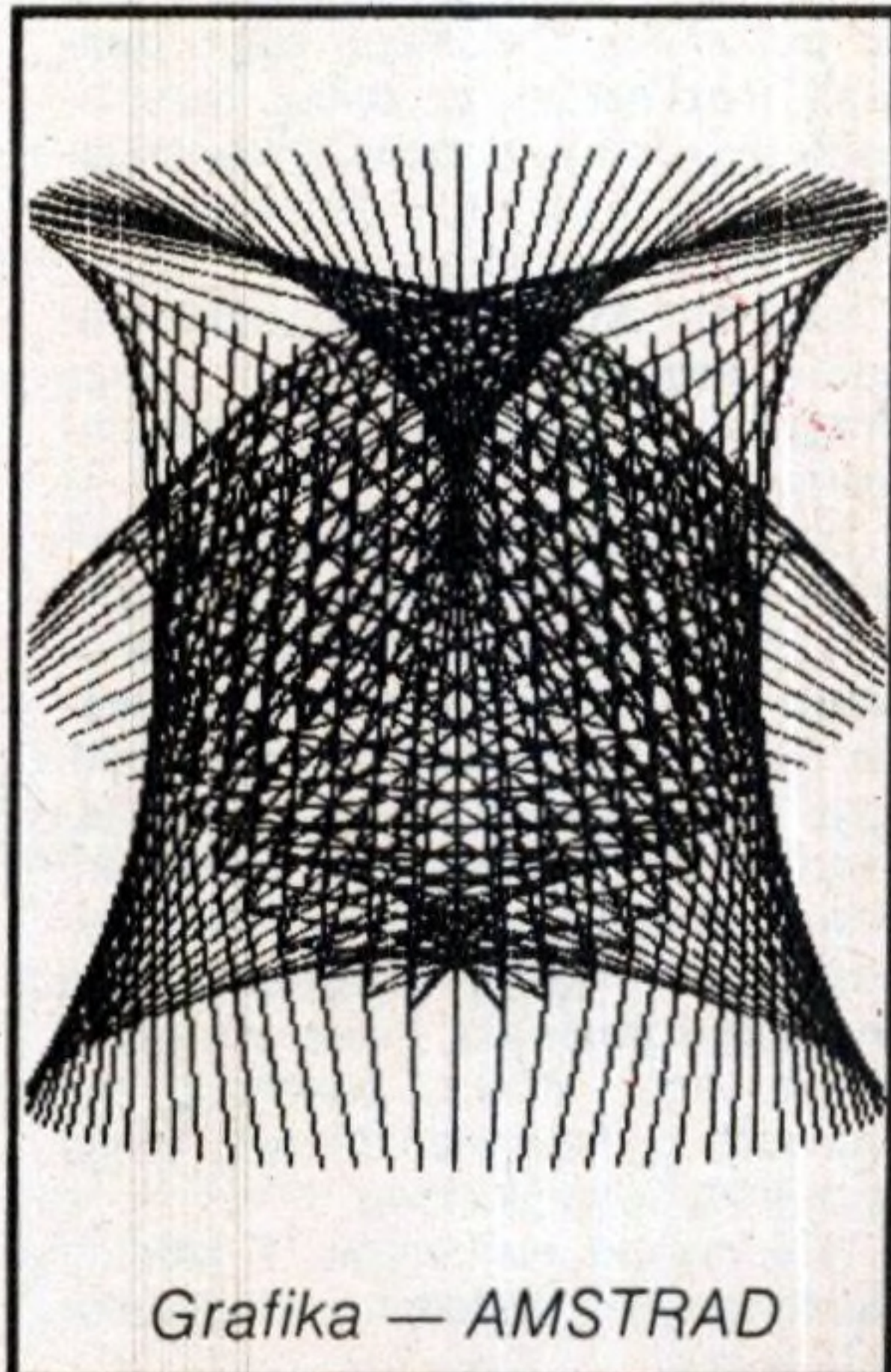
Istnieje wiele przyczyn stosunkowo niewielkiej powszechności UNIXa. Do niedawna jedną z nich było małe zainteresowanie producenta, firmy AT&T w dystrybucji tego systemu. Inną przyczyną jest obecnie zbyt ubogi sprzęt — aby UNIX poprawnie pracował potrzebne jest — 1MB RAM i 20 MB twardego dysku. Po trzecie dotychczas rynek był na tyle nasycony 16-bitowymi komputerami dla jednego użytkownika, że zainteresowanie systemem wielodostępu było nikłe.

Stan ten zmienił się radykalnie. Firma AT&T rozszerzyła swoje komercyjne zainteresowania rynkiem komputerowym. Ponadto IBM wydaje się być poważnie zainteresowany UNIXem. Dowodzą tego starania zlikwidowania technicznych ograniczeń, tym bardziej że użytkownicy IBM PC dostrzegają pewne ograniczenia, które narzuca 8088/86 MS DOS. Maksymalna dostępna pod tym systemem pamięć wynosi zaledwie 640 KB, a np. pakiet typu SYMPHONY, tylko dla celów systemowych, wymaga 320 KB. Oznacza to po prostu głód pamięci RAM do celów użytkowych.

Procesor 80286, który pracuje w IBM AT może operować na pamięci ponad 4 MB, ale nie pod systemem MS DOS. Dla pojedynczych użytkowników całą taką dużą pamięć nadzoruje DOS 286.

Zalety UNIXa dostrzegło wiele firm deklarując ścisłą współpracę w rozwoju tego systemu np.: ICL, NIXDORF, PHILIPS, SIEMENS, BULL, OLIVETTI i oczywiście Japończycy.

Włodzimierz GOGOLEK



Grafika — AMSTRAD

ARCHITEKTURA INTEGRACJI

W 1971 r. firma INTEL skonstruowała pierwszy na świecie mikroprocesor, oznaczony symbolem 4004, o architekturze czterobitowej a wykonany w technologii PMOS. Dało to początek rywalizacji kilkudziesięciu firm i tysięcy inżynierów o najkorzystniejsze rozwiązania architektury mikroprocesorów i technologii ich wytwarzania. Wśród firm obecnie dominujących na świecie należy wymienić: INTEL, ZILOG, MOTOROLA, MOS TECHNOLOGY, RCA, FAIRCHILD, NATIONAL SEMICONDUCTOR, TEXAS INSTRUMENTS, TOSHIBA.

Mikroprocesor poza przewrotem, którego dokonał w przemyśle komputerowym, na stałe pojawił się w automatyce przemysłowej, w telekomunikacji, w medycynie, w technice pomiarowej, w sprzęcie powszechnego użytku i w technice wojskowej.

Mikroprocesorem nazywana jest jednostka centralna (CPU — Central Processor Unit) systemu mikrokomputerowego, wykonana w postaci jednego lub kilku układów wielkiej skali integracji. Mikroprocesor przeznaczony jest do wykonania dowolnego ciągu **operacji (rozkażów)** ze zbioru określonego jako **lista rozkażów**, a zawierającego zestaw operacji podstawowych mikroprocesora danego typu.

Ciąg operacji oraz dane, które stanowią **argumenty operacji**, tworzą **program** dla systemu mikrokomputerowego.

W tym miejscu autor chce się przyznać, iż nie może podać precyzyjnych definicji **systemu mikrokomputerowego** i **systemu mikroprocesorowego**. Fizycznie systemy te nie różnią się, gdyż składają się z tych samych elementów. Można przyjąć, że systemem mikrokomputerowym jest system wyposażony w **język programowania wyższego poziomu**, daleki od języka dwójkowego, którym posługuje się procesor a możliwie zbliżony swoją symboliką do języka ludzi.

Wróćmy do elementów, z których zbudowany jest system mikrokompute-

rowy (system mikroprocesorowy). Zawiera on następujące elementy, przedstawione na rys. 1.:

— mikroprocesor, przeznaczony do przetwarzania informacji, inaczej wykonania operacji na danych i do sterowania pracą pozostałych elementów systemu;

— pamięci: stałą ROM (Read — Only — Memory) i zapisywalną RAM (Random — Access — Memory);

— układy wejścia — wyjścia.

Wymienione elementy systemu połączone są ze sobą za pomocą **magistral**, w tym: **magistrali adresowej** (Address Bus), **magistrali danych** (Data Bus) i **magistrali sterującej** (Control Bus), w prostszych rozwiązaniach określanej jako **linie sterujące**.

W **pamięci stałej ROM** przechowywany jest program oraz dane nie zmieniające się w czasie pracy systemu, znajdujące się na trwałe nawet po wyłączeniu zasilania. Pamięć ta zapisywana jest przez producenta, znajdują się w niej **interpretery, system operacyjny** itp. Przykładowo w komputerze AMSTRAD CPC 464 znajduje się pamięć ROM o pojemności 32 Kilobajty (bajt — słowo 8-bitowe, inaczej długość słowa mikroprocesora 8-bitowego, przy czym 1 KB = 1000 bajtów). W pamięci tej 16 KB to system operacyjny i 16 KB to interpreter języka BASIC.

W **pamięci zapisywalnej RAM**, zwanej **pamięcią operacyjną**, zapisywane (i odczytywane) są dane powstałe w czasie pracy systemu oraz programy użytkownika.

Układy wejścia — wyjścia umożliwiają komunikację systemu z otoczeniem czyli wszelkimi urządzeniami zewnętrznymi np. monitorem, klawiaturą, pamięciami masowymi, urządzeniami pomiarowymi itp.

Wymienione zasadnicze elementy systemu wykonane są jako układy LSI technologią MOS, umożliwiającą osiągnięcie dużej **integracji (scalenia)** tranzystorów. Musimy pamiętać, że tranzystory to podstawowy element używany do konstrukcji mikroprocesorów. Tranzystory te noszą nazwę **unipolarnych** typu MOS. Technologia MOS zapewnia zastępowanie innych elementów konstrukcyjnych mikroprocesorów np. rezystorów również tranzystorami. Pozwala ona na osiągnięcie integracji, jak we wspomnianym w poprzednim numerze „IKS-a” koprocesorze arytmetycznym MC 68881.

W systemach mikrokomputerowych do budowy połączeń sprzęgających ich elementy, w dalszym ciągu wyko-

rzystywane są **bipolarne** układy MSI i SSI. Technologiami bipolarnymi wykonywane są również **mikroprocesory segmentowe**.

Wróćmy na chwilę do terminologii. Określenie mikrokomputer pojawiło się przez analogię do mikroprocesora. Obecnie „mikro-” nie świadczy o małych możliwościach, należy to chyba rozumieć z punktu widzenia technologii produkcji. Przykładowo porównajmy komputer IBM 650 z dwoma mikroprocesorami, z Z-80 i MC 68000. Komputer IBM 650 posiadał pamięć bębnową o pojemności 2000 słów, miał listę 60 rozkażów, czas wykonywania rozkażu wynosił 20 milisekund [ms]. W 8-bitowym Z-80 czas wykonania rozkażu najkrótszego/najdłuższego wynosi 1 μ s (5,75 μ s czyli 0,001 ms) 0,00575 ms, mikroprocesor adresuje 64 KB przestrzeni pamięci i posiada listę zawierającą 158 rozkażów. 16-bitowy MC 68000 posiada listę 61 rozkażów, czas wykonania rozkażu 0,5 μ s czyli 0,0005 ms i adresuje przestrzeń adresową 16 megabajtów [MB]!!

Zanim przejdziemy do ogólnej charakterystyki mikroprocesorów jeszcze kilka słów. Obecnie budowane są systemy mikrokomputerowe złożone z wielu mikroprocesorów, o różnym przeznaczeniu, są to tzw. **systemy wielomikroprocesorowe**, charakteryzujące się dużą efektywnością i niezawodnością pracy. Stosuje się w tych systemach m.in. **koprocesory arytmetyczne**, przeznaczone do wykonywania operacji arytmetycznych (których nie wykonuje już **mikroprocesor centralny**) oraz stosuje się **mikroprocesory wejścia — wyjścia**, przeznaczone do obsługi szybkich transmisji między urządzeniami zewnętrznymi a pamięcią operacyjną. Przykładowo 16-bitowy komputer personalny IBM PCAT zawiera, obok mikroprocesora 80286, koprocesor arytmetyczny 80287.

Mikroprocesory można zgrupować w cztery **generacje**.

Mikroprocesorami **pierwszej generacji** są dwie konstrukcje firmy INTEL, czterobitowy **4004** i ośmiobitowy **8008**, charakteryzujące się ubogimi listami rozkażów i małymi szybkościami działania.

Czterobitowy mikroprocesor INTEL 4004, zaprojektowany i wykonany w 1971 r., posiada listę 46 rozkażów, w tym 16 rozkażów procesora, 14 rozkażów akumulatora i 16 rozkażów wejścia—wyjścia. W skład systemu ozna-

czonę MCS-4 wchodzi, oprócz mikroprocesora, układ pamięci ROM 4001 o pojemności 256×4 bity, układ pamięci RAM 4002 o pojemności 80×4 bity, układ 4003 do współpracy z wejściem — wyjściem do — z systemem. Cały cykl rozkazowy mikroprocesora trwa około $10,8 \mu s$.

INTEL 4004, wykonany w technologii PMOS, może adresować przestrzeń 4 K czyli 4096 słów 4-bitowych. Rozwinięciem rodziny MCS-4 jest MCS-40. INTEL 4040, adresuje 8192 słowa 4-bitowe. Jego architektura jest rozwinięta w stosunku do poprzednika, umożliwia ona obsługę **przerwań jednopoziomowych**.

Ośmiobitowy mikroprocesor INTEL 8008, posiadający listę 45 rozkazów, stanowi rozwinięcie architektury mikroprocesora 4004. Posiada on możliwość działania na danych ośmiobitowych. Produkcję mikroprocesora rozpoczęto w 1972 r.

Cały cykl rozkazowy 8008 wynosi około $12,5 \mu s$. Mikroprocesor może adresować 16 KB czyli 16 384 słowa 8-bitowe. MCS-8 jest systemem równoległym, wykonanym w technologii N MOS, komunikującym się z pamięcią i układami wejścia — wyjścia przez magistrale 8-bitowe. Mikroprocesor 8008 posiada listę 45 rozkazów, w tym rozkazy 8-, 16- i 24-bitowe.

INTEL 8008 zapoczątkował rozwój najliczniejszej grupy mikroprocesorów ośmiobitowych. W ciągu następnych kilku lat powstało kilkadziesiąt różnych konstrukcji mikroprocesorów ośmiobitowych, wykonanych przede wszystkim technologią N MOS, tworzących **drugą generację**.

Najważniejszymi przedstawicielami drugiej generacji mikroprocesorów są

rodziny INTEL MCS-80 i MCS-85, mikroprocesor firmy MOTOROLA MC 6800 i rodzina mikroprocesorów firmy MOS TECHNOLOGY MCS 650X. Mikroprocesory te posiadają dużo większe szybkości działania, bogatsze listy rozkazowe i przestrzeń adresową 64 KB. Wykonano je technologią N MOS.

Największą popularność z wymienionych wyżej mikroprocesorów zyskał 8080. Wprowadzony na rynek w 1974 r. dominował przez kilka lat. Wzięło się to z monopolu firmy INTEL na produkcję pamięci (stałych) programowalnych PROM i reprogramowalnych EPROM, wśród nich standardem światowym stały się EPROM 1702 A o pojemności 256×8 bitów i EPROM 2708 o pojemności 1024×8 bitów. Charakterystyką mikroprocesorów 8008 i 8085 zajmujemy się w kolejnych numerach „IKS-a”, a więc na razie tylko tyle.

Mikroprocesor MC 6800 znacznie przewyższa mikroprocesor 8080 zarówno łatwością organizacji systemu jak również: mocą obliczeniową, szybkością obsługi przerwań, elastycznością adresowania i wymaga on tylko jednego napięcia zasilającego +5 V. W skład rodziny MC 6800 wchodzi następujące układy towarzyszące: układy interfejsu MC 6820, MC 6850, MC 6852, MC 6843, MC 6854, MC 68488, układ bezpośredniego dostępu do pamięci (DMA—Direct Memory Access) MC 6844, programowalny zegar czasu rzeczywistego MC 6840, modulator cyfrowy MC 6862, modem MC 6860 i inne.

Podstawowe cechy charakterystyczne mikroprocesora 6800 to: 16-bitowa magistrala adresowa, 8-bitowa magistrala danych, lista złożona z 72 rozkazów, czas wykonania rozkazu najkrótszego/najdłuższego $1/2,5 \mu s$.

Produkcję układów rodziny mikroprocesorowej MOTOROLA MC 6800

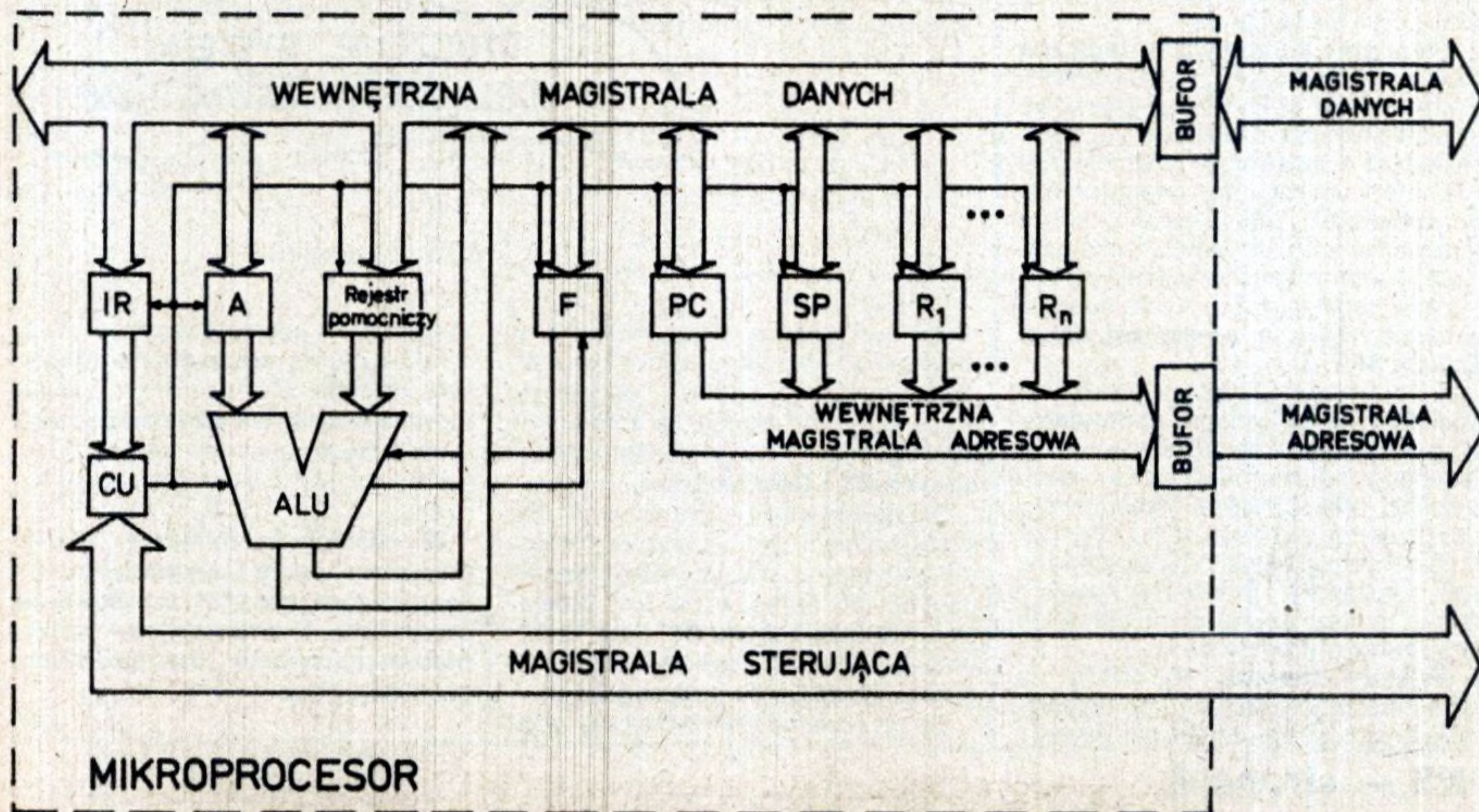
uruchomiono w Bułgarii, oznaczając rodzinę symbolem CM 600. Z uwagi na niewielkie (jak dotąd) zainteresowanie tym mikroprocesorem w naszym kraju, nie będziemy się nim zajmować. Mikroprocesor ten możemy spotkać w nowoczesnej aparaturze kontrolno-pomiarowej firm Hewlett-Packard czy Tektronix.

Udoskonalenie MC 6800 stanowią mikroprocesory rodziny MCS 650X, opracowane w firmie MOS TECHNOLOGY. Najważniejsze cechy tej rodziny to: duża liczba sposobów adresowania (13), elastyczność przerwań oraz załaskowe wykonywanie instrukcji — wykonanie jednej mikroinstrukcji i interpretacja następnej w czasie jednego cyklu zegarowego. Mikroprocesor MCS 6502, mimo tylko 56-elementowej listy rozkazów uważany jest za najlepszego przedstawiciela drugiej generacji.

Trzecią generację otwierają, w dalszym ciągu 8-bitowe, mikroprocesor ZILOG Z-80, wprowadzony na rynek w 1976 r. i ulepszona wersja MC 6800 oznaczona MC 6809, wprowadzona w 1977 roku. Mikroprocesor Z-80 stanowi daleko idące ulepszenie i rozwinięcie INTEL-a 8080, będzie również dokładnie omawiany w następnych numerach „IKS-a”. Zasadniczy trzon generacji tworzą mikroprocesory 16-bitowe. Najważniejszymi ich przedstawicielami są mikroprocesory: INTEL 8086 i INTEL 8088, MOTOROLA 68000 i ZILOG Z-8000. Na pograniczu **czwartej generacji** jest mikroprocesor INTEL 80286. Z uwagi na pewien ciąg rozwojowy zajmujemy się również 16-bitowymi mikroprocesorami firmy INTEL.

Czwarta generacja to mikroprocesory 32-bitowe, jeden firmy INTEL oznaczony 80386 i jeden firmy MOTOROLA

dokończenie na str. 8



MC 68020. Dzięki nim **piąta generacja** komputerów jest już coraz bliżej.

Po ogólnej charakterystyce mikroprocesorów zapoznajmy się z zasadą ich działania i architekturą.

W każdym z mikroprocesorów, zgodnie ze znaną nam koncepcją Von Neumanna, występują następujące podzespoły, przedstawione na rys. 2:

- **jednostka arytmetyczno-logiczna ALU** (Arithmetic and Logic Unit);
- **układ sterowania CU** (Control Unit);
- **akumulator A** (Accumulator);
- **zestaw rejestrów roboczych R_n** (General Purpose Registers);
- **rejestr rozkazów IR** (Instruction Register);
- **licznik rozkazów PC** (Program Counter);
- **wskaźnik stosu SP** (Stack Pointer);
- **rejestr wskaźników SF** (Status Flag).

Cykl rozkazowy mikroprocesora rozpoczyna się od fazy pobrania rozkazu do **rejestru rozkazów IR**, z komórki pamięci adresowanej przez **licznik rozkazów PC**. Po pobraniu przez mikroprocesor słowa z **pamięci programu** licznik zwiększa swoją zawartość o jeden tzn. adresuje następną komórkę pamięci lub w przypadku rozkazu skoku skokowo zmienia swoją zawartość.

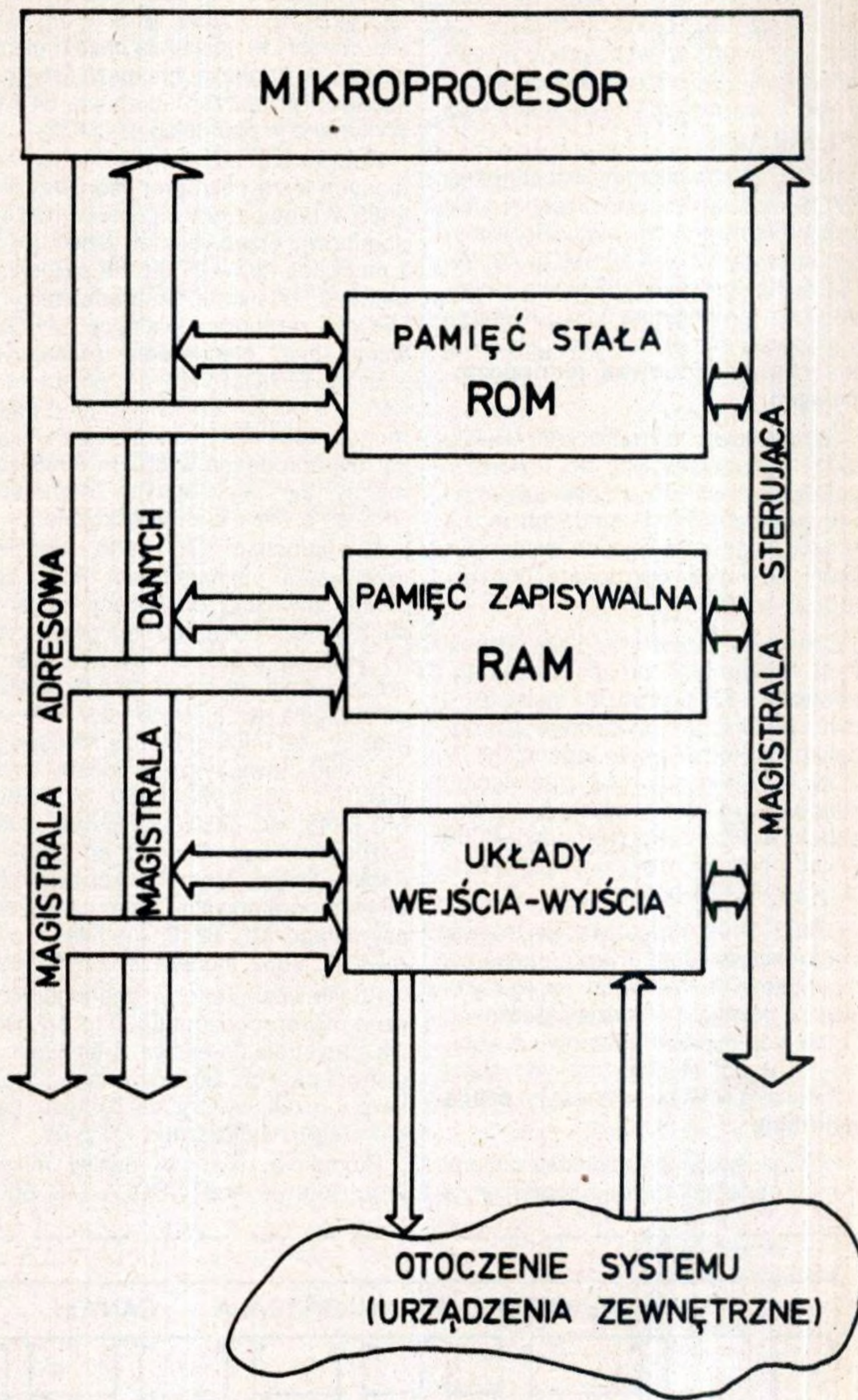
Po pobraniu rozkazu następuje faza wykonania rozkazu. **Układ sterowania CU** generuje sekwencję sygnałów sterujących, zapewniających wykonanie rozkazu.

Każdy cykl rozkazowy składa się z kilku **cykli maszynowych**. Cykl maszynowy składa się z kilku **taktów zegarowych**. Faza pobrania rozkazu posiada tyle cykli maszynowych ile słów zawiera rozkaz, natomiast faza wykonania rozkazu może zawierać różną liczbę cykli maszynowych.

Jednostka arytmetyczno-logiczna ALU, poysterowaniu przez układ sterowania CU, wykonuje podstawowe działania arytmetyczne i logiczne. Są to zazwyczaj dodawanie arytmetyczne (binarne i dziesiętne), odejmowanie, porównywanie, przesuwanie słowa o jeden bit w lewo lub prawo, suma logiczna, iloczyn logiczny, różnica symetryczna, negacja słowa. Wyniki operacji zapamiętywane są w **rejestrze wskaźników SF**.

Z ALU funkcjonalnie związany jest specjalny rejestr roboczy **akumulator A**, przeznaczony do przechowywania jednego z argumentów (słowa) wykonywanej operacji dwuargumentowej i do przechowywania wyniku. Za pośrednictwem akumulatora dokonywane są najczęściej przesłania danych (słów) między ALU a pamięciami lub urządzeniami zewnętrznymi.

Rejestry robocze R_n służą do przechowywania danych i do określania stanu magistrali adresowej. Rejestry



mogą być łączone w pary tworząc, w przypadku mikroprocesorów 8-bitowych, rejestr 16-bitowy. Taki rejestr może służyć do określania adresu komórki pamięci, nosi on nazwę **licznika danych DC (Data Counter)**.

Do adresowania wydzielonego obszaru pamięci RAM, przeznaczonego do przechowywania (i odczytywania) danych lub adresów na tzw. **stosie**, służy **wskaźnik stosu SP**. Stos umożliwia zawieszenie wykonywania programu i przejście do **podprogramu**.

Do przesyłania danych między pod-

zespołami mikroprocesora służy dwukierunkowa **wewnętrzna magistrala danych**. Jednokierunkowa **wewnętrzna magistrala adresowa** pozwala na adresowanie zawartościami rejestrów pamięci lub urządzeń zewnętrznych.

W następnym numerze „IKS-a” omówimy logikę i arytmetykę mikroprocesora, sposoby adresowania, przerwania i rozpoczniemy szczegółowe omawianie mikroprocesora ośmiobitowego.

J.W.

Jesteśmy zdecydowanymi przeciwnikami takich gier, w których OSZUST, posługujący się ZNACZONYMI kartami, bez skrupułów OGRABIA naiwnego.

Ponieważ jednak kuszenie szczęścia zawsze korciło, zatem proponujemy Wam grę UCZCIWA, w której przeciwnikiem (niezainteresowanym materialnie) będzie Wasz komputer.

Prezentowana gra jest PROGRAMOWO NIERÓWNA — uprzywilejowany jest gracz: MOŻE on ZMIENIĆ jedną, dwie, a nawet pięć kart. Komputer (programowo) NIE WYMIENIA: „gra” tym, co wylosuje.

Zatem po nabraniu wprawy, gdy rozgromimy komputer, dajmy mu szansę: OGRANICZAJMY ZMIANY. Gdy z nich zrezygnujemy, gra będzie WYRÓWNANA.

Program wywołuje RUN „A.

Karta wymieniona jest oznakowana gwiazdeczką.

Życzymy dobrej zabawy!

Budowa programu

— „wydawaniem kart” zajmuje się generator liczb losowych. Losuje „honor” RND 6, potem „kolor” RND 4,

— wydane karty są rejestrowane w dwuwymiarowej pamięci. To UNIEMOŻLIWIA POWTÓRZENIA. Macierz ma wymiar 6 x 4, zatem operujemy 24 kartami. „Honory” pamięta jednokolumnowa macierz X\$ zaś „kolory” macierz Y\$.

Janusz MILLER

SHARP PC-1500

KOMPUTER...
...POKERZYSTA

```

100: "A": CLEAR : DIM
      T(5, 3): DIM X$(
      5): DIM Y$(3):
      DIM M$(5): DIM
      P$(5): WAIT : K=
      RND 4: W=RND 8
110: X$(0)="A": X$(1
      )="K": X$(2)="D
      ": X$(3)="W": X$
      (4)="10": X$(5)
      ="9"
120: Y$(0)="KIER": Y
      $(1)="KARO": Y$
      (2)="TREFL": Y$
      (3)="PIK"
130: PRINT "MOJE KA
      RTY"
140: FOR L=1 TO 5
150: GOSUB "R": W$=
      STR$ L+" "
160: M$(L)=Z$: PRINT
      W$: Z$
170: NEXT L
180: INPUT "ILE ZMI
      ENIC ? "; Z: Z=
      INT Z: Z=ABS Z:
      IF Z=0 GOTO 340
190: IF Z<0 OR Z>5
      THEN BEEP 10, 1
      00, 100: GOTO 18
      0
200: IF Z=1 THEN
      INPUT "ZMIEN W
      IERSZ "; A: GOTO
      290
210: IF Z=2 THEN
      INPUT "ZMIEN D
      WA WIERSZE ", A
      , B: GOTO 280
220: IF Z=3 THEN
      INPUT "ZMIEN T
      RZY WIERSZE ",
      A, B, C: GOTO 270
230: IF Z=4 THEN
      INPUT "ZMIEN C
      ZTERY WIERSZE
      ", A, B, C, D: GOTO
      260
240: A=1: B=2: C=3: D=
      4: E=5

```

```

250: GOSUB "R": M$(E
      )=Z$+"*"
260: GOSUB "R": M$(D
      )=Z$+"*"
270: GOSUB "R": M$(C
      )=Z$+"*"
280: GOSUB "R": M$(B
      )=Z$+"*"
290: GOSUB "R": M$(A
      )=Z$+"*"
300: CLS : PRINT "ZM
      IENIONE"
310: FOR L=1 TO 5
320: W$=STR$ L+" "
      : PRINT W$: M$(L
      )
330: NEXT L
340: PRINT " KOMPUT
      ER "
350: FOR L=1 TO 5
360: GOSUB "R": W$=
      STR$ L+" "
370: P$(L)=Z$: PRINT
      W$: Z$
380: NEXT L
390: INPUT "GRASZ D
      ALEJ? T=1 N=0
      "; D
400: IF D=1 GOTO 100
410: END
420: "R": W=RND 6: W=
      W-1
430: K=RND 4: K=K-1
440: IF T(W, K)=1
      GOTO 420
450: T(W, K)=1: Z$=X$
      (W)+" "+Y$(K):
      RETURN

```

PRZYKŁAD
MOJE KARTY
1. A KIER
2. A TREFL
3. 10 KARO
4. 9 KIER
5. 9 KARO
ZMIENIONE
1. A KIER
2. A TREFL
3. K KIER*
4. 9 KIER
5. 9 KARO
KOMPUTER
1. A KARO
2. 9 TREFL
3. K KARO
4. 10 KIER
5. K TREFL

Pełne możliwości programu przedstawia tzw. menu. Program „Książka telefoniczna” umożliwia wprowadzenie i zachowanie na dyskietce (w zbiorze „DANE”) dowolnej liczby abonentów. Prezentowany program może zachować informacje o 100 abonentach, ale chcąc zwiększyć tę liczbę do np. 1000 należy w wierszu o nr. 20 zmienić $n = 100$ na $n = 1000$. W trakcie korzystania z programu, w celu wyszukania konkretnego abonenta lub przejrzania całego zbioru abonentów należy najpierw odzyskać tę informację z dyskietki (czynność 5). Zobrazowanie informacji w tzw. oknach oraz sygnalizacja dźwiękowa wyraźnie poprawia komfort korzystania z programu.

UWAGA:

Chcąc wprowadzić do istniejącego zbioru na dyskietce nowych abonentów, należy najpierw odzyskać cały zbiór abonentów (czynność 5 z „menu”).

```

10 REM KSIAZKA TELEFONICZNA
20 n=100
30 DIM imie$(n)
40 DIM tel$(n)
50 GOSUB 1260
60 MODE 1
70 WINDOW #1,7,35,8,16
80 WINDOW #2,3,20,22,20
90 PRINT #1,"1.- wprowadzanie danych"
100 PRINT #1
110 PRINT #1,"2.- przeglądanie danych"
120 PRINT #1
130 PRINT #1,"3.- wyszukiwanie danych"
140 PRINT #1
150 PRINT #1,"4.- zachowywanie danych"
160 PRINT #1
170 PRINT #1,"5.- odzyskiwanie danych"
180 INK 1,24;PEN #2,0,0;PAPER #2,13
190 LOCATE #2,2,25
200 PRINT #2, "WYBIERZ CZYNNOSC"
210 m$=INKEY$
220 IF m$="" THEN 210
230 IF m$="1" THEN 340
240 IF m$="2" THEN 550
250 IF m$="3" THEN 830
260 IF m$="4" THEN 1020
270 IF m$="5" THEN 1100
280 GOTO 210
290 CLS #0
300 LOCATE 5,12
310 PRINT "POWROT DO MENU - [DOWOLNY KLAWISZ]"
320 IF INKEY$="" THEN 320
330 GOTO 60

```

```

340 CLS:INK 1,26;k=0
350 FOR x=1 TO n
360 IF LEN(imie$(x))>0 THEN 540
370 a=1:CLS #0
380 LOCATE 3,25
390 PRINT "KONIEC WPROWADZANIA - [RETURN]"
400 WINDOW #7,3,32,20,20
410 PAPER #7,13;PEN #7,0
420 PRINT #7," WPROWADZONO ";k;" ABONENTOW "
430 LOCATE 1,2
440 PRINT "imie i nazwisko"
450 INPUT imie$(x)
460 IF imie$(x)="" THEN 290
470 PRINT "telefon"
480 INPUT tel$(x)
490 k=k+1
500 ENV 1,#9,2000;ENT -1,6,3,1
510 SOUND 1,200,0,1,1
520 NEXT
530 GOTO 290
540 LOCATE 12,14:PRINT "prosze czekac":GOTO 520
550 CLS
560 WINDOW #3,3,29,4,22
570 WINDOW #4,33,39,4,22
580 PAPER #3,13;PAPER #4,13;PEN #3,0;PEN #4,0;CLS #3;CLS #4
590 PRINT #0," IMIE I NAZWISKO";SPC(13);"TELEFON"
600 IF imie$(1)="" THEN 780
610 z=1
620 IF imie$(z)="" THEN 750
630 FOR x=0 TO 18
640 PRINT #3,imie$(z+x)
650 SOUND 1,200#RND,12
660 PRINT #4,tel$(z+x)
670 SOUND 1,400#RND,12
680 IF imie$(z+x)="" THEN 750
690 NEXT x
700 LOCATE 1,25
710 PRINT #0,"NACISNIJ DOWOLNY KLAWISZ"
720 IF INKEY$="" THEN 720
730 CLS #3;CLS #4
740 z=z+19:GOTO 620
750 LOCATE 1,25
760 PRINT "KONIEC PRZEGLADANIA - [DOWOLNY KLAWISZ]"
770 GOTO 320
780 LOCATE #3,6,10
790 PRINT #3," BRAK ABONENTOW"
800 FOR a=200 TO 250 STEP 10
810 SOUND 1,a,12:NEXT
820 GOTO 750
830 CLS
840 WINDOW #5,5,30,4,4
850 PAPER #5,13;PEN #5,0;CLS #5
860 PRINT #0,"PODAJ IMIE I NAZWISKO"

```

PROGRAM

35

```

870 INPUT #5,a$
880 FOR x=1 TO n
890 IF INSTR(imie$(x),a$)=0 THEN 950
900 LOCATE 7,6
910 PRINT imie$(x);SPC(5);tel$(x)
920 ENV 3,=9,9000
930 SOUND 1,300,0,13,3
940 IF LEN(imie$(x))>0 THEN 300
950 NEXT
960 WINDOW #7,5,38,6,6
970 PAPER #7,13;PEN #7,0
980 PRINT #7," BRAK ABONENTA W ZBIORZE "
990 ENV 3,=9,9000
1000 SOUND 1,100,0,13,3
1010 GOTO 300
1020 OPENOUT "dane"
1030 FOR x=1 TO n
1040 WRITE #9,imie$(x),tel$(x)
1050 NEXT
1060 CLOSEOUT
1070 CLS
1080 PRINT "DANE O ABONENTACH - ZACHOWANO"
1090 GOTO 300
1100 OPENIN "dane"
1110 FOR x=1 TO n
1120 INPUT #9,imie$(x),tel$(x)
1130 NEXT
1140 CLOSEIN
1150 CLS
1160 PRINT "DANE ODZYSKANO"
1170 k=0
1180 FOR x=1 TO n
1190 IF imie$(x)="" THEN 1220
1200 k=k+1
1210 NEXT
1220 j=n-k
1230 LOCATE 1,4
1240 PRINT "MOZESZ JESZCZE ZACHOWAC "j;" ABONENTOW"
1250 GOTO 300
1260 t1$="T E L E F O N Y"
1270 T2$=CHR$(164)+CHR$(32)+CHR$(32)+" CJS "
1280 MODE 0
1290 ENV 1,=9,2000;ENT -1,6,3,1
1300 ENV 2,127,0,0,127,0,0,127,0,0,127,0,0;ENV 3,=9,9000
1310 FOR c=1 TO LEN(t1$)
1320 LOCATE 2+c,10
1330 FOR n2=LEN(t1$) TO c STEP -1
1340 PRINT MID$(t1$,n2,1)
1350 LOCATE 2+c,10
1360 SOUND 135,20#n2,5,12,2,1
1370 NEXT n2,c
1380 SOUND 135,100,0,13,3,1,20
1390 LOCATE 5,15;PRINT t2$
1400 FOR c=1 TO 2000;NEXT
1410 CLS
1420 RETURN

```

Program KONSTRUKTOR LICZB jest grą edukacyjną, przeznaczoną przede wszystkim dla dzieci w wieku szkolnym. Może ona zainteresować również inne osoby. Celem gry jest rozwijanie umiejętności posługiwania się podstawowymi działaniami arytmetycznymi i szacowania wyników tych działań. Program KONSTRUKTOR LICZB opracowany został na mikrokomputer **Commodore 64, Plus/4 i Commodore 16**. W wypadku mikrokomputerów Commodore 16 i Plus/4 zamiast wiersza 10 wprowadzić należy wiersz:

```

10 COLOR0,1,1:COLOR4,1,1:Q=205:
K=202:B=239:KL=2648:PT=3670:A*=
CHR$(159)

```

W grze może brać udział od jednej do trzech osób. Każdy z graczy jest konstruktorem liczb, dysponującym narzędziami i materiałami. Narzędziami są cztery działania arytmetyczne: +, -, *, /. Materiałami są cyfry generowane przez mikrokomputer. Liczba cyfr jest zależna od tego, jakimi konstruktorami są gracze. Zmienia się od trzech cyfr (konstruktor zaawansowany) do pięciu cyfr (konstruktor początkujący). Dla każdego z graczy określany jest projekt liczby z zakresu od 100 o 999. Zadaniem gracza jest skonstruowanie za pomocą dostępnych narzędzi i materiałów własnej liczby, jak najbliższej zadanemu projektowi, w jak najkrótszym czasie. Liczba operacji, która w końcowym efekcie może być skonstruowana jest ograniczona do siedmiu. Czas tworzenia liczby zmienia się od jednej do trzech minut, zależnie od poziomu reprezentowanego przez konstruktorów.

Gra trwa pięć rund. W każdej rundzie gracze mają do wykończenia po jednym projekcie. Liczba uzyskanych punktów jest wypadkową czasu wykonywania konstrukcji liczby oraz dokładności, z jaką nasza liczba odpowiada projektowi.

Grę rozpoczynamy od podania liczby graczy oraz poziomu reprezentowanego przez konstruktorów (początkujący, przeciętny, zaawansowany). Określony gracz (numer jest wyświetlany na ekranie) po pojawieniu się komunikatu GOTÓW! przyciśnięciem dowolnego klawisza powoduje rozpoczęcie gry. Pojawiają się wówczas na ekranie: aktualnie dostępne materiały, projekt liczby a zegar zaczyna odliczać czas. Wymieniony gracz zaczyna konstrukcję liczby. Konstruowana liczba nie może wyjść poza zakres 0 — 9999. Używane materiały i narzędzia są w tej samej kolejności wyświetlane na ekranie, tworząc ciąg operacji arytmetycznych. Wyświetlany jest wynik bieżącego ciągu operacji arytmetycznych. Eliminację dowolnego elementu ciągu (materiału lub narzędzia) uzyskujemy po ustawieniu kursora na tym elemencie i przyciśnięciu klawisza DELETE. Przesuwanie kursora wewnątrz ciągu wykonujemy klawiszami → i ←. Zakończenie gry następuje po upływie określonego czasu lub po przyciśnięciu przez gracza klawisza RETURN, gdy konstrukcję liczby chce zakończyć przed czasem. Następuje wówczas podliczenie punktów uzyskanych przez gracza i do gry zapraszany jest kolejny gracz.

Uwaga: w tekście programu zamiast znaku \$ użyty jest znak :O'.

dokończenie na str. 12

```

0 REM ***KONSTRUKTOR LICZB***
10 POKE53280,0:POKE53281,0:Q=214:K=
211:B=198:KL=55896:PT=1622:A=CHR*(
150)
20 B=CHR*(156):C=CHR*(5):D=CHR*(
98):E=CHR*(99):F=CHR*(173)
30 G=CHR*(174):H=CHR*(176):I=CHR
*(189):J=CHR*(147):K=CHR*(32)
40 L=CHR*(157):S="":G=1:RN=1
50 PRINTJ*SPC(11)A*H*;;FORI=1TO17:P
RINTE*;;NEXT:PRINTG*
60 PRINTSPC(11)D*C*"KONSTRUKTOR LIC
ZB"A*D*
70 PRINTSPC(11)F*;;FORI=1TO17:PRINT
E*;;NEXT:PRINTI*
80 PRINT:PRINT:PRINTB*"NARZEDZIA:"C
*" + - * /"SPC(12)A*"CZAS"
90 PRINT:PRINTB*"MATERIALY:"C*"0
0 0 0"SPC(9)"0:00"
100 PRINT:PRINTSPC(9)B*H*E*E*E*G*
110 PRINT"PROJEKT:"D*C*"000"B*D*
120 PRINTSPC(9)F*E*E*E*I*
130 PRINT:PRINT:PRINTSPC(32)A*H*E*E
*E*E*G*
140 PRINTC*"0 ";:FORI=1TO7:PRINT"+
0 ";:NEXT:PRINTCHR*(146)"="A*D*C*"
0000"A*D*
150 FORI=1TO15:PRINTCHR*(163)K*;;NE
XT:PRINTSPC(2)F*E*E*E*I*
160 POKEQ,18:PRINT:PRINT"LICZBA GRA
CZY? (1-3)"
170 GETGR*: IFGR*("1"ORGR*)"3" THEN
170
180 FORI=1TOVAL(GR*):G(I)=0:NEXT:PO
KEQ,18:PRINT:PRINT"JESTES KONSTRUKT
OREM?"
190 PRINT:PRINT"1 - POCZATKUJACYM":
PRINT"2 - PRZECIETNYM":PRINT"3 - ZA
AWANSOWANYM"
200 GETKR*: IF KR*("1"ORKR*)"3" THEN
200
210 M=3-(KR*("3")-(KR*("2")):KR=4-VA
L(KR*)
220 POKEQ,17:PRINT:PRINTSPC(26)B*"P
UNKTACJA":PRINT:GR=VAL(GR*)
230 FORI=1TOGR:PRINTSPC(24)B*"GRACZ
"I;L*": "C*"0000":P(I)=0:NEXT:GOSUB
560
240 REM KOLEJNA GRA
250 POKEQ,17:PRINT:PRINTA*"RUNDA"C*
RN;S*A*"GRACZ"C*G
260 FORI=1TOM:M(I)=0:NEXT:FORI=1TOM
270 R=INT(RND(0)*9)+1:J=1
280 IFR=M(J)THEN270
290 J=J+1:IFJ<=MTHEN280
300 M(I)=R:NEXT:FORI=M105:M(I)=M(M)
: NEXT

```

```

310 W=INT(RND(0)*900)+100:R*="00000
0":N=1:O=1:L=0:F=0
320 POKEQ,20:PRINT:PRINTSPC(7)C*"GO
TOW!":FORI=0TO29:POKEKL+I,7:NEXT
325 WAITB,1:GETZ*:POKEQ,20:PRINT:PR
INTSPC(5)S*S*
330 POKEQ,6:PRINT:PRINTSPC(10);:FOR
I=1TOM:PRINTM(I);:NEXT
340 PRINTS*S*:PRINT:PRINT:PRINTSPC(
9)W:POKEQ,9:PRINT:PRINTSPC(9)B*D*
350 POKEQ,14:PRINT:FORI=1TO7:PRINTS
*;;NEXT:PRINTK*SPC(4)S*C*:TI*="0000
00"
355 POKEQ,6:PRINT:PRINTSPC(32)KR;L*
":00"
360 AZ=PT+N*2:ZN=PEEK(AZ):POKEAZ,ZN
+128:IFNOT((N=15)AND(O<0))THENO=-O
370 GETZ*:POKEB,0:IFVAL(MID*(TI*,4,
1))>=KRTHEN610
380 T*=TI*:POKEQ,6:PRINT:IFT*=R*THE
N410
390 R*=T*:POKEK,33:PRINTCHR*(47+KR-
VAL(MID*(T*,4,1)))":";
400 PRINTCHR*(53-VAL(MID*(T*,5,1)))
CHR*(57-VAL(MID*(T*,6,1)))
410 IFZ*=" THEN370
420 Z=ASC(Z*):C=VAL(Z*):IFZ=13 THEN
610
430 F=(N+1)/2:IF(Z=42ORZ=43ORZ=45O
RZ=47)AND(O<0)ANDN<16THENO(F)=Z:GOTO
480
440 IF(C=M(1)ORC=M(2)ORC=M(3)ORC=M
(4)ORC=M(5))AND(O<0)ANDN<16THENC(F)=
C:GOTO480
450 IFZ=20ANDN=LANDN>0 THEN POKEAZ,
32:L=L-1:O=-O:C(F)=-C(F)*(N>1):GOT
O490
460 IFF=1ORNOT((Z=29 AND(N<L+1)AND
N<15)OR(Z=157ANDN>1))THEN370
470 IF(Z=29 AND(N<L+1)ANDN<15)OR(Z
=157ANDN>1)THENPOKEAZ,ZN:N=N-(Z=29
OR1):GOTO360
480 PRINTC*:POKEAZ,Z:L=L-(N>L):N=N
-(N<15)
490 POKEQ,14:PRINT:PRINTSPC(33)S*::
FORI=1TO8:W(I)=C(I):NEXT
500 FORI=1TO7:W(I+1)=(O(I)=45)*W(I
+1)-(O(I)<45)*W(I+1):NEXT:F=0:W%=
0:J=L:I=0
510 I=I+1:IFNOT(O(I)=42ORO(I)=47)O
RI)=.1/2THEN530
520 W(I+1)=-((W(I)*W(I+1))*(O(I)=42
)-(W(I)/W(I+1))*(O(I)=47):GOTO510
530 H=W%:IFH+W(I)>9999ORH+W(I)<0TH
EN570
540 W%=W%+W(I):IFI<J/2THEN510
550 POKEQ,14:PRINT:PRINTSPC(32)W%:
F=0:POKEQ,14:PRINT:PRINTSPC(32)A*D
*C*:GOTO360

```

```

560 POKEQ,18:PRINT:FORI=0TO4:FORJ=
0TO21:PRINTK*;;NEXT:PRINT:NEXT:RET
URN
570 POKEQ,19:PRINT:PRINT"PRZEKROCZ
ONO ZAKRES":PRINT"LICZB! (0-9999)"
580 POKEAZ,32:N=N+(N<15):L=L+(N=LA
NDL<>15):O=-O
590 POKEQ,14:PRINT:PRINTSPC(33)S*:
WAITB,1
600 GOSUB560:GOTO360
610 POKEAZ,ZN:E=W-W%:E=E+2*E*(E<0)
:E=-E*(E<=100)-100*(E>100)
620 TR=KR*60-(60*VAL(MID*(T*,4,1))

```

```

+VAL(RIGHT*(T*,2)))
630 U=INT((100-E)*VAL(KR*)*TR/10):
G(G)=G(G)+U:POKEQ,18+G:PRINT
635 PRINTSPC(33)S*CHR*(145):PRINTS
PC(32)G(G)
640 G=G+1:IFG>VAL(GR*)THENG=1:RN=R
N+1
650 IFRN<6THEN240
660 POKEQ,20:PRINT:PRINT"KOLEJNA G
RA? (T/N)"
670 GETZ*:IFZ*<"T"ANDZ*<"N"THEN
670
680 PRINTJ*:IFZ*="T"THENRUN

```

Zaczarowana skrzynka

Racuchy pałaszowały jeden za drugim, a przy tym wbrew upomnieniom pani wychowawczynie, bardzo hałasowały i gestykowały. Jak się okazało, tyle emocji wśród przedszkolaków wzbudziła „zaczarowana skrzynka” czyli komputer Amstrad 6128.

— Jeszcze na stałe nie posiadamy komputera — powiedziała Marianna Błaszczak, dyrektorka w warszawskim przedszkolu, mieszczącym się przy ulicy Koszykowej, ale nosimy się z zamiarem jego zakupu. Może nie Amstrad, bo co tu ukrywać, nie dysponujemy aż tak dużymi finansami, ale może Commodore lub Spectrum.

Rodzice przyklasnęli inicjatywie dyrektorki przedszkola. „Niech nasze dzieci nie zostają w tyle za nowinkami techniki — powiadają. Komputery w wielu krajach traktowane są jako narzędzie pracy, a u nas wciąż wzbudzają wiele emocji. A wiadomo czym skorupka za młodu nasiąknie...”

Na razie sześciolatki korzystają z

uprzejmości zaprzyjaźnionych placówek i... zamożniejszych rodziców.

W ciągu roku grupa zerowa musi poznać 22 litery wielkie i małe. Nauczyć się czytać proste zdania i krótkie teksty. Liczyć od 0 do 10, poznać cyfry, znaki i podstawowe pojęcia matematyczne. Muszą umieć układać wzory z poznanych znaków. W tym wszystkim mógłby pomóc im właśnie komputer — chociaż...

Pani wychowawczynie dużo wcześniej zapowiedziała zajęcia z komputerem. Kiedy wreszcie będzie — niecierpliwiły się przedszkolaki. Ja tam już widziałem — chwalił się Krzysiek. I co, co — dopytywali się rówieśnicy.

— Wujek grał z komputerem w karty a dla mnie włączył wyścigi. Mówię wam bomba. Emocje doszły zenitu. Nareszcie po śniadaniu spotkanie z „wielką niewiadomą”.

Amstrad „zawitał” do świetlicy. Nim pracownik Wojskowego Instytutu Informatyki zdążył cokolwiek omówić, dzieci zarzuciły go mnóstwem pytań. A po co są te klawisze! A co to za ekran? A te blaszki do czego służą? Każde chciało dotknąć skrzynkę, wydającą dziwne dźwięki, nacisnąć chociaż jeden klawisz.

Proszę się nie przepychać — karcita przedszkolaków pani wychowawczynie, Elżbieta Kąca. Wszyscy zobaczcie. Za klawiaturą zasiada Kamil. Dla niego to nie nowina. Komputer ma w domu. Umiejętnie zdobywa punkty, zbijając „Uciekającego Franciszka”. Ja bym chciał mieć takie cacko w domu — wzdycha Paweł. Dziewczynki bardziej sceptycznie się wypowiadały. Może jednak wolą Zuzię, lalę niedużą...

— Według mnie — powiedział pracownik Wojskowego Instytutu Informatyki, mjr Romuald Głąb — nie powinniśmy wpadać w przesadną euforię.



Jak do tej pory, na rynku nie pojawiły się programy opracowane dla potrzeb szkolnictwa, a zwłaszcza grup zerowych. Programy z prawdziwego zdarzenia, które po konsultacji nauczycieli z zawodowymi informatykami mogłyby pomóc, a nie zaszkodzić w nauczaniu początkowym. Chałupnicze wymysły, to nie to. Zresztą jeden komputer na 30 dzieci, to przysłowiowa kropla w morzu — podsumował.

No cóż, od czegoś należy zacząć. Poprzednia grupa zerowa uwielbiała zajęcia z komputerem (wypożyczonym). Na ekranie ukazywały się proste działania i jeśli je dobrze wykonały, premiowane były punktami. W przeciwnym wypadku rozlegał się sygnał — popraw się. Kilku sześciolatków, w domu, uczyło się języka LOGO, w którym się programuje, tworzy za pomocą prostych instrukcji — obrazy.

— Wszystkie zaczęłyśmy się do kształcać — powiedziała pani Błaszczak. Przecież nie możemy zostać w tyle... za naszymi wychowankami, z których w przyszłości może wyrosną informatycy.

Jak na razie dzieci wychodzą na przywitanie dnia jutrzejszego. Uczą się myślenia twórczego, by w 2000 roku, już jako dwudziestolatki nie miały poczucia straconego czasu.

ANNA TOMCZYKOWSKA



Foto: Jan Zelman

Zmienne systemowe ROM SPECTRUM

WSKAŹNIKI RAM

PROG 23635/6
 VARS 23627/8
 E—LINE23641/2
 WORKSP 23649/50
 STKBOT 23651/2
 STKEND 23653/4
 ERR—SP 23613/4
 RAMTOP 23730/1
 UDG 23675/6
 P—RAMT 23732/3

Kiedy piszemy program w BASIC-u, pamięć RAM jest na bieżąco porządkowana. Komputer musi wiedzieć wszystko, a będzie to możliwe wtedy, gdy nasze posunięcia będą kontrolowane. Z tego powodu RAM oznaczona jest wieloma wskaźnikami. Wszystkie zajmują w obszarze zmiennych po 2 bajty, gdyż zawierają adresy.

P—RAMT przechowuje wielkość adresowanej pamięci, a ściślej bajt o najwyższym adresie. Dla Spectrum 48K jest to adres 65535. Jeśli coś nie jest w porządku z naszym przyjacielem, lepiej sprawdzić zawartość tej zmiennej. Uzyskamy informację, czy w porządku są moduły RAM.

Program w BASIC-u lokuje się od adresu przechowywanego przez PROG. Zwykle jest to 23755, jednak po przyłączeniu INTERFACE 1 wynosi on 23812, co wynika z faktu pojawienia się obszaru mapy microdrive'u.

Pozostałe wskaźniki wyznaczają początek następujących obszarów:

- | | |
|--------|---|
| VARS | — zmienne BASIC-u |
| E—LINE | — bufor edycyjny, gdzie redaguje się linie BASIC, |
- | | |
|--------|--|
| WORKSP | — przestrzeń robocza, gdzie trzymane są dane wczytane w czasie egzekucji komendy INPUT oraz tworzony jest nagłówek po wczytaniu instrukcji SAVE... |
|--------|--|
- | | |
|--------|---|
| STKBOT | — stos kalkulatora, na którym zachodzą wszystkie obliczenia arytmetyczne, |
|--------|---|
- | | |
|--------|---|
| STKEND | — wolna pamięć zmniejszająca się w miarę dodawania linii do programu. |
|--------|---|

Granica ekspansji programu BASIC jest adres trzymany przez RAMTOP, zmniejszony o 80. Przy przekroczeniu tej wielkości otrzymamy komunikaty *Out of memory* lub *No room for line*. W trakcie powstawania programu użyteczne mogą być następujące informacje:

```

9000 CLS: PRINT '''
9010 PRINT "dług. programu",FN a(23
627,23635)
9020 PRINT "dług. zmiennych",FN a(2
3641,23627)-1
9030 PRINT "wolna pamiec",65536- US
R 7962
9040 PRINT "ramtop", PEEK 23730+256
* PEEK 23731
9050 DEF FN a(h,1)=PEEK h- PEEK 1+2
56*(PEEK (h+1)-PEEK (1+1))
  
```

Od RAMTOP w dół pamięci odkłada się stos GOSUB'ów, na którym przechowywane są 3-bajtowe (numer linii i wyrażenia) informacje o powrocie z podprogramu po komendzie RETURN. Oto przykład wymuszania powrotu:

```

10 GOSUB 100: REM stos gosub'ow zaw
iera linie 10 i wyrażenie 2 jako po
wrot
20 PRINT "liczba dodatnia": STOP
100 INPUT "podaj dowolna liczbe";zn
ak
110 IF SGN znak=1 THEN RETURN
120 LET linia=200: LET wyr=1
130 LET stos=2+ PEEK 23613+256*PEEK
23614
140 POKE stos,linia-256* INT (linia
/256)
150 POKE stos+1, INT (linia/256)
160 POKE stos+2,wyr
170 RETURN
200 PRINT "liczba nie jest dodatnia
": STOP
  
```

Poniżej stosu podprogramów odkłada się stos maszynowy, a rozdziela je adres obsługi wydruku raportów błędów, który zawsze jest wskazywany przez zmienną ERR—SP.

RAMTOP, jako jedyną zmienną systemową, można ustawiać komendą CLEAR adres. Początkowo wynosi on 65367. Zaraz za nim (65368) rozpoczyna się obszar grafik użytkownika, który wskazuje UDG. Zaletą tego obszaru (168 bajtów) jest możliwość umieszczenia go w innym, rozsądnym miejscu RAM, choćby w buforze drukarki, gdy nie jest przyłączona.

Jeśli nie wystarcza 21 grafik użytkownika, można ich liczbę powiększyć.

```

200 RESTORE 200+obszar*10
202 READ bajt1,bajt2
204 POKE 23675,bajt1
206 POKE 23676,bajt2
208 RETURN
210 DATA 8,254
220 DATA 176,254
230 DATA 88,255
  
```

Ten podprogram umożliwia korzystanie z 63 znaków graficznych, lecz na każdy z obszarów trzeba się przełączyć. Odbywa się to przez sekwencję np.:

```
LET obszar = 1 : GOSUB 200
```

Konieczne jest na początku programu głównego obniżenie RAMTOP przez CLEAR 65031.

Sposobów przechowywania procedur w kodzie maszynowym jest kilka. Najpopularniejsze z nich to bezpośrednio wczytywanie pliku z nośnika (taśmy, dyskietki) do określonego obszaru pamięci, trzymanie bajtów w instrukcjach DATA oraz przechowywanie ich w instrukcjach REM. Ten ostatni sposób jest używany najczęściej. Pierwszą linię programu zaopatrujemy komendą REM i tyłoma znakami, ile bajtów zajmuje procedura. Następnie wczytujemy kolejne bajty, korzystając z podprogramu:

```

10 REM ilosc znakow rowna dlugosci kodu
20 INPUT "dlugosc kodu: ";d
30 IF d<1 THEN GOTO 20
40 LET prog= PEEK 23635+256* PEEK 23636
50 FOR a=1 TO d
60 INPUT bajt
70 IF bajt>255 OR bajt<0 THEN GOTO 60
80 POKE prog+4+a,bajt
90 NEXT a
100 PRINT "adres początkowy =";prog+5
  
```

PROGRAM

36

Problem jest prosty: rozwiązanie trójkąta o danych trzech bokach. Po wprowadzeniu danych obliczony jest obwód trójkąta, pole (ze wzoru Herona) oraz wszystkie kąty (z twierdzenia Carnota). Linie 360... 430 szkicują trójkąt proporcjonalnie do wymiarów ekranu. Wysokość prowadzona jest do najdłuższego boku. Objaśnienia:

- 1) Nie ma potrzeby deklarowania zmiennych rzeczywistych, gdyż Amstrad robi to automatycznie.
- 2) Kolory ustala się poprzez podwójne przyporządkowanie, co może stwarzać nieco kłopotów debiutantom. Najlepiej zacząć od „napłynienia” kałamarzy (INK) odpowiednimi kolorami tuszu (linie 50... 70).
W trakcie pisania programu wylewamy tusz z kałamarza na tło (PAPER) lub maczamy w nim pióro (PEN). Jeśli definiujemy okna trzeba określić dla nich odpowiednie atrybuty (linie 80... 90).
- 3) W stosunku do standardu Spectrum uproszczono maksymalnie definiowanie grafik użytkownika (linie 100...110).
- 4) Wprowadzenie funkcji użytkownika należy umieszczać przed korzystaniem z nich (w Spectrum można je umieszczać w dowolnym miejscu programu).
- 5) Czyszczenie okien najlepiej przeprowadzać od największego do najmniejszego (linia 170). Należy pamiętać, że CLS niszczy tryb obliczania funkcji trygonometrycznych w stopniach katowych.
- 6) Deklaracja tablicy 11-wymiarowej (linia 150) tworzy pamięć dla 11 + 1 zmiennych uwzględniając indeks zerowy (linie 310...330).
- 7) Przed wprowadzeniem danych z klawiatury dobrze jest wyczyścić bufor, w którym Amstrad skrupulatnie odnotowuje każdy przyciśnięty klawisz (linia 190).
- 8) Pomysłowa instrukcja MASK pozwala na tworzenie różnych linii. Argumentem jest wartość 8 bitów, które będą logicznie mnożone przez każde kolejne 8 punktów linii. Wartość MASK obowiązuje od momentu określenia do zmiany wartości argumentu.

dokończenie ze str. 14

Na zakończenie metoda ochrania programu w języku BASIC, polegająca na zapisie na nośnik sekwencji bajtów symulujących program w języku wewnętrznym.

```
9900 INPUT "nazwa zbioru ",LINE n$
9910 IF LEN n$>8 THEN IF USR 1602 THEN
9920 LET s= PEEK 23653+256* PEEK 23654
9930 SAVE n$ CODE 23552,s-23500
9940 RUN
```

Autostart programu następuje dzięki linii 9940. Proszę zwrócić uwagę, że zachowany zostaje cały obszar zmiennych systemowych, m.in. zmienna przechowująca numer ostatnio wykonanej linii.

W kolejnym numerze kilka sztuczek z grafiki.

Krzysztof MAMCARZ

AMSTRAD

- 9) Amstrad nie ma kłopotów z zaokrągleniem liczby do kilku cyfr po przecinku, co realizuje wygodne ROUND (linie 460...470). Jednak zaokrąglenie części ułamkowej kąta nie pada minut katowych, lecz wartość w systemie dziesiętnym. Przejście na minuty realizuje linia 520.
- 10) Niefortunny brak funkcji arcsin, arccos czy arctg zmusza do odwiedzenia biblioteki (linia 130).

K. M.

```
10 * ** trojkat 86
20 * deklaracje funkcji i zmiennych
30 * definicje grafik i kolorow
40 BORDER 6
50 INK 0,21
60 INK 1,0
70 INK 2,22
80 WINDOW #1,30,40,1,15
90 PAPER #1,3:PEN #1,2
100 SYMBOL 255,22,12,24,52,24,12,22,0
110 SYMBOL 254,6,9,9,6,0,0,0,0
120 DEF FN k(a,b,c)=-((a^2-b^2-c^2)/(2*b*c))
130 DEF FN l(a)=90-ATN(a/SQR(1-a^2))
140 DEF FN p(o,a,b,c)=SQR(o/2*(o/2-a)*
*(o/2-b)*(o/2-c))
150 DIM k(3)
160 * wprowadzanie danych
170 CLS:CLS#1:DEG
180 MASK 255
190 CLEAR INPUT
200 INPUT #1,"bok 1 ";a
210 INPUT #1,"bok 2 ";b
220 INPUT #1,"bok 3 ";c
230 IF ABS(b-c)<a AND a<b+c THEN ELSE
FRINT"trojkat nie istnieje":GOTO 540
240 * obliczenia
250 k(1)=FN k(a,b,c)
260 k(2)=FN k(b,c,a)
270 k(3)=FN k(c,a,b)
280 ob=a+b+c
290 m=MAX(a,b,c)
300 n=MIN(a,b,c)
310 k(0)=FN l(FN k(ob-m-n,m,n))
320 x=n*COS(k(0))
330 y=n*SIN(k(0))
340 m1=300/m
350 * konfiguracja trojkata
360 ORIGIN 20,20
370 PLOT 0,0
380 DRAW m*m1,0
390 DRAW ::*m1,y*m1
400 DRAW 0,0
410 MOVE ::*m1,y*m1
420 MASK 219
430 DRAW ::*m1,0
440 * wydruk wynikow
450 CLS #1
460 PRINT#1,"obwod=";ROUND(ob,2)
470 PRINT#1,"pole =";ROUND(FN p(ob,a,
b,c),2)
480 FOR a=1 TO 3
490 PRINT#1
500 PRINT#1,CHR$(255);
510 IF a=1 THEN PRINT#1,"(a,b)" ELSE
IF a=2 THEN PRINT#1,"(b,c)" ELSE PRIN
T#1,"(c,a)"
520 PRINT#1,FIX(k(a));CHR$(8);CHR$(2
54);ROUND((k(a)-FIX(k(a)))*60);CHR$(
8);CHR$(39)
530 NEXT
540 PRINT"inne dane ? T/N"
550 IF INKEY$(51)=-1 THEN 550 ELSE 170
```



KLUBY KOMPUTEROWE MŁODYCH MISTRZÓW

...w których przygoda z komputerem jest na wysokim poziomie". Autorzy najlepszych rozwiązań i pomysłów są laureatami Konkursu im. J. Piłsudskiego w dziedzinie Kultury i Nauki.





PUTEROWE RZÓW TECHNIKI

już „bliskim spotkaniem III
programów spotkali się w Pa-

Foto: Jan Zelman



/.!/.!/.!/.!/.!/.!/.!/.!/.!/.!
ale fajny szlaczek
/.!/.!/.!/.!/.!/.!/.!/.!/.!/.!

W zależności od tego, co umieściliśmy jako parametr procedury „w ramce”, to właśnie było umieszczone między szlaczkami. Najpierw była to liczba, później słowo (napis po cudzysłowie) w końcu LISTA słów ujęta w nawiasy kwadratowe. Parametr określa zatem dowolną pod względem znaczenia rzecz — obiekt, którego sens widać dopiero przy wykonywaniu procedury.

Czas teraz przystąpić do napisania „poważnego” programu (zespołu procedur) w LOGO. Najpierw postaramy się wyspecyfikować problem. Jest to bardzo ważny element (nie ma się co śmiać) projektowania każdego programu, czy całego systemu programów. Wiele problemów pojawiających się w profesjonalnej informatyce (choćby nieprzewidziane błędy w programie), wielkie tzw. pady systemów, są głównie spowodowane zaniedbaniem w początkowej fazie projektowania (zawodowcy dosyć dobrze programują). Specyfikacje zresztą to już cała gałąź informatyki.

Stawiam zatem ogólny problem:

realizacja testu z dowolnej dziedziny (może być z wielu) wiedzy. A ściślej: chcę aby komputer zadawał pytania, sprawdzał poprawność odpowiedzi — wyświetlał np. „Brawo” lub „Niedobrze” oraz liczył punkty — dobrze 1 punkt, źle 0. Na końcu powinien podać liczbę zdobytych punktów. Ocenę możemy sobie na razie darować. Pytania i prawidłowe odpowiedzi podajemy redagując (korzystając z edytora—redaktora) pewną procedurę.

Taka specyfikacja powinna być wystarczająca do zbudowania prostego systemu procedur (z jedną oczywiście główną). Od niej też zaczniemy budowanie naszego programu. Zastanówmy się jak ma przebiegać testowanie. A więc, inicjalizacja (wyświetlenie tytułu itp.), zadawanie pytań i zakończenie, czyli podanie wyniku i pożegnanie. Po prostu:

? oto test
> inicjalizacja
> pytania
> zakończenie
> już
TEST zdefiniowanie

Prawda, że proste? Oczywiście pozostaje zdefiniowanie procedur „inicjalizacja”, „pytania” i „zakończenie”. No więc do roboty. Do pamiętania punktów, dla ułatwienia, użyjmy sobie zmiennej:

? oto inicjalizacja „przyp” to skrót od
> przyp „punkty 0” „przypisz”
> pp [Dzień dobry. Powodzenia:] „pp” to skrót od „pisz”
> już
INICJALIZACJA zdefiniowane

Teraz pytania. Musimy wykazać się inwencją (na razie się bawimy) i wymyślić jakieś testowe pytania. Na przykład:

? oto pytania
> pytaj [Stolica ZSRR] [Moskwa]
> pytaj [Ile jest 11x11?] [121]
> pytaj [Prószy w zimie?] [śnieg]
> pytaj [Często miauczy?] [kot]
> już
> PYTANIA zdefiniowane

Procedura „pytaj”, którą zdefiniujemy, ma dwa parametry. Jeden to pytanie, a drugi to prawidłowa odpowiedź. Oba w

konkretnym naszym wywołaniu są LISTAMI słów (w nawiasach KWADRATOWYCH). „Najtrudniejsza” procedura „pytaj” może wyglądać tak:

? oto pytaj : pyt: odp.
pp : pyt
jeśli czytaj listę = odp (pp „Brawo!”)
przyp „punkty: punkty + 1” (pp „Niedobrze!”)
już
PYTAJ zdefiniowane

Słowo „czytaj listę” to instrukcja WEJŚCIA. W miejsce tego słowa zostanie wstawione to, co testowany delikwent wpisze z klawiatury — odpowiedź. Równoprawnym skrótem jest „cl”. „Czytaj listę” to LISTA słów wprowadzona z klawiatury, aż do naciśnięcia ENTER. Jeśli wciśniemy od razu ENTER, lista będzie pusta — J. Może też to być jedno słowo np. kot i ENTER, a wtedy lista ta to (kot). W tym miejscu jeszcze raz przypomnę, że instrukcja „jeśli”, razem ze swoimi listami do wykonania, w zależności od warunku, musi być w jednej linii logicznej. Nie można po drodze wcisnąć klawisza ENTER. Tutaj w procedurze „pytaj”, będzie się ona mieścić w 3 wierszach na ekranie, a dwa wiersze będą kontynuowane (znaczek „!” w ostatniej kolumnie na ekranie).

Pozostaje jeszcze problem „zakończenia”. Nic prostszego:

? oto zakończenie
> pp „koniec!”
> pp (liczba zdobytych punktów:)
> pp: punkty
> pp: [Do widzenia!]
> już
ZAKOŃCZENIE zdefiniowane

i koniec. Wywołajmy teraz procedurę „test” i posadźmy kogoś przy klawiaturze — będzie trochę śmiechu! Można rozszerzyć repertuar pytań i prawidłowych odpowiedzi. Trzeba w tym celu przeredagować procedurę „pytania” (umiemy chyba korzystać z edytora — red. „pytania”) i dodać np.:

pytaj [ile jest 13x13] [169]
pytaj [polskie morze] [Bałtyk]

dokończenie na str. 20



Rys. M. Przybyłowski

Zauważmy, że odpowiedź „Bałtyk” będzie zła. Należy wpisać ją dokładnie tak jak jest w nawiasie kwadratowym, czyli zacząć wielką literą. Dla przypomnienia sposobu pracy z edytorem — redaktorem, napiszę jak dodać nowe linie do procedury „pytania”. A więc : najpierw piszemy:

? red. pytania

„wesliśmy” w edytor i na dole jest to sygnalizowane: Edytor LOGO... Teraz „najeżdżamy” kursorem (używając strzałek pionowych i poziomych) na koniec linii, po której chcemy wstawić nową i wciskamy ENTER. Mamy nową pustą linię. Wpisujemy tam treść nowej linii. Aby zapamiętać zmiany, wciskamy Extended mode i klawisz „C”. Będzie:

PYTANIA zdefiniowane

Dla przypomnienia kilka instrukcji edytora — redaktora:

Extended mode i

- | | |
|------------------|---|
| strzałka w prawo | skok na koniec linii |
| strzałka w lewo | skok na początek linii |
| strzałka w górę | skok na początek ekranu |
| strzałka w dół | skok na koniec ekranu |
| klawisz „B” | skok na początek tekstu (może być na kilku ekranach) |
| klawisz „E” | skok na koniec tekstu |
| klawisz „N” | przejdźcie do następnej strony |
| klawisz „P” | przejdźcie do poprzedniej strony |
| klawisz „C” | zapamiętanie zmian |
| klawisz „Y” | skasowanie linii na prawo od kursora |
| klawisz „R” | powielanie ostatnio skasowanej lub wprowadzonej linii |

kasowanie liter — kursor na prawo od kasowanej litery i klawisz DELETE wyjście z edytora bez dokonania zmian — BREAK
Redagowanie (wejście w edytor) to napisanie

? red. „nazwa lub
? red. [nazwa 1 nazwa 2... nazwa]
i wtedy redagujemy więcej niż jedną procedurę
wejście w pusty edytor (tylko po co?) to

? red []

wejście w ostatnio używany edytor to po prostu

? red

Extended mode i klawisz „Y” lub „R” można też używać poza edytorem do usprawnienia naszej konwersacji. Sprawdźcie to.

Na zakończenie przedstawiam cały program (5 procedur — główna „test” — ją wywołujemy) realizujący testowanie. Wydruk został zrobiony na drukarce Seikosha GP50. O tym jak współpracować z drukarką i magnetofonem, będzie jeszcze mowa w następnych odcinkach.

```

T
OTO TEST
INICJALIZACJA
PYTANIA
ZAKOŃCZENIE
JUŻ
    
```

```

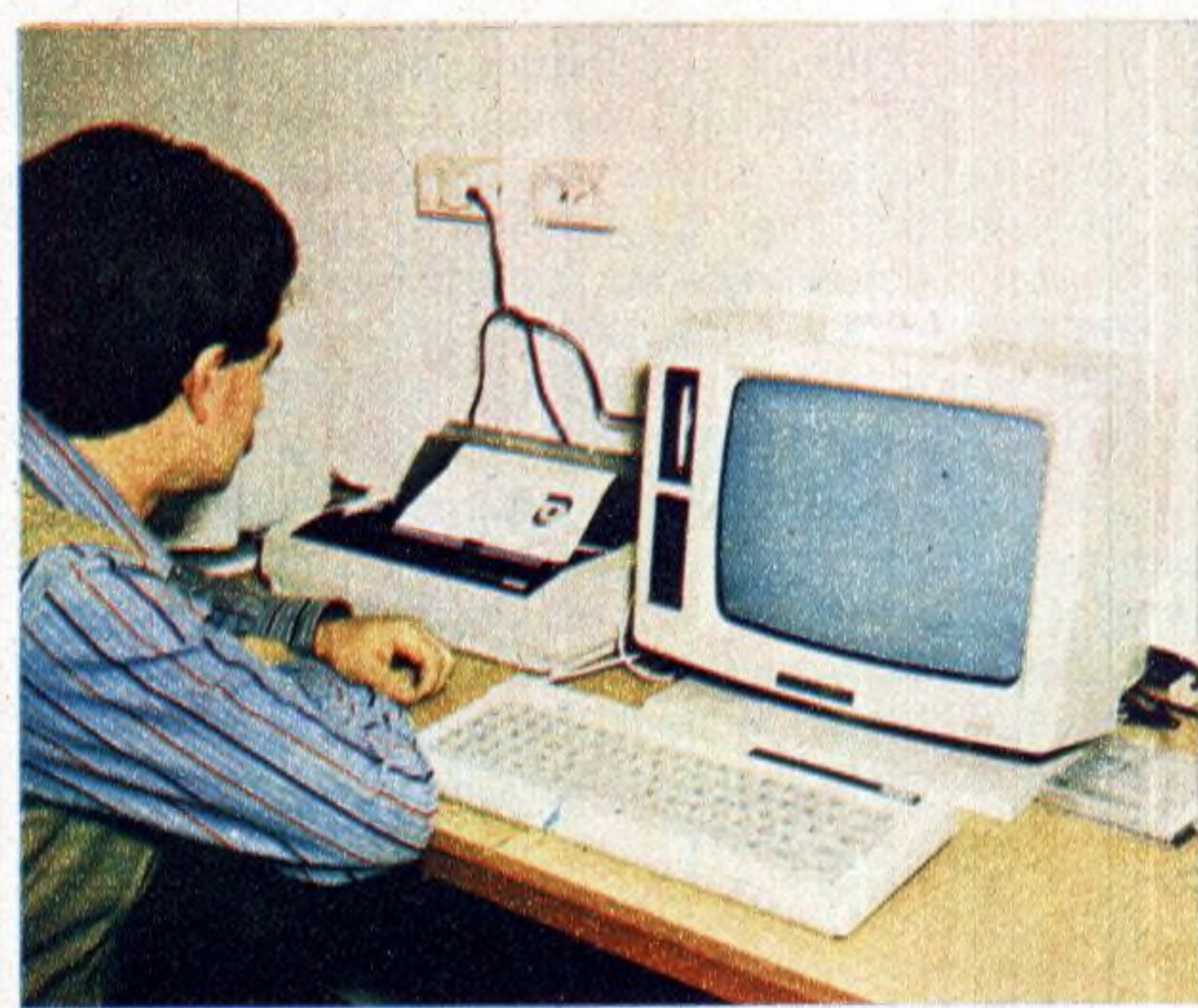
OTO INICJALIZACJA
PRZYP "punkty 0
PP [dzień dobry. Powodzenia!]
JUŻ

OTO PYTANIA
PYTAU [Stolica ZSRR?] [Moskwa]
PYTAU [Ile jest 11 + 11] [22]
PYTAU [Czasz w zimie?] [zimno]
PYTAU [Często miauczy?] [kot]
JUŻ

OTO ZAKOŃCZENIE
PP "Koniec!
PP [liczba zdobytych punktów:]
PP :punkty
PP [Do widzenia!]
JUŻ

OTO PYTAU :pyt :odp
PP :pyt
JESLI CZYTAJLISTA = :odp [pp "Br
swol! PRZYP "punkty :punkty + 1]
[pp "Niedobrze!]
JUŻ
    
```

SIS



Fani mikro mają nowe miejsce „pielgrzymek”. W przejściu podziemnym u zbiegu Wolskiej i Młynarskiej na warszawskiej Woli firma System oferuje mikrokomputery, urządzenia peryferyjne i programy. Najwięcej jak zwykle oglądających...
Foto: Jan Zelman



METODA ELIMINACJI ELEMENTÓW GŁÓWNYCH

Jak rozwiązać układ równań liniowych? Metod jest wiele. Jednak każda z nich jest uciążliwa, szczególnie kiedy dysponujemy jedynie kartką papieru, a równań w układzie jest więcej niż 3.

Zacznijmy zatem od początku. Mamy rozwiązać układ równań zapisany ogólnym wzorem:

$$\sum_{j=1}^n a_{ij}x_j = a_{i,n+1} \quad (i = 1, 2, \dots, n)$$

Do rozwiązania tego układu, czyli znalezienia wszystkich x_j posłużymy się metodą eliminacji elementów głównych, polegającą na kolejnym eliminowaniu współczynników układu w porządku zależnym od ich wielkości.

Pierwsze równanie dzielimy stronami przez ten współczynnik a_{1j} , który ma największą wartość bezwzględną. Związaną z nim niewiadomą eliminuje się z pozostałych równań. Z otrzymanym układem $n-1$ równań z $n-1$ niewiadomymi postępujemy podobnie: eliminujemy z niego niewiadomą, której w pierwszym równaniu nowego układu towarzyszy współczynnik o największej wartości bezwzględnej. Analogicznie eliminujemy następne niewiadome. W efekcie otrzymujemy układ równań zawierających odpowiednio $n, n-1, n-2, \dots, 2, 1$ niewiadomych. Otrzymujemy również numery p_1, p_2, \dots, p_n kolejno eliminowanych niewiadomych. Ciąg ten powstaje stopniowo, w miarę eliminacji niewiadomych, z ciągu $1, 2, \dots, n$. Po ustaleniu, że najpierw należy wyeliminować niewiadomą x_j , przestawia się liczby 1 i j :

$$j, 2, \dots, j-1, 1, 1-2, \dots, n.$$

Za pomocą pierwszego równania eliminuje się z pozostałych równań tę niewiadomą, której numer występuje w powyższym ciągu na pozycji pierwszej i oblicza się nowe współczynniki przy pozostałych niewiadomych. W następnym kroku bada się tylko współczynniki przy tych właśnie niewiadomych w drugim równaniu. Numer niewiadomej, której odpowiada współczynnik o największej wartości bezwzględnej, przestawia się z drugą liczbą powyższego ciągu.

Za pomocą pierwszego równania (zredukowanego) eliminuje się z pozostałych równań tę niewiadomą, której numer występuje w nowym ciągu

wskaźników na drugiej pozycji. Podobne operacje wykonujemy, aby wyeliminować pozostałe niewiadome.

Równania otrzymanego układu zawierają coraz mniej niewiadomych. Kolejne x_j wyznaczamy w kolejności odwrotnej do porządku eliminacji.

Powyższą metodę najłatwiej sprawdzić oczywiście na niewielkim układzie równań — większe rozwiąże poniższy program.

Pierwszy z nich jest wiernym zapisem przedstawionego algorytmu w języku ALGOL, drugi natomiast jest jego odpowiednikiem BASICu na Spectrum.

```
BEGIN
  COMMENT rozwiązanie układu
  metodą elementów głównych
  INTEGER h, H, i, j, n, N, ph, pj;
  REAL s, t;
  ininteger (1, n);
  N := n+1;
  BEGIN
    INTEGER ARRAY p [1 : N];
    ARRAY a [1 : n, 1 : N], x [1 : n];
    inarray (1, a);
    FOR i := 1 STEP 1 UNTIL n DO
```

```
5 REM *****
7 REM
10 REM Program rozwiązuje n równan liniowych o n niewiado-
nych metoda elementów glow- nych.
13 REM
15 REM *****
20 INPUT "Podaj liczbę zmiennych "in
30 LET z:=n+1
40 DIM p(z)
50 DIM x(n)
60 DIM a(n,z)
65 REM
67 REM *****
69 REM
70 REM Rozpoczyna sie wprowa-dzanie kolejnych wspolczynnkow układu
72 REM
75 REM *****
80 FOR i=1 TO n
90 FOR l=1 TO z
100 INPUT "Podaj wspolczynnik a('i(i)l','l(l)l') "fa(i,l)
110 NEXT l
120 NEXT i
130 FOR i=1 TO z
```

```
p[i] := i;
FOR h := 1 STEP 1 UNTIL n DO
  BEGIN
    s := 0;
    FOR i := h STEP 1 UNTIL n DO
      BEGIN
        t := abs (a[h,p [i]]);
        IF t >= s
          THEN BEGIN
            s := t;
            j := i
          END t >= s
      END i;
    ph := p [j];
    p [j] := p [h];
    p [h] := ph;
    s := a [h, ph];
    H := h+1
    FOR j := H STEP 1 UNTIL n Do
      BEGIN
        pj := [j];
        t := a [h, pj]: a [h, pj] /s;
        FOR i := H STEP 1 UNTIL n DO
          a [i, pj]: = a [i, pj] - a [i, ph]* t
        EDN j
      END h;
    FOR h := n STEP - 1 UNTIL 1 DO
      BEGIN
        s := a [h, N];
        FOR j := h+1 STEP 1 UNTIL n
          DO
            BEGIN
              pj := p [j];
              s := s-a[h, pj]*[pj]
            END j;
            x [p[h]] := s
          END h;
        outarray (1, x)
      END
    END
```

dokończenie na str. 30

PASCAL⁽¹⁾

PASCAL jest jednym z najpopularniejszych języków programowania. Opracowany przez N. Wirtha w końcu lat sześćdziesiątych (opublikowany w roku 1971) szybko zdobył sobie niezwykle powodzenie. Zdecydowały o tym także jego cechy, jak:

- łatwość zapisu programu,
- możliwość wykorzystania do opisu licznej klasy problemów,
- przystosowanie do wymagań programowania systematycznego.

PASCAL jest obecnie powszechnie stosowany w nauczaniu programowania praktycznie na całym świecie. Nowe języki programowania w wielu wypadkach wzorowane są na języku PASCAL i jego konstrukcjach. PASCAL przyjął też rolę języka publikacyjnego, co oznacza, że większość algorytmów publikowanych w książkach i pismach specjalistycznych zapisywana jest właśnie w tym języku.

Niniejszy kurs programowania w języku PASCAL jest przeznaczony dla licznego kręgu odbiorców, którzy opowiadali podstawowe wiadomości z dziedziny programowania. Zakładamy jednak, że PASCAL może być pierwszym językiem programowania, z którym styka się Czytelnik. Z tego powodu nie odwołujemy się do innych języków programowania, ilustrujemy opis języka przykładami i proponujemy zadania do samodzielnego rozwiązania. Wszystkie prezentowane przykłady zostały uruchomione w systemie TURBO PASCAL, zaimplementowanym¹ na komputerze AMSTRAD CPC 6128, pracującym pod systemem operacyjnym CP/M PLUS. Wybierając system TURBO PASCAL, kierowaliśmy się następującymi względami:

- TURBO PASCAL obejmuje (poza nielicznymi wyjątkami) właściwości standardowego języka PASCAL, a ponadto dostarcza wiele możliwości ułatwiających programowanie i zwiększających zakres języka,
- TURBO PASCAL wyposażony jest w efektywny edytor ekranowy², ułatwiający tworzenie i poprawianie programów źródłowych,
- TURBO PASCAL umożliwia pełne wykorzystanie systemu operacyjnego.

O wyborze komputera zdecydowało to, że z powodu ceny jest on bardziej dostępny niż inne, na których również zaimplementowano system TURBO PASCAL, np. komputery serii IBM PC.

Całość kursu, ze względu na dość obszerny zakres materiału, została podzielona na kilka części. W każdej z nich oprócz opisu elementów języka, zamieszczamy opisy wybranych własności systemu TURBO PASCAL.

1. Podstawowe zasady korzystania z systemu TURBO PASCAL

System TURBO PASCAL obejmuje edytor, kompilator i zestaw programów pomocniczych. Ponieważ prezentowany TURBO PASCAL pracuje pod kontrolą jednego z systemów operacyjnych, CP/M PLUS lub CP/M-80 2.2, dowolny z nich należy załadować do pamięci komputera

przed użyciem TURBO PASCAL. Uzyskuje się to przez wprowadzenie z klawiatury komendy: CPM. Po załadowaniu systemu operacyjnego należy wprowadzić z klawiatury komendę TURBO i wcisnąć klawisz RETURN. Spowoduje to załadowanie kompilatora do pamięci komputera i wyświetlenie komunikatu wstępnego:

TURBO PASCAL system Version 3.00E

CP/M-80,Z80

Copyright 1985 AGROCOMPUTER

Terminal: AMSTRAD CPC6128

Include error messages (Y/N)?

zawierającego pytanie o sposób sygnalizowania błędów kompilacji. Na pytanie należy dać odpowiedź, wciskając klawisz **Y** lub **N**. Wciśnięcie klawisza **N** spowoduje, że błędy kompilacji będą sygnalizowane wyłącznie numerami. Wciśnięcie klawisza **Y** sprawi, że sygnalizacja błędów kompilacji będzie zawierała również słowne opisy błędów. Rezygnacja z opisów błędów przyniesie zwiększenie pamięci dostępnej dla programu o 1,5 K bajtów, jednak w pierwszym etapie nauki języka opisy błędów będą raczej przydatne. Po wprowadzeniu odpowiedzi zostaje wyświetlone tzw. menu główne, zawierające zbiór dostępnych komend systemu TURBO PASCAL:

Logged drive: A

Work file:

Main file: Edit Compile Run Save

execute Dir Quit compiler Options

Text: 0 bytes

Free: 28909 bytes

Wykonanie określonej komendy następuje po wciśnięciu odpowiedniego klawisza (W,M,E,C,R,S,X,D,Q,O). Po wykonaniu komendy ponowne wyświetlenie menu głównego można uzyskać przez wciśnięcie klawisza **RETURN**.

Wyjście z systemu TURBO PASCAL umożliwia komenda Quit. Po wciśnięciu klawisza Q następuje powrót do systemu operacyjnego. Aby wrócić do systemu TURBO PASCAL, należy ponownie wykonać komendę **TURBO**.

2. Jak uruchomić pierwszy program

Po napisaniu programu należy wywołać edytor (komenda Edit), używając komend edytora wprowadzić program źródłowy do zbioru roboczego edytora, skompilować go i uruchomić (komenda Run). Ciąg czynności prowadzących do uruchomienia programu można przedstawić w postaci następującego algorytmu:

- wcisnąć klawisz **E** (wywołanie edytora),
- podać nazwę zbioru roboczego edytora i wcisnąć klawisz **RETURN**,
- wprowadzić z klawiatury wiersz tekstu programu źródłowego (zostanie on wyświetlony na ekranie monitora),
- wcisnąć klawisz **RETURN**,
- czynności c i d wykonać dla kolejnych wierszy tekstu programu źródłowego,
- po wprowadzeniu ostatniego wiersza tekstu programu źródłowego wcisnąć jednocześnie klawisze **CONTROL** i **K**, a następnie wcisnąć klawisz **D** (wykonanie komendy edytora **Koniec Edycji**),
- wcisnąć klawisz **R** (kompilacja i uruchomienie programu). Po wciśnięciu klawisza **E** nastąpi wyświetlenie komunikatu:

Work file name:

po którym należy podać nazwę zbioru w postaci:

nazwa lub nazwa. typ

gdzie:

nazwa — ciąg od jednego do ośmiu znaków, określający nazwę zbioru,

typ — ciąg od jednego do trzech znaków, określający typ zbioru

(pominięcie typu jest równoważne przyjęciu typu domyślnego **PAS**).

W obu wypadkach żaden ze znaków nie powinien być znakiem:

= ? * . , < > : [] { }

Po podaniu nazwy zbioru (np. **FILENAME.PAS**) i wciśnięciu klawisza RETURN w pierwszym wierszu, u góry ekranu, zostanie wyświetlony opis stanu edytora w postaci:

Line 1 Col 1 Insert Indent A:FILENAME.PAS

(od tego momentu edytor reaguje tylko na własne komendy, a nie na komendy systemu TURBO PASCAL). W kolejnych wierszach będą wyświetlane wprowadzane wiersze tekstu programu źródłowego. Gdy cały ekran zostanie wypełniony, wprowadzenie nowego wiersza następuje po przesunięciu zawartości ekranu (bez opisu stanu edytora) o jeden wiersz w górę. Nie należy się jednak obawiać utraty „znikających” wierszy. Można je bowiem wyświetlić, wykorzystując odpowiednie komendy edytora.

Wciśnięcie klawisza **R** (po zakończeniu edycji) spowoduje kompilację programu i, jeśli zakończy się ona poprawnie, jego uruchomienie. W razie błędnej kompilacji należy poprawić program źródłowy i ponownie wykonać komendę Run. Poprawienie programu źródłowego dokonuje się za pomocą odpowiednich komend edytora. Opis tych komend oraz sposób ich użycia przedstawiamy w następnej części kursu.

3. Proste programy w języku PASCAL

Każdy program zapisywany w języku PASCAL składa się z trzech części:

- nagłówek (początku programu),
 - bloku (utworzonego z części definiującej i części operacyjnej),
 - separatora "." kończącego program,
- które będziemy stopniowo definiowali, odwołując się do intuicyjnie wykazanych potrzeb.

Zacznijmy od najprostszego wariantu — wyprowadzenia na monitor ekranowy rezultatów wykonania programu, niezbędnego w większości programów. Budowę takiego programu ilustruje przykład 1.

Przykład 1

```
PROGRAM prog1;  
BEGIN  
    WRITE ('pierwszy program w języku pascal')  
END.
```

Nagłówek programu składa się z następujących elementów:

- tzw. słowa kluczowego **PROGRAM**,
- nazwy programu, będącej kombinacją liter i cyfr rozpoczynającą się od litery (prog 1),
- separatora "." oddzielającego nagłówek od bloku programu.

Blok programu, w tym prostym wypadku, zawiera tylko część operacyjną, w której pomiędzy słowami kluczowymi **BEGIN — END** zapisano jedną instrukcję wyjścia **WRITE**. Jej zadaniem jest tutaj wyprowadzenie na ekran monitora tekstu ujętego w apostrofach.

W wyniku programu "program 1" na ekranie monitora został wyświetlony tekst o postaci:

pierwszy program w języku pascal

W ogólnym wypadku cały proces obliczania przez komputer składa się z trzech faz ⁴:

- wprowadzenia danych wejściowych,
- przetworzenia danych wejściowych,
- wyprowadzenia wyników.

Każda z tych faz występuje w programie "prog 2" przedstawionym w przykładzie liczb całkowitych

Przykład 2

```
PROGRAM prog 2;  
VAR x, y, suma, różnica, iloczyn: integer;  
    iloraz: real; (deklaracje zmiennych)  
BEGIN  
    READ (x,y);
```

```
WRITELN;
```

```
WRITELN ('pierwsza liczba = ' x:2, 'druga liczba =  
        y:2);
```

```
suma: = x+y;
```

```
różnica: =x-y;
```

```
iloczyn: =x*y;
```

```
iloraz: =x/y;
```

```
WRITELN (' suma = ', suma: 3, ' różnica = ' różnica:2,  
        iloczyn = ', iloczyn:4, ' iloraz = ' iloraz:  
        8:5)
```

```
END
```

oraz wyznaczenie wyprowadzenie na monitor ekranowy ich sumy, różnicy, iloczynu i ilorazu. Zauważmy, że w części operacyjnej bloku programu występuje pięć zmiennych: X, Y, SUMA, RÓŻNICA, ILOCZYN, które będą przyjmowały wartości całkowite oraz zmienna ILORAZ, która będzie przyjmowała wartości rzeczywiste. Wykorzystanie zmiennych (symbolicznie reprezentujących obiekty rzeczywiste, w tym wypadku liczby (pozwala na wykorzystanie programu dla różnych danych wejściowych bez konieczności wprowadzania jakichkolwiek zmian w zapisie programu. W danym momencie realizacji programu zmienne reprezentują tylko jedną wartość, zwaną bieżącą wartością zmiennej. Wartość zmiennej może być zmieniona na inną jedynie w wyniku wykonania instrukcji przypisania (w której jako lewy argument operatora "=" występuje nazwa zmiennej) lub instrukcji wejścia (w której nazwa zmiennej występuje jako jeden z argumentów).

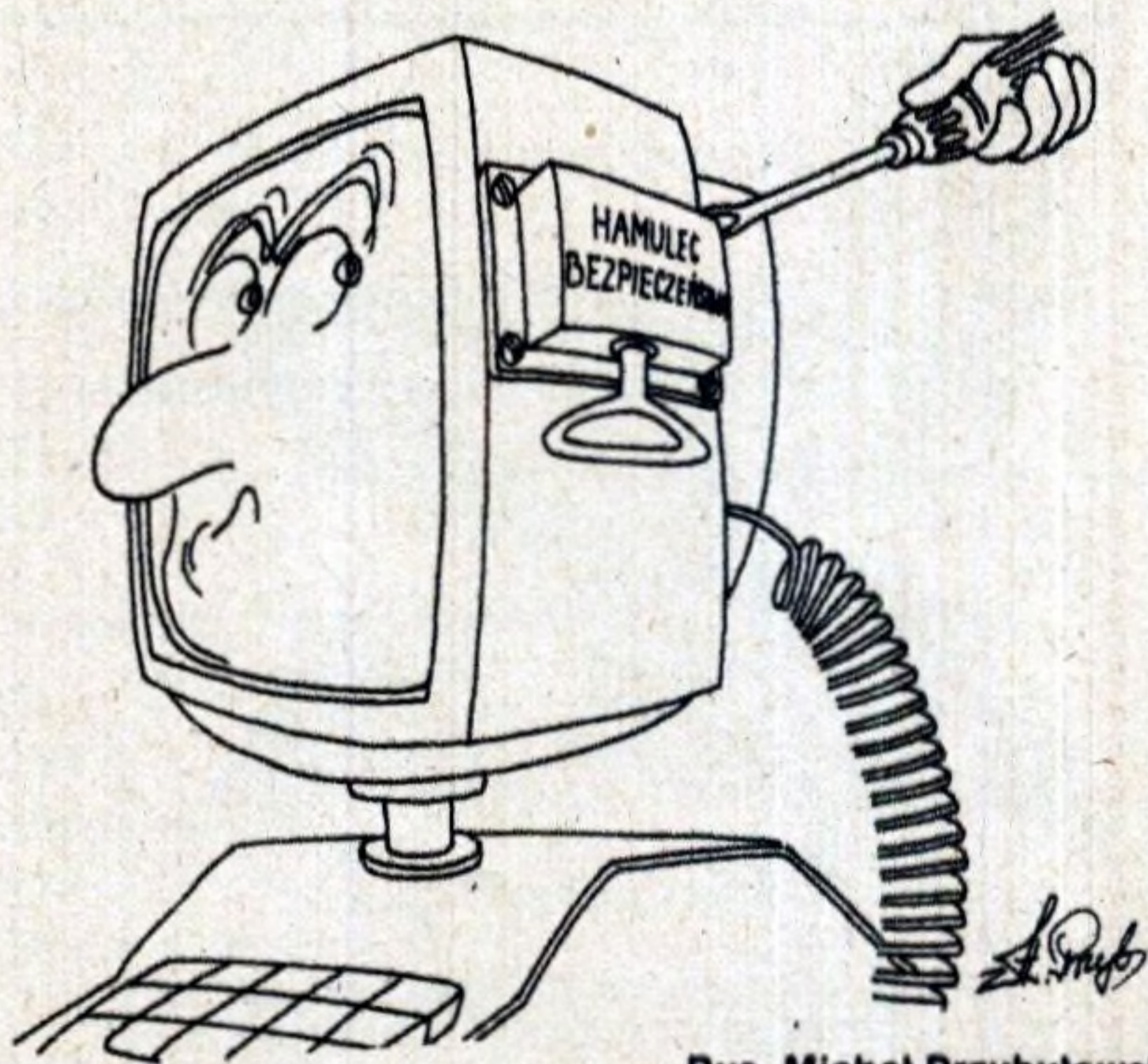
Opisy zmiennych programu występują w części definiującej bloku programu w formie tzw. deklaracji zmiennych.

Deklaracja zmiennych składa się ze słowa kluczowego VAR i zapisów deklaracji zmiennych. Każdy zapis deklaracji zmiennych składa się z następujących elementów:

- listy nazw deklarowanych zmiennych danego typu (np. liczby całkowite) oddzielonych od siebie przecinkami,
- znacznika ":" oddzielającego listę zmiennych od nazwy typu,
- nazwy typu (klasy wartości, przyjmowanej przez deklarowane zmienne, np. integer, dla liczb całkowitych, real — dla liczb rzeczywistych),
- separatora ";" oddzielającego zapis danej deklaracji zmiennej od innych zapisów.

Występująca w części operacyjnej instrukcja wejścia READ wykazuje pierwszą z wymienionych faz — wprowadzenia danych wejściowych. Zgodnie z tą instrukcją następuje czytanie liczb, na których będą wykonywane żądane operacje. Wartości wczytanych liczb są przypisywane zmiennym X i Y.

dokończenie na str. 24



Rys. Michał Przybyłowski

Fazę drugą uruchamiają cztery instrukcje przypisania, w których kolejno wyznacza się sumę, różnicę, iloczyn i iloraz wczytanych liczb. Każda z tych instrukcji składa się z nazwy zmiennej, operatora przypisania "=" i wyrażenia (w tych wypadkach arytmetycznego). Występujące w wyrażeniach operatory +, -, *, / oznaczają odpowiednio dodawanie, odejmowanie, mnożenie, dzielenie. Należy zauważyć, że wartość każdego wyrażenia ma typ zgodny z typem zmiennej, do której jest przypisywana.

Do wyprowadzenia wyników na ekran monitora (realizacja fazy trzeciej) wykorzystano instrukcję WRITELN. Instrukcja ta, podana bez parametrów, umożliwia przejście do nowej linii. Wykonanie instrukcji WRITELN z parametrami jest podobne do wykonania instrukcji WRITE, przy czym po wyprowadzeniu wartości parametrów następuje przejście do nowej linii. Parametrami instrukcji WRITELN są tutaj, oprócz tekstów, nazwy zmiennych, których wartości mają być wyprowadzane. Określona jest przy tym liczba pozycji znakowych, jakie ma zajmować wyprowadzana wartość. Dla zmiennej typu real określono ponadto liczbę pozycji znakowych przeznaczonych na część ułamkową (w tym przypadku 5). Rozplanowując rozmieszczenie wyników na ekranie monitora założono, że wprowadzane liczby będą liczbami z zakresu (1,99). Dla liczb 27 i 13 uzyskano następujący wynik działania programu:

pierwsza liczba = 27 druga liczba = 13

suma = 40 różnica = 14 iloczyn = 351 iloraz = 2.07692

Liczby te zostały wprowadzone z klawiatury (wpisanie liczb 27 13 i naciśnięcie klawisza RETURN lub ENTER).

W przykładzie 2 zamieszczono tekst objaśniający, ujęty w nawiasy klamrowe. Jest to komentarz nie mający żadnego wpływu na przebieg obliczeń (jest pomijany przez kompilator). Komentarze będą używane również w dalszych programach w celu zwiększenia czytelności programów i ułatwienia zrozumienia ich treści.

Dotychczas poznane instrukcje umożliwiają już wykonanie prostych obliczeń na wprowadzonych danych oraz prezentację wyników. Przeważnie jest to niewystarczające, gdyż wykonanie takich działań często zależy od spełnienia określonych warunków. Zagadnienie to zilustrujemy bliżej na przykładzie wyznaczania pierwiastków rzeczywistych trójmianu kwadratowego (przykład 3). Jak wiemy, liczba pierwiastków rzeczywistych zależy od wartości

Przykład 3

PROGRAM prog 3;

VAR a, b, c, (współczynniki trójmianu)

x1, x2 (pierwiastki trójmianu)

delta:real; (zmienna pomocnicza)

BEGIN

READLN (a);

IF a <= 0 THEN

WRITELN (' niedopuszczalna wartość współczynnika a')

ELSE

BEGIN

READ (b, c);

WRITELN;

WRITELN (' dla współczynników: a = ', a);

WRITELN (' b = ', b);

WRITELN (' c = ', c);

delta := b*b-4*a*c; (obliczenie wyróżnika trójmianu)

IF delta < 0 THEN

WRITELN (' brak pierwiastków rzeczywistych')

ELSE

IF delta = 0 THEN

BEGIN

x1 := -b/(2*a);

WRITELN (' trójmian ma jeden pierwiastek podwójny x = ', x1)

END

ELSE

BEGIN

x1 := (-b-SQRT(delta))/(2*a);

x2 := (-b+SQRT(delta))/(2*a);

WRITELN;

WRITELN (' trójmian ma dwa pierwiastki: x1 = ', x1);

WRITELN (' x2 = ', x2)

END

END

END.

wyróżnika trójmianu. Zatem w zależności od jego wartości należy wykonać inną grupę instrukcji. Wybór umożliwia instrukcja alternatywy o postaci:

IF warunek THEN instrukcja 1 ELSE instrukcja 2

Wykonanie instrukcji alternatywy polega na zbadaniu warunku i jeśli jest on spełniony, następuje wykonanie instrukcji 1. W przeciwnym razie wykonywana jest instrukcja 2. W warunkach z przykładu 3 zastosowano operatory relacyjne: < (mniejszy), <= (mniejszy lub równy), = (równy). Pozostałe operatory relacyjne to: > (większy), >= (większy lub równy), <> (nierówny). W celu umożliwienia wykonania określonej grupy instrukcji w wypadku spełnienia (niepełnienia) warunku wykorzystano instrukcję złożoną. Instrukcja złożona jest sekwencją instrukcji zawartą między parą słów kluczowych BEGIN — END. Jest ona traktowana jako pojedyncza instrukcja. Do wyznaczania wartości pierwiastka wyróżnika trójmianu użyto standardowej funkcji SQRT. Odpowiednią kolejność wykonywania działań w wyrażeniach arytmetycznych zapewniają nawiasy (podobnie, jak w powszechnie stosowanej notacji matematycznej). Instrukcję READLN, wykorzystaną do wczytania współczynnika "a", wykonuje się jak instrukcję READ, przy czym po wczytaniu danej wejściowej następuje przeskok do następnej linii wczytania (w tym przypadku do następnego wiersza ekranu). Wartości wyznaczonych pierwiastków (jeśli istnieją) wyprowadzane są w standardowym formacie zmiennoprzecinkowym, przyjętym dla liczb rzeczywistych. Ma on postać:

□ □ #.#####E * ##

dla liczb większych lub równych zero oraz postać:

□ - #.#####E * ##

dla liczb mniejszych od zera, przy czym □ reprezentuje spację, # reprezentuje cyfrę dziesiętną, * reprezentuje znak + lub —.

Wartość tak wyprowadzonej liczby należy odczytywać jako

$c \cdot 10^m$

gdzie:

c — liczba znajdująca się przed literą E (cecha),

m — liczba znajdująca się za literą E (mantysa).

Program przedstawiony w przykładzie 3 wykonano dla następujących wartości współczynników trójmianu $a = 2,67$, $b = 13,63$, $c = 7,21$ i uzyskano następujące wyniki:

dla współczynników:

$a = 2.6700000000E+00$

$b = 1.3630000000E+01$

$c = 7.2100000000E+00$

trójmian ma dwa pierwiastki: $x1 = -4.5055210206E+00$

$x2 = -5.9934789329E-01$

Dane zostały wprowadzone z klawiatury w następujący sposób:

— wpisanie liczby 2.67 i wciśnięcie klawisza RETURN,

— wpisanie liczb 13.63 i 7.21 i wciśnięcie klawisza RETURN.

Zamiast klawisza RETURN można używać również klawisza ENTRY.

Wprowadzona instrukcja alternatywy może być wykorzystywana do opisu działań o wybiornym przebiegu. Nie pozwala natomiast opisać operacji wykonywanych powtarzalnie. Jedną z instrukcji dostępnych w języku PASCAL, umożliwiających taki opis, jest instrukcja WHILE o postaci:

WHILE wyrlog Do
instr

gdzie:

wyrlog — wyrażenie logiczne przyjmujące wartość TRUE(prawda) lub FALSE (fałsz),

instr — pojedyncza instrukcja języka PASCAL.

Powyższy zapis oznacza, że wykonanie instrukcji „instr” ma być powtarzalne tak długo, aż wartość wyrażenia logicznego „wyrlog” zostanie doprowadzona do wartości FALSE. Zauważmy, że jeżeli już przed wykonaniem instrukcji

WHILE wartością „wyrlog” jest FALSE, to instrukcja „instr” nie jest w ogóle wykonywana. Instrukcję WHILE wykorzystano w programie wyznaczania największego wspólnego dzielnika dwóch całkowitych liczb dodatnich (przykład 4).

Przykład 4.

```
PROGRAM prog 4;
VAR x,y: integer;
BEGIN
  READ (x, y);
  WRITELN;
  WRITE (' największy wspólny dzielnik liczb
        'x,' i 'y,' wynosi ');
  WHILE x<>y DO
  IF x > y THEN
  x := x - y
  ELSE
  y := y - x;
  WRITELN (x)
END
```

Nietrudno zauważyć, że wpływ na wartość wyrażenia logicznego w instrukcji WHILE ($x \diamond y$) ma wykonanie instrukcji alternatywy, a więc instrukcji „instr”. Należy przy tym zdawać sobie sprawę z tego, że jeżeli po żadnym z powtórzeń instrukcji „instr” wyrażenie logiczne nie osiągnie wartości FALSE, to wykonywanie instrukcji WHILE nigdy się nie zakończy. Mówi się wtedy często o „zapętleniu” programu.

Wyrażenia występujące w przykładzie 4 zawierają nazwy tych samych zmiennych po obu stronach operatora „:=”, np.

$x := x - y.$

W takim wypadku przy obliczaniu wartości wyrażenia brana jest pod uwagę wartość zmiennej x przed wykonaniem instrukcji podstawienia. Wyznaczona wartość wyrażenia jest następnie przypisywana zmiennej x jako nowa wartość (a więc poprzednia wartość zostaje zniszczona).

Program wykonano dla liczb 256 i 64. Uzyskano następujące wyniki:

Największy wspólny dzielnik liczb 256 i 64 wynosi 64
Ponieważ w instrukcjach wyjścia nie określono liczby pozycji znakowych dla wyprowadzonych liczb całkowitych, zajmują one tyle pozycji, ile mają cyfr znaczących.

Kolejne wiadomości o języku PASCAL — w następnej części.

4. Zadania

Wychodząc z założenia, że programowania nie można nauczyć się bez intensywnego i samodzielnego opracowywania programów, ich kompilacji, uruchamiania i testowania, w każdej części, będziemy przedstawiali zadania do samodzielnego opracowania. Zestawy zadań postaramy się tak dobrać, aby ich opracowanie było możliwe na podstawie już przedstawionych wiadomości o języku. Zachęcamy Czytelnika do rozwiązania chociażby jednego z zadań.

Zadanie 1

Napisać program wyznaczający wartość wyrażenia:

$$a) \frac{x^2 - ax^3}{2ax};$$

$$b) \frac{(a \cdot b)^2 + \frac{c}{d}}{a - (c - \frac{d}{b})^2}$$

Zadanie 2

Napisać program, który sprawdza czy trzy całkowite liczby dodatnie a, b, c, są długościami boków trójkąta i jakiego (równoramiennego, równobocznego, prostokątnego).

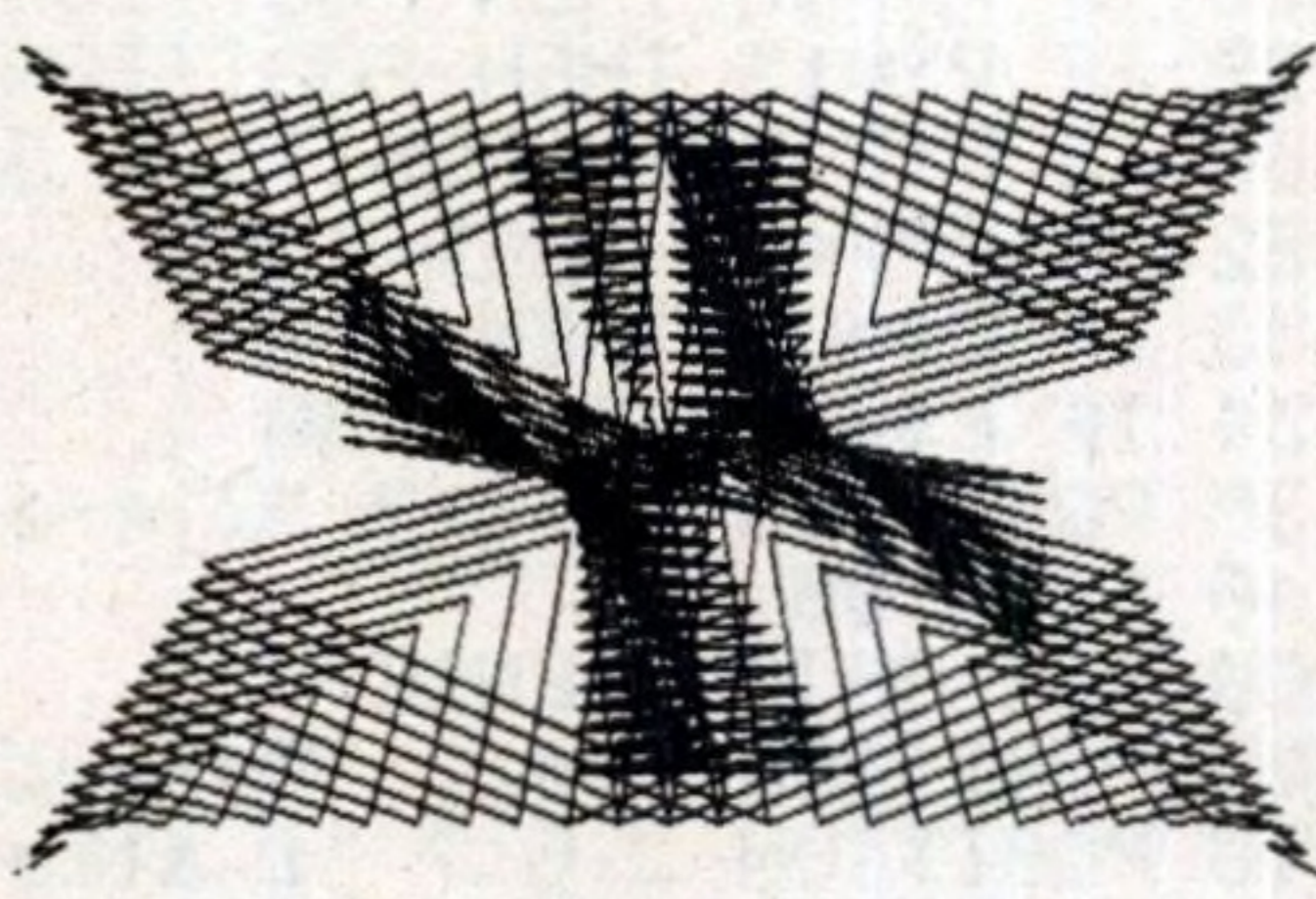
Zadanie 3

Napisać program obliczania wartości szeregu

$$1 + \frac{x^2}{2} + \frac{x^4}{2 \cdot 4} + \frac{x^6}{2 \cdot 4 \cdot 6} + \dots$$

- 1 Implementację języka uzyskuje się przez opracowanie translatora, tj. programu tłumaczącego teksty z danego języka (źródłowego) na inny język (wynikowy). Translatory dzielimy na kompilatory i interpretery. System TURBO PASCAL wyposażony jest w kompilator języka PASCAL.
- 2 Edytor ekranowy pozwala na bezpośrednie działanie na tekście aktualnie wyświetlanym na ekranie monitora. Za pomocą odpowiednich komend tekst taki można rozsuwać, wstawiać nowe słowa, kasować określone słowa lub wiersze tekstu itp.
- 3 Słowa kluczowe języka są słowami o określonym znaczeniu i mogą być używane tylko zgodnie z regułami dopuszczanymi przez język. W przykładach programów słowa kluczowe są napisane, dla odróżnienia, dużymi literami.
- 4 Jest to podział logiczny. W programach, części programów zaliczane umownie do poszczególnych faz z reguły się przeplatają.

S.R.

<pre>10 CLS 20 ORIGIN 320,200 30 FOR x=1 TO 10 35 FOR y=1 TO 15; 40 DRAW -7*x,10*y 45 DRAW -5*y,10*y 47 NEXT y 48 ORIGIN 390,200 50 DRAW 5*y,-10*x 60 NEXT x 70 ORIGIN 320,200 80 FOR k=1 TO 10 90 FOR m=1 TO 15 100 DRAW 7*k,-10*m</pre>	<pre>110 DRAW 5*m,-10*k 120 NEXT m 125 ORIGIN 270,200 130 DRAW -5*m,10*k 135 NEXT k 137 ORIGIN 580,400 140 FOR x=1 TO 15 143 DRAW -20*x,-15*x 145 DRAW -5*x,-10*x 147 DRAW -20*x,-25 155 NEXT x 160 ORIGIN 580,0 170 FOR x=1 TO 15 180 DRAW -20*x,15*x</pre>	<pre>185 DRAW -5*x,10*x 190 DRAW -20*x,25 200 NEXT x 210 ORIGIN 60,0 220 FOR x=1 TO 15 230 DRAW 20*x,15*x 240 DRAW 5*x,10*x 250 DRAW 20*x,25 260 NEXT x 270 ORIGIN 60,400 280 FOR x=1 TO 15 290 DRAW 20*x,-15*x 300 DRAW 5*x,-10*x 310 DRAW 20*x,-25 320 NEXT x</pre>	 <p style="text-align: center;"><i>Grafika komputerowa (Amstrad)</i></p>
---	--	---	---

„Węglowodory”, to prosty program edukacyjny, przeznaczony do sprawdzenia wiadomości o węglowodorach od C_1H_4 do C_8H_{16} . Program może być użyty jako sprawdzian wiedzy lub do zobrazowania wzorów strukturalnych. Przy niewielkim nakładzie pracy możliwe jest rozbudowanie programu o inne grupy związków organicznych.

```

5 REM ** WEGLOWODORY **
7 OPEN #1,4,0,"K:"
10 DIM A$(90),B$(7),C$(75),D$(6)
12 DIM E$(6),F$(40),Q$(1)
14 F$=""
16 Q$=""
20 GOSUB 1000
23 ? " ) " : POKE 752,1
25 POSITION 14,2: ? "[ WEGLOWODORY ]"
30 POSITION 6,8: ? "Czy potrzebujesz wyjasnienia"
35 POSITION 16,10: ? "[T]ak/[N]ie"
45 GET #1,X
50 IF X=84 THEN GOTO 200
55 IF X<>84 THEN GOTO 65
65 P=0:PP=0
80 I=INT(RND(0)*15):J=I*6+1:K=I*5+1
82 B#=A$(J,J+5)
83 D#=C$(K,K+4)
85 IF PP=0 THEN 92
90 FOR F=1 TO 500:NEXT F
92 GOSUB 1200
99 POSITION 14,2: ? "[ WEGLOWODORY ]"
101 POSITION 1,20: ? "[ Wzory strukturalne           W           ]"
102 POSITION 1,21: ? "[ Prawidlowe odpowiedzi           Ocena       ]"
103 POSITION 1,22: ? "[ Uruchomienie T Kasowanie N           ]"
105 POSITION 27,21: ? P
107 POSITION 7,7: ? B$
110 POSITION 7,9: INPUT E$
111 PP=PP+1
112 IF E$="W" THEN GOSUB 400
114 IF E$="T" THEN 23
116 IF E$="N" THEN 1300
120 L=LEN(E$)
122 IF L=4 THEN E$(LEN(E$)+1)=Q$
126 IF PP<16 THEN 133
127 IF P>=7 THEN POSITION 36,21: ? "3"
128 IF P>=10 THEN POSITION 36,21: ? "4"
129 IF P>=13 THEN POSITION 36,21: ? "5"
130 IF P<7 THEN POSITION 36,21: ? "2"
132 IF PP>=16 THEN 1100
133 IF E#=D$ THEN 300
134 IF E$="W" THEN 80
135 POSITION 5,16: ? "Zle-sprobuj jeszcze raz"
140 GOTO 90
200 REM OBJASNIENIE
205 POSITION 2,4: ? "Sprawdzenie wiadomosci o alkanach"
210 POSITION 2,6: ? "i alkenach."
215 POSITION 2,8: ? "Gdy pokaze sie nazwa wewlowodoru"
220 POSITION 2,10: ? "podaj jego wzor sumaryczny."
225 POSITION 2,12: ? "Naciśnij klawisz W i RETURN otrzy-"

```

```

230 POSITION 2,14: ? "mujesz wzor strukturalny."
235 POSITION 2,16: ? "Klawisz T i RETURN uruchamiasz"
240 POSITION 2,18: ? "program od poczatku."
245 POSITION 2,20: ? "Klawisz T i RETURN kasuje program."
250 POSITION 2,22: ? "[ Wcisnij dowolny klawisz ]"
260 GET #1,X
265 ? ">"
270 IF X<>27 THEN 65
300 POSITION 5,16: ? "Wynik dobry - kontynuuj"
310 P=P+1: POSITION 27,21: ? P
320 GOTO 80
400 REM WZORY STRUKTURALNE.
402 L=0
405 IF I>7 THEN 450
407 I=I+1
410 POSITION 20,8: ? "H": POSITION 20,9: ? "I": POSITION 18,10: ? "H-C-H": POSITION 20
,11: ? "I": POSITION 20,12: ? "H"
415 IF I=1 THEN 495
425 FOR A=2 TO I: L=L+2
430 POSITION 20+L,8: ? "H": POSITION 20+L,9: ? "I": POSITION 20+L,10: ? "C-H": POSITIO
N 20+L,11: ? "I": POSITION 20+L,12: ? "H"
435 NEXT A
440 GOTO 495
450 POSITION 20,8: ? "H H": POSITION 20,9: ? "I I": POSITION 20,10: ? "C=C": POSITION
20,11: ? "I I": POSITION 20,12: ? "H H"
455 IF I=8 THEN 495
460 POSITION 22,8: ? " ": POSITION 22,9: ? " "
465 FOR A=9 TO I: L=L+2
470 POSITION 22+L,8: ? "H": POSITION 22+L,9: ? "I": POSITION 21+L,10: ? "-C-H": POSITI
ON 22+L,11: ? "I": POSITION 22+L,12: ? "H"
475 NEXT A
495 FOR F=1 TO 700: NEXT F
500 RETURN
1000 A$="METAN ETAN PROPANBUTAN PENTANHEKSANHEPTANOKTAN ETEN PROPENBUTEN PENTE
NHEKSENHEPTENOKTEN "
1010 C$="C1H4 C2H6 C3H8 C4H10C5H12C6H14C7H16C8H18C2H4 C3H6 C4H8 C5H10C6H12C7H14C
8H16"
1020 RETURN
1100 GOSUB 1200
1110 POSITION 5,16: ? "Czy chcesz sprobowac jeszcze raz"
1120 POSITION 16,18: ? "[T]ak/[N]ie"
1130 GOTO 110
1200 FOR F=4 TO 19
1210 POSITION 0,F: ? F$
1220 NEXT F
1230 RETURN
1300 ? ">": POKE 752,0: NEW

```

UWAGA !

Sposob wprowadzania nietylchorych znakow niedostepnych w druku :

[T] - REVERSE VIDEO a potem T
) - ESC a nastepnie CONTROL i CLEAR



Po długim oczekiwaniu rynek oprogramowania mikrokomputerów wzbogacił się o systemy „interfejsu graficznego”. Sprawiają one, iż komputer wydaje się bardziej „życzliwy”.

Firma DIGITAL RESEARCH była pierwszą, która dostarczyła użytkową wersję systemu graficznego — GEM. Znakomicie ułatwia on wykonywanie żmudnych i kłopotliwych czynności związanych z przekazywaniem komunikatów do systemu operacyjnego mikrokomputera, a także ułatwia odczyt informacji adresowanych przez maszynę do użytkownika.

Praktycznie, w dużej części służy temu tak zwana myszka. Jest to urządzenie w postaci niewielkiego (jak paczka papierosów) pudełka z klawiszami (najczęściej od jednego do trzech klawiszy). W dolnej płaszczyźnie pudełka znajduje się kulka, która, podobnie do zainstalowanej w końcówce długopisu, przesuwana po dowolnej płaszczyźnie, obraca się. Każdy ruch (obrót) kulki, dzięki odpowiednim układom elektronicznym myszki, sygnalizowany jest komputerowi (za pośrednictwem kabla — ogonka myszki).

Przesuwanie myszki, np. po stole, odzwierciedlane jest na ekranie monitora zmianą położenia charakterystycznego punktu, tak zwanego znacznika.

Zatem urządzenie to pozwala między innymi na wskazywanie jednego z wielu wyświetlanych na ekranie komunikatów, np. składową menu — „wyczyść ekran”. Gdy znacznik znajdzie się w polu wyświetlanego komunikatu, należy po prostu nacisnąć określony klawisz myszki. Jest to jednoznaczne z napisaniem wyróżnionych słów za pośrednictwem klawiatury.

OKNA (produkt firmy MICROSOFT) także zapewniają realizację tego typu zadań. Ponadto pozwalają na pseudowieloprogramowość. Oznacza to, że stwarzane jest wrażenie jednoczesnej realizacji więcej niż jednego użytkowego programu.

Podstawową ideą tej formy graficznej komunikacji z mikrokomputerem jest uproszczenie czynności manualnych związanych z kierowaniem pracą systemu, a także poprawienie jakości postaci informacji przekazywanej przez mikro.

Na przykład, jeśli przetwarzane są jednocześnie dwa programy, dwie wyróżnione części ekranu, umownie nazywane **oknami**, prezentują obrazy odzwierciedlające aktualny stan procesu przetwarzania informacji przez te programy. Monitor służy do „podglądania” pracy mikro.

Podobnie jak ekrany monitorów w supersamie służą do „podglądania” wielu miejsc sklepu jednocześnie przez

jedną osobę. W przypadku sklepu źródłem informacji są kamery. W mikro rolę tę pełni np. system OKNA.

Myszka jest wykorzystywana do szeregu różnych zadań, jednym z nich jest wskazywanie wybranej czynności, którą ma wykonać mikro, na przykład: nowy rozmiar okna, otworenie nowego okna.

Czynności związane z instalowaniem systemu OKNA w nowym komputerze nie są zbyt złożone. Wymagane jest tylko wspomniane manipulowanie myszką. Polega ono na wskazywaniu wyróżnionych elementów menu, które jest wyświetlane na ekranie w jednym z okien.

Zazwyczaj w pierwszej kolejności dokonuje się tym sposobem odpowiednie przystosowanie dysku do pracy pod kontrolą okien. Następnie nale-

OKNA

ży przekazać do systemu informacje o konfiguracji sprzętowej (rodzaj drukarki, wielkość i typ pamięci itp.).

Sterowanie systemem OKNA jest tak proste, że najczęściej podręcznik okazuje się zbędny. Jednak mając na uwadze małe doświadczenie użytkowników końcówek mikro, firma MICROSOFT zamierza jednak sprzedawać odpowiednią dokumentację.

OKNA wykorzystują tę samą metodę prezentacji co konkurencyjne systemy: GEM i MACINTOSH, jednak nie dysponują możliwością polegającą na wyświetlaniu komend sterowania systemem oraz zbiorów w postaci IKON.

Jednym z podstawowych założeń współczesnych metod porozumiewania się obrazem, jest maksymalne ograniczenie sztucznych, wyuczonych symboli, jakimi są między innymi litery i cyfry. Lansowany jest powrót do naturalnego języka, jakim wydają się obrazy, w swej uproszczonej formie (przy zachowaniu meritum treści) są nimi, według umownej terminologii, ikony. Najczęściej ilustrują przedmiot lub czynność. Przykładem takiej konwencji ikon są niektóre oznaczenia w miejscach publicznych, a także część znaków drogowych.

W odniesieniu do techniki mikrokomputerowej prace w tym kierunku prowadzone są od dosyć dawna i ich wynikiem jest między innymi SNOBOI 4 — język ikon.

Ikony w sposób istotny usprawniły i ułatwiły proces porozumiewania się człowieka z komputerem i odwrotnie. Często zamiast mało czytelnych słów niewielka rycina ukazana na ekranie monitora dostarcza nam szybko i bezbłędnie określoną treść.

Obecnie ikony dostępne są we

wszystkich profesjonalnych mikrokomputerach. Pokonują bariery językowe i są najbliższe naturalnej tendencji porozumiewania się. Redukują czas uczenia się, jest mniej kłopotów podczas pracy z mikro, staje się ona atrakcyjna. Doskonałą ilustracją tej metody jest większość gier komputerowych, które dzięki ikonom są zrozumiałe w czasie kilku chwil, nawet dla dzieci. Okazuje się bowiem, że język jakim posługuje się autor gry nie ma znaczenia. W zasadzie bawiący się nie muszą umieć czytać. Klasycznym przykładem takiej gry jest SHADOWFIRE (zabawa, której twórcą jest Danton Designs z firmy BEYOND).

Kiedy zamierzamy sprawdzić zawartość dysku — OKNA wyświetlają aktualną listę zbiorów w postaci tekstu. Natomiast GEM i MACINTOSH oferują tekst i ikony.

Wydaje się, że jednak OKNA są systemem lepszym od swych konkurentów. Górują między innymi szybkością odpowiedzi (na polecenia wydawane za pomocą myszki), łatwiejsze jest sterowanie pracą mikrokomputera i użytkuje się zdecydowanie bardziej czytelną formę prezentacji menu.

Szybkie otwieranie różnych okien, „podglądanie” różnych newralgicznych punktów realizowanego programu, np.: procesu kontroli danych wejściowych, postaci danych zapisywanych na dysku, pośrednich wyników przetwarzania, następuje praktycznie bezwzględnie, ale pod warunkiem, że przetwarzany jest tylko jeden program. Przyczyna tkwi w szybkości procesora, który jest udostępniany przez system kolejno realizowanym programom. Wyświetlane okna są „zdjęciami” zrobionymi podczas ostatniego kontaktu programu z procesorem. Stąd okres między kolejnymi zdjęciami (oknami) tego samego programu rośnie wraz z liczbą przetwarzanych programów. Gdy pracuje jeden program, okres ten jest niezauważalny.

Niestety OKNA są w pełni wykorzystane tylko wówczas, gdy wspomagają pracę oryginalnych programów MICROSOFTa. Dołączany jest wówczas tak zwany pamiętnik programu użytkowego, notatnik dla użytkownika, kalkulator i zegar. Wszystkie te zdolności są odpowiednio prezentowane na ekranie monitora w oknach, np. zegar w postaci tarczy i tradycyjnych wskazówek.

System ten posiada także prostą bazę danych, rejestrowane są one w postaci obrazów i tekstów.

Przykładem firmowego oprogramowania w pełni wykorzystującego możliwości OKIEN jest PAINT. Pozwala on tworzyć obrazy o wyszukanej formie i dowolnej treści. Stanowi konkurencję dla uznanego przez profesjonalistów

dokończenie na str. 30



Przyjechali... aż zaprzęgiem. Oprócz wielu atrakcji, przywieźli również komputery. A najwięcej radości z „Dni Berlina” w Warszawie miały dzieci.

Foto: Ryszard Rogoń



CHIŃSKIE KOMPUTERY PERSONALNE

Chińska Republika Ludowa oferuje mikrokomputery oznaczone 0520A, 0520 C-E, 0520 C-H kompatybilne z IBM PC. Model 0520 C-H ma mikroprocesor 8088, 512 KB pamięci operacyjnej RAM, dwa mechanizmy dysków elastycznych o gęstości zapisu 360 KB oraz dysk stały 20 MB.

W monitorze kolorowym ekran przy modzie graficznym pracuje z 648 x 504 punktami bezpośrednio adresowanymi, w monitorze monochromatycznym z 972 x 700 punktami bezpośrednio adresowanymi.

ARM

Mikroprocesor ARM, czyli Acorn RISC (Reduced Instruction Set Computer) Machine, o architekturze z przetwarzaniem potokowym (pipelining), może wykonać 3 miliony rozkazów w ciągu sekundy. Zbudowany jest z 25 000 tranzystorów, umieszczonych na powierzchni 7 mm², a wykonanych technologią CMOS. ARM ma 25 pełnych rejestrów 32-bitowych. Mikroprocesor dysponuje 26-bitową magistralą adresową, czyli może adresować przestrzeń 64 MB a dokładniej — adresuje 67 108 864 słowa 32-bitowe. Wyposażony jest w 32-bitową magistralę danych.

Dyski laserowe

Firma AGA Inc. wprowadza system dysków laserowych dla komputerów personalnych IBM. Dyski o średnicy 12 cali mają umożliwić przy zapisie danych tekstowych osiągnięcie pojemności 3 GB, przy zapisie obrazów graficznych — pojemność 50 GB.

J. W.

ZAKŁADY ELEKTRONICZNE „MICRONET”



ODDZIAŁ Gdański
ul. Krasickiego 9
81-836 Sopot
tel. 51-13-17, 41-32-25 w. 36, 65
tlx 051-2299

ODDZIAŁ BYDGOSKI
ul. Sobieszewska 17
85-717 Bydgoszcz
tel. 42-29-32

oferują do sprzedaży:

TERMINAL EKRANOWY SERII AN-2000

- 24 wiersze po 80 znaków
- duże i małe litery, alfabet polski
- interfejs szeregowy V-24

Wykorzystując ww. terminal instalujemy również wielodostępne systemy na mikrokomputery typu IBM PC

OFERUJEMY PONADTO WSPÓŁPRACĘ W ZAKRESIE:

- konstrukcji i oprogramowania mikroprocesorowych systemów kontrolno-pomiarowych
- elektronicznego sprzętu medycznego
- montażu i uruchomień urządzeń elektronicznych w małych seriach na podstawie zleconej dokumentacji technicznej

ZAPRASZAMY!

LIGA MYŚLĄCYCH

Zadanie nr 1

Dwie beczki zawierają razem 240 litrów wody. Gdyby z pierwszej beczki przelać do drugiej tyle litrów wody, żeby zawartość drugiej beczki podwoiła się, a następnie z drugiej przelać do pierwszej tyle litrów wody, żeby zawartość pierwszej beczki podwoiła się to w obu beczkach będzie jednakowa liczba litrów wody.

Ile litrów wody było pierwotnie w każdej beczce?

Zadanie nr 2

Trzech robotników może wykonać pracę w ciągu t godzin. Pierwszy z nich, pracując sam, może wykonać pracę dwa razy szybciej od trzeciego, a o godzinę krócej od drugiego. W jakim czasie każdy z nich może wykonać tę pracę samodzielnie?

Zadanie nr 3

Ile najmniej kuponów TOTO LOTKA należy skreślić, aby uzyskać co najmniej jedno prawidłowe skreślenie sześciu liczb przy założeniu, że skreślamy sześć liczb z 49? Proszę podać sposób rozwiązania.

Rozwiązania zadań prosimy przysyłać pod adresem redakcji do końca grudnia br., z dopiskiem „Liga Myślących”. Punktacja zależy od liczby prawidłowych rozwiązań. Wśród uczestników rozlosujemy książki, a na zwycięzców „Ligi” czekają dodatkowe cenne nagrody — niespodzianki.

```
140 LET p(i)=i
150 NEXT i
154 FOR h=1 TO n
155 LET s=0
160 FOR i=1 TO n
170 LET t=ABS(a(i,p(i)))
180 IF t>s THEN LET s=t: LET j=i
190 NEXT i
200 LET g=p(j): LET p(j)=p(h)
210 LET p(h)=g
220 LET s=a(h,g)
230 LET w=h+1
240 FOR j=w TO z
250 LET q=p(j)
260 LET a(h,q)=a(h,q)/s
270 LET t=a(h,q)
280 FOR i=w TO n
290 LET a(i,q)=a(i,q)-a(i,g)*t
300 NEXT i
310 NEXT j
320 NEXT h
330 FOR h=n TO 1 STEP -1
340 LET s=a(h,z)
350 FOR j=h+1 TO n
360 LET q=p(j)
370 LET s=s-a(h,q)*x(q)
380 NEXT j
390 LET x(p(h))=s
400 NEXT h
410 REM Wdruk      rozwiązanie      układu równan
420 FOR i=1 TO n
430 PRINT "x("i") = "ix(i)
440 NEXT i
```

dokończenie ze str. 21

dokończenie ze str. 28

pakietu o nazwie MACPAINT (zbudowany on został przez Macintosha na komputer Apple).

Dostatecznie duża pamięć komputera pozwala na jednoczesne wykorzystywanie kilku wersji tego samego programu. Jest to bardzo użyteczne np. podczas stosowania PAINTa bowiem zbiór obrazów może być wówczas wspólny dla kilku programów. Wielu użytkowników korzysta z tego samego zbioru obrazów, także obrazy są przekazywane między programami.

Jednym z najaktualniejszych zamierzeń MICROSOFTa (twórcy OKIEN) jest rozwiązanie problemu wieloprogramowości i wielodostępu. Prace w tym kierunku uzależnione są od uzyskania odpowiedniej wersji systemu operacyjnego MS DOS.

Ciekawostką stanowi zapewne całkowicie poprawna praca OKIEN pod systemem GEM.

Odpowiedzią IBMa na prezentowane oprogramowanie Macintosha, OKIEN MICROSOFTa oraz GEMa firmy DIGITAL RESEARCH jest TOPVIEW.

Zamiarem IBMa było osiągnięcie efektu „życzliwych zachowań” personalnych komputerów. Jednak porównując TOPVIEW z OKNAMI lub z GEMem należy z całą stanowczością stwierdzić, iż jest to zdecydowanie gorsze oprogramowanie.

Powierzchnie TOPVIEW jest bardziej podobny do OKIEN niż do GEMa lub Macintosha. System ten nie reprezentuje jednak programu graficznego, nie używa ikon, ponadto sterowanie tym systemem jest bardzo kłopotliwe. Wymagana jest myszka i klawiatura. Część poleceń należy wybierać myszką, a część tradycyjnie przekazywać za pośrednictwem klawiatury. Na przykład, aby cokolwiek napisać, najpierw trzeba skorzystać z myszki, by ustalić miejsce kursora na ekranie monitora mikro, dopiero potem można napisać wybrany komunikat na klawiaturze.

W zasadzie TOPVIEW jest sterowany klawiaturą.

Pierwszą informacją wyświetlaną na ekranie przez ten system jest „TOPVIEW WITA” i pierwszą czynnością, którą należy wykonać jest naciśnięcie klawisza by program pracował dalej — czynność, która nie ma sensu.

Prawdopodobnie IBM chciał w ten sposób zmanifestować komputerową życzliwość...

OKNA i TOPVIEW wymagają pewnych informacji o realizowanych programach użytkowych. Informacje te są, za pomocą specjalnych edytorów, rejestrowane w Program Information File (PIF).

TOPVIEW ma kłopoty z innymi programami niż ze stworzonymi przez IBM. Np. GEM i OKNA doskonale dają sobie radę z LOTUSEm 1-2-3, natomiast TOPVIEW jest tu nieużyteczny. Nawet w LOGO interpreter IBM wprowadza bałagan na ekranie, praktycznie uniemożliwiający pracę.

Zasygnalizowane prace MICROSOFTa, MACINTOSHa, DIGITAL RESEARCH, a także IBMa, bez względu na już uzyskane efekty, są objawem stosunkowo nowego kierunku w informatyce — doskonalenie kontaktu człowiek — komputer. Często efekty tych prac nazywane są interfejsem użytkownika, interfejsem graficznym, „łatwe w użyciu”. Jednak najtrudniej tendencję tę oddaje określenie: ŻYCZLIWOŚĆ KOMPUTERÓW.

Opracował: Włodzimierz GOGOLEK

Pocztowa giełda

Rozwiązanie krzyżówki z „IKS-a” nr 5

Hasło brzmi: **IKS UCZY, BAWI, POMAGA**. Bony pieniężne (1000 zł) wylosowali: Tomasz Dąbrowski — Słupsk, Henryk Duklas — Zgierz, Jolanta Mężyk — Ustroń, Dorota Niezgoda — Sandomierz, Artur Pacholec — Przemyśl.

Nagrody książkowe otrzymują: Magdalena Swatowska — Warszawa, Przemysław Krzak — Kozięglowy, Arek Klimczak — Leśna, Krzysztof Pielarz — Janów Lubelski, Andrzej Eckersdorf — Gorzów Wlkp., Andrzej Ciszek — Gliwice, Janina Sajdak — Warszawa, Paweł Nowak — Poznań, Ewa Sokoll — Bytom, Bartosz Szpunt — Koszalin.

UWAGA — KONKURS!

Miłośnicy literatury science fiction z oryginalnymi pomysłami i fantazją! Każdy z Was ma szansę spróbować swych sił w naszym konkursie na nowelę lub opowiadanie pod hasłem „Przed nami XXI wiek”.

„Najfantastyczniejsze” utwory zostaną uhonorowane przez ziemskie jury, które przyzna nagrody:

- I — 15 tys. złotych
- II — 10 tys. złotych
- III — 5 tys. złotych

Nad ich sprawiedliwym i obiektywnym podziałem czuwać będzie komputer. Autorzy materiałów zakwalifikowanych do druku zostaną ponadto usatysfakcjonowani godziwym honorarium zgodnie z obowiązującymi stawkami.

Entuzjastów informatyki i komputerów nie musimy przekonywać, że i fantazja musi mieć swoje ścisłe parametry. A więc mile widziane będą prace o objętości nie przekraczającej 15 stron maszynopisu bądź czytelnego rękopisu. Jak będzie wyglądał nasz glob opanowany przez komputery n-tej generacji, w jaki sposób informatyka zmieni życie człowieka XXI wieku, jak homo sapiens ułoży sobie koegzystencję z rozumnymi maszynami? To tylko niektóre sugestie organizatorów dotyczące tematyki nadesłanych utworów.

Prace opatrzone godłem należy nadsyłać do końca grudnia, dołączając kopertę podpisaną godłem, zawierającą dane o autorze, pod adresem:

„IKS” — ul. Grzybowska 77
00-950 W-wa

Przychylności weny, zdyscyplinowanej fantazji —
życzy REDAKCJA

PS Zastrzegamy sobie prawo pierwszeństwa publikacji prac nagrodzonych oraz wybranych spośród nie nagrodzonych.

W naszym komputerlandzie

W naszym komputerlandzie rozsiano ostatnio plotki, że według wszelkich reguł prawdopodobieństwa istnieją duże szanse, iż nowy rok będzie nosił numer 1987. Zaniepokojony tymi doniesieniami pracowity Spektrus zwołał kierowany przez siebie zespół robotów na roboczą naradę. Najstarsze i najzacniejsze roboty wypowiedziały się więc konstruktywnie na interesujący wszystkich temat.

Pierwszy robot swoją wypowiedź utrzymał w tonie optymistycznym, stwierdzając jednak, że byłby za ustaleniem numeru 1975, gdyż w komputerlandzie uzyskano właśnie poziom techniki komputerowej przewidzianej w planie na ten rok.

Robot drugi krytycznie ustosunkował się do tej kwestii, gdyż jego zdaniem nie ma sensu dyskutować o roku 1987, skoro nie wyjaśniona jest jeszcze sprawa roku 467 przed naszą erą.

Trzeci dyskutant proponował, by w komputerlandzie przeprowadzić badania ankietowe, które zadecydują o tym czy będzie to rok 1987 czy jakiś inny. Badania najlepiej byłoby przeprowadzić w III kwartale roku przyszłego, a wyniki po opracowaniu przez komputer byłyby znane najdalej pod koniec I półrocza roku następnego.

A czwarty mówca był za wyrównaniem wszystkich cyferek i ustaleniem na przykład numeru 2222. Jeśli bowiem nawet jakiś spekulant będzie próbował sztuczek czytania numeru od końca, to i tak wyjdzie mu to samo, zaś numer 1987 czytany od końca daje 7891, a to już wielka przesada i w ogóle brak wyobraźni.

W swoim podsumowującym słowie Spektrus zauważył, że wstępne rozeznanie pozwala założyć, że numer następnego roku będzie pochodną funkcji roku poprzedniego według wzoru:

$$N = R^2 + \sqrt{(X - Y_z)} + n$$

co zostanie obliczone, jak tylko zespół robotów połapie się, o co chodzi. W porównaniu jednak do roku 1986, rok następny powinien wzrosnąć o około 0,050352 procenta. Dla porównania — jest to wzrost znaczący, zwłaszcza w zestawieniu z latami poprzednimi...

Podglądał:
Eugeniusz MLECZAK



Nie wiemy, kiedy na rynku ukaże się ELWRO 800 Junior, ale na pewno w następnym numerze „IKS-a” znajdziecie:

- kolejny odcinek Basica
- „Jak to zrobić” cz. 2
- rozwiązania zadań z „Ligi Myślących”
- oraz wiele innych ciekawych informacji

IKS — strona 31

Krzyżówka nr 7



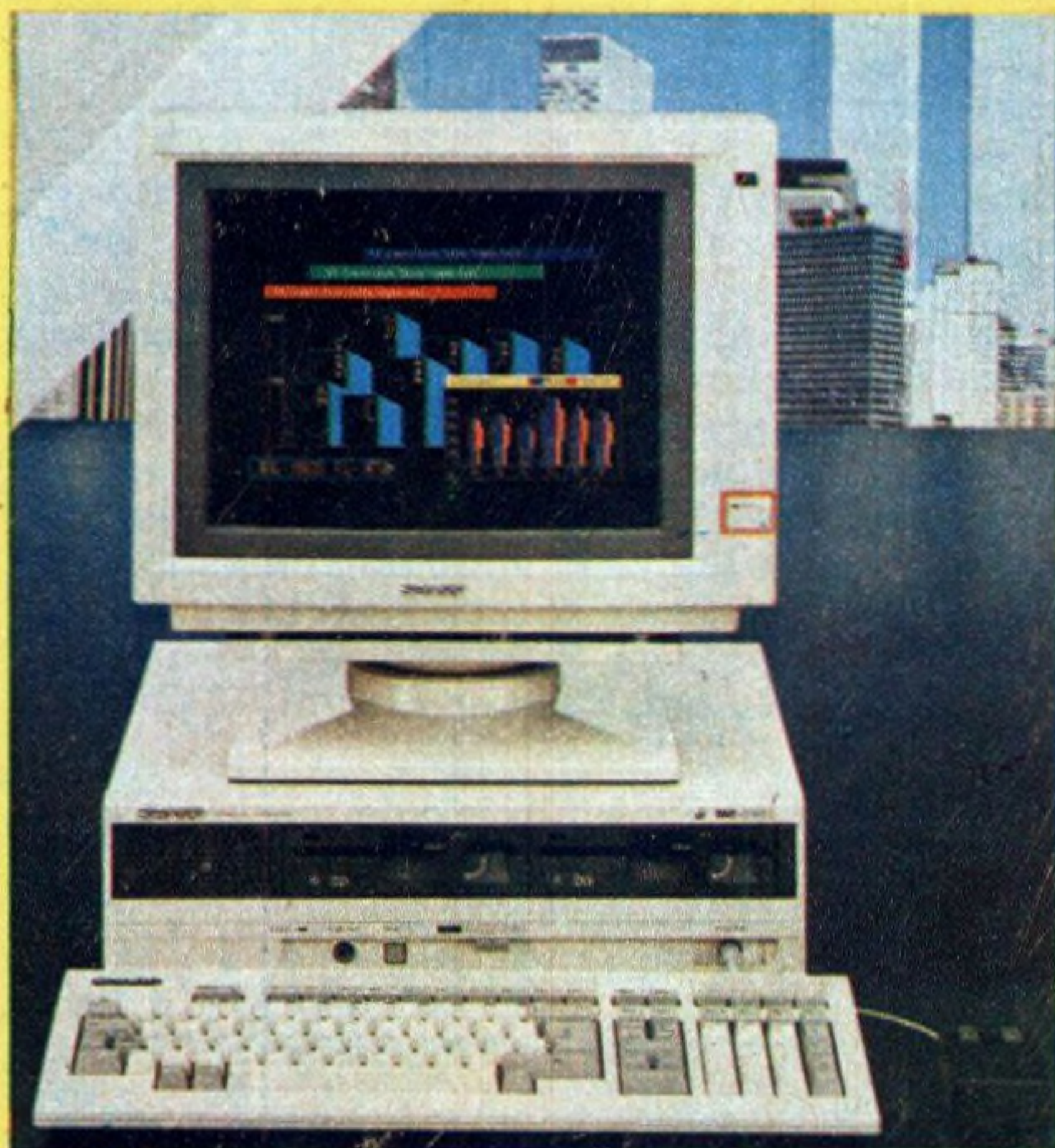
POZIOMO: 1-B naczynie lub tytuł 3-A jeden z rodziny komputerów, 5-A założenie, 5-F jednostka pamięci, 7-D może być senna, 8-A silnikowy lub roślinny, 9-C ...tenisowa 10-A coś rzadko spotykanego, 11-D działacz emigracyjny czł. Młodej Polski, 12-A rajska dziewczyna, 13-A w tytule powieści F. Dostojewskiego, 13-F rezerwa MO, 15-A popisy cyrkowe, 17-B w załodze statku.

PIONOWO: 1-C nietypowy aktor, 1-E jabłko, 1-G rynek w starożytnej Grecji, 1-I akcja na cele dobroczynne, 2-A uzyskuje się go z drzewa lub owoców, 5-B C₂H₅, 6-E jednostka oporu elektrycznego, 7-D złot, 7-F koń, 7-H jego imieniem nazwano Dwór w Gdańsku z XIV w., 8-A pańskie konia tuczy, 8-C okres w dziejach Ziemi, 10-B mała czarna, 11-E 0,01 ha, 11-G jednostka powierzchni, 11-I rodak, 12-A tytuł czasopisma, 12-C wulkaniczny masuw na Wyż. Armeńskiej, 14-E rzeka w ZSRR, 15-G król w Bułgarii.

HASŁO: 3-A, 5-C, 7-B, 9-H, 16-G, 15-H, 9-F, 15-B, 4-I, 9-D.

ROZWIĄZANIA (tylko hasło) prosimy przysyłać pod adresem redakcji do końca grudnia. Wśród czytelników, którzy nadesłają prawidłowe rozwiązania rozlosujemy bonusy pieniężne i nagrody książkowe

Mikrociekawostki



Nowa japońska propozycja — SHARP MZ-5600 to cała seria 16-bitowych mikrokomputerów (MZ-5631, MZ-5641, MZ-5645) zbudowanych na procesorze 8086-2 (8 MHz). Pamięć RAM ma standardowo 256 KB z możliwością rozbudowy do 512 KB. Komputery tej serii mogą być wyposażone w dysk sztywny o pojemności 10 MB. Standardowym systemem operacyjnym na MZ-5600 jest MS-DOS, a językiem programowania oczywiście BASIC.



Rys. Michał Przybyłowski

„IKS” — dodatek „Żołnierza Wolności”. Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stali współpracownicy: Włodzimierz Gogolek, Ireneusz Miernik, Janusz Miller, Michał Przybyłowski, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77, Fotoskład i druk rotograwiurówy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 8241. Nr ind. 382809