

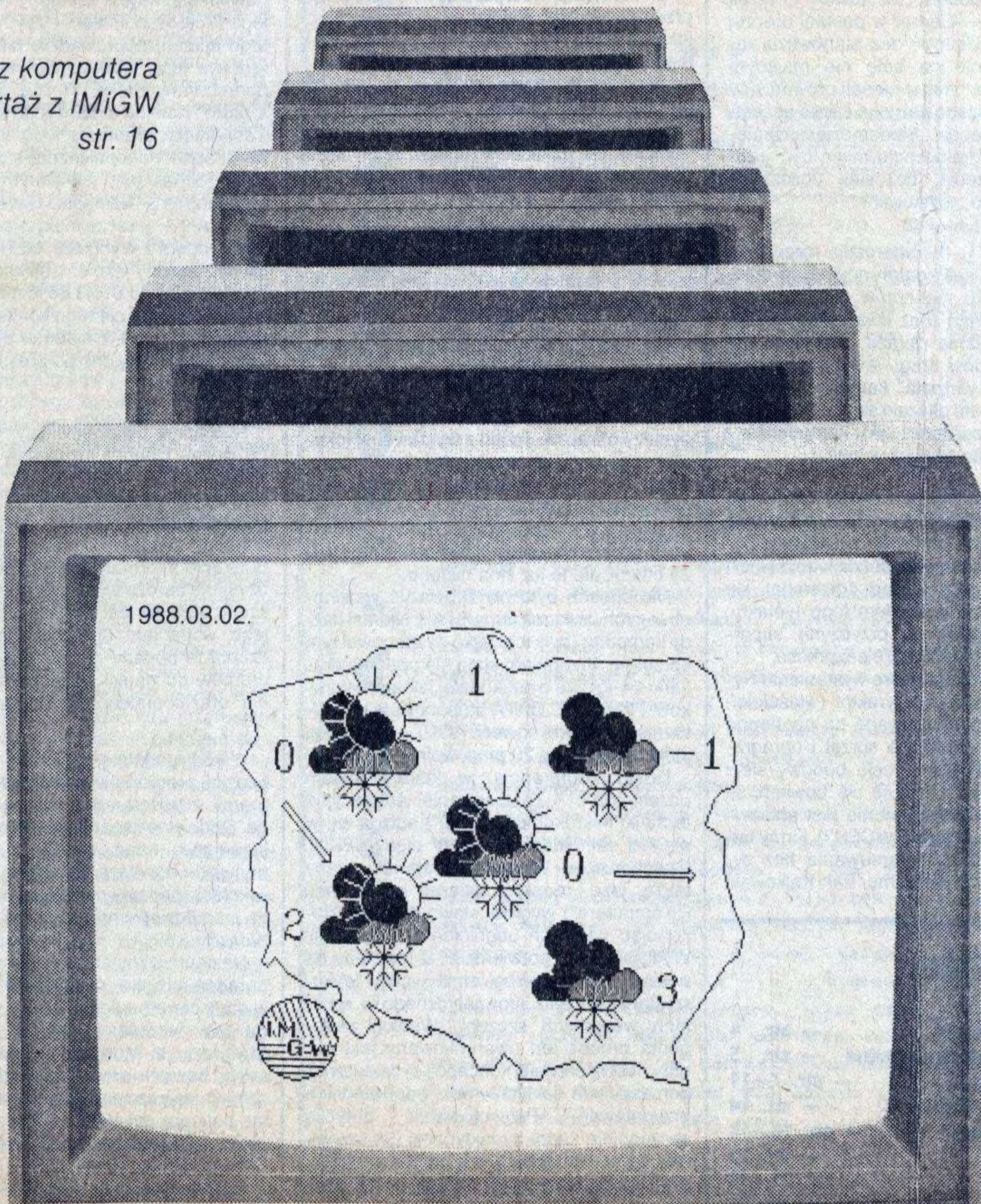
I NFORMATYKA K OMPUTERY S YSTEMY



CENA 80 zł

DODATEK „ŻOŁNIERZA WOLNOŚCI” Nr 2/1988 ISSN 0860—2794

Pogoda z komputera
— reportaż z IMiGW
str. 16



W dniach 9—12 lutego br. odbyła się kolejna konferencja informatyczna. Tym razem aż pod dziewięcioma niezależnymi tytułami. Między innymi omawiane były takie tematy, jak: sieci, wielodostęp, wspomaganie pomiarów, prac wydawniczych, zarządzania i edukacji.

Zważywszy na to, że konferencja była imprezą towarzyszącą wystawie pod tym samym tytułem, jej uczestnicy mogli zobaczyć wiele ciekawego sprzętu i oprogramowania (stanowiąc to będzie przedmiot samodzielnych publikacji, zapewne nie tylko w „IKS-ie”).

Miałem opłacone dwie konferencje (a' 13 000 zł): „Sieci lokalne i wielodostęp” oraz „Zastosowania edukacyjne mikrokomputerów”.

„Sieci” odbywały się już od świtu pierwszego dnia. Konkretnie zgodnie z planem konferencja ta miała się rozpocząć o godz. 10.30. Niestety rozdawanie materiałów i „przepustek” na salę konferencyjną („wizytówki na pierś”) zmieniło nieco ten plan. Bieda tym co nie pobrali „wizytówek” i chcieli wejść na salę — Cerber w postaci uroczej dziewczyny delikatnie, acz stanowczo zagradzał przejście na salę nie mającym „przepustki”. Natomiast na sali prowadzący co kilka minut szacował, jaka część uczestników jest obecna. Siedemdziesięciopięcioprocentowa frekwencja miała być „progim” rozpoczęcia, pozostali uczestnicy nadal czekali po „wizytówki”.

O godzinie 11.10 konferencja rozpoczęła się. Na scenie sali konferencyjnej zgromadzono około 10 monitorów (w tym jeden ciekłokrystaliczny) oraz dwa ekrany — jeden do powielania obrazu LCD oraz wyświetlania slajdów, drugi — wideoskop ukazujący to, co „widziała” kamera — ekrany monitorów. Potem okazało się, że monitory te stanowiły wyposażenie kilku komputerów z terminalami połączonymi w sieć.

Wykład rozpoczął pan doktor (nazwisko nie zostało podane). Było to autentyczne wprowadzenie na poziomie przeliczenia bodów na bajty i odwrotnie, definiowania sieci i wielodostępu. Mój sąsiad zdrzemnął się nieco. Osobiście uniknąłem tego nietaktu, wobec prelegenta, za przyczyną supermocnych lamp panów z Teleexpressu.

Ożywiło nieco atmosferę wystąpienie Ryszarda Kajkowskiego. Dyrektor (właściciel) CSK z Gdyni zwrócił uwagę na pochopne wyrzucanie pieniędzy na sprzęt i oprogramowanie kupowane w celu budowy sieci lub wielodostępu. Okazuje się bowiem, że znaczna część transakcji nie jest przemysłowa (nie konsultowana w CSK?). Firmy kupują sprzęt lub oprogramowanie bez systemowej analizy problemu. Pan Kajkowski

PC STANDARD '88

niemal apelował o kompleksowe podejście do kupowania tego typu systemów. Należy mieć na uwadze całość: odpowiedni sprzęt, oprogramowanie użytkowe oraz system operacyjny. Egzemplifikacją tego problemu jest XENIX, który może pracować tylko przy instalacji niektórych kart interfejsu szeregowego.

Dwa terminale podłączone do XT, a cztery do AT to maksimum znacznie przekraczające moce obliczeniowe tych mikrokomputerów. Są one substytutami minikomputerów. Niestety nie mamy doświadczeń w eksploatacji tego typu sprzętu. Pozwała on na uzasadnione dzielenie swego potencjału na wiele, bardzo wiele terminali.

Interesującą propozycję (handlową) przedstawił gość z Holandii — zwartą sieć lokalną MINSTREL. Idea tego systemu polega na zgromadzeniu całej „ważnej” elektroniki w komputerze centralnym (po jednym procesorze dla każdego terminala). Wątpliwą atrakcją wystąpienia była fatalna jakość obrazów na dużych ekranach. Miały one ukazywać to, co się dzieje na ekranach monitorów sieci.

Potem świadkami kampanii reklamowej sieci (konferencja?) w atrakcyjnej formie pytań, np.: „dlaczego ta sieć jest taka wspaniała?”, odpowiedź: „bo ...”, itp.

Następnie były inne firmy, inne coraz mniej ciekawe wysiłki kolejnych prelegentów, wybrałem oglądanie wystawy.

Najbardziej utkwiała mi w pamięci wizyta na stoisku prezentującym efekty cyfrowej syntezy obrazów. Jeden z wystawców (okazał się artystą-kreatorem komputerowych obrazów) opowiadał mi wrażenia z pracy na sprzęcie, który odtwarza obrazy na monitorze w postaci mozaiki 8000×8000 pikseli, przy czym każdy piksel opisywany jest 24 bitami, ale to już inna historia.

Nie sposób było nie zauważyć wszechobecnych drukarek firmy STAR. Niemal każdy komputer (były ich setki) dysponował tym sprzętem. Przebojem stają się obecnie drukarki 24-igłowe o doskonałej jakości wydruków. Firma ABC DATA proponowała ponadto między innymi nowość LC10 — drukarkę, która zastępuje o 20 proc. wolniejszą NL10.

Druga konferencja, w której mogłem uczestniczyć („Zastosowanie edukacyjne techniki mikrokomputerowej”) odbyła się w pięknej sali Rudniewa na IV piętrze PKiN. Gospodarze — Władysław Majewski, znany także jako redaktor naszej konkurencji („Komputera”) wygłosił słowo wstępne, dokonując pewnych uogólnień i klasyfikacji. Wystąpienie to potwierdziło, iż jesteśmy na początku długiej drogi efektywnego wykorzystania sprzętu informatycznego w szkołach wszystkich szczebli. Według prelegenta proces ten uwarunkowany jest poprawnością przyjętej koncepcji dydaktycznej, poniesionych nakładów oraz odpowiednich przygotowań. Przygotowania dotyczą szczególnie kadry dydaktycznej (jej szkolenia), którą wg głosu z sali pana z ministerstwa, doksztalca w tej materii aż 12 wyższych uczelni na podyplomowych kursach.

Redaktor Majewski zwrócił ponadto uwagę na: brak kryteriów oceny komputerowych programów dydaktycznych oraz na ucieczkę autorów programów dydaktycznych do firm zagranicznych i spółek z o.o. Prelegent wyraził zadowolenie z „pluralizmu” rynku sprzętu komputerowego, wytykając jednocześnie szereg uchybień naszemu byłemu monopolistom — ELWRO (mając na uwadze komputer marki „Junior”). Co mamy obecnie? Co mogą kupować szkoły? Komputer ACORN wymieniony został jako pierwszy (alfabetycznie). Ten brytyjski komputer, firmowany przez BBC, jest obecnie, wg red. Majewskiego, najlepiej edukacyjnie oprogramowanym komputerem na świecie. Nie zmienia to faktu, że komputer ten z całą pewnością nie jest dostępny — o czym prelegent nie wspominał — zbyt wysoka cena tego sprzętu doprowadziła niemal do bankructwa jego producenta (niezbędna była pomoc firmy OLIVETTI, zob. Byte 85/10/9 s. 386). Kolejna propozycja to AMSTRAD. Zasadniczy argument jego stosowania to niska cena. Podobnie rzecz się ma z ATARI, który jednak jest już starym („moralnie”) komputerem. Następnie wymieniony był BOSMAN — jeszcze mało znany nasz komputer rodem z Wybrzeża. MERITUM III został wymieniony jako ostatni komputer szkolny. Podobno od 01.03.88 rozpoczęta będzie masowa produkcja tego komputera (powiedział o tym pan z ministerstwa). Jeśli idzie o klany IBM PC są one po prostu dla szkół za drogie.

Następnie dwie niewiasty, reprezentujące British Council, uzupełniły nieco słowo wstępne red. Majewskiego, podkreślając między innymi, że człowiek w nowoczesnym społeczeństwie powinien dysponować minimalną wiedzą i kulturą informatyczną. Edukacja informatyczna w Wielkiej Brytanii uzyskuje bardzo wymierną pomoc od władz, czego efektem jest (był) komputer BBC oraz jego wspaniałe oprogramowanie. Stwierdzenia te poparte zostały przykładami programów do nauki „ekonomicznego myślenia” oraz poprawności wytapiania stali.

Wśród uogólnień, ciekawy wydaje się podział programów edukacyjnych na programy: z treścią, narzędziowe i symulacyjne. Oddzielne zagadnienie to programy „zastępujące nauczyciela”. Doświadczenia brytyjskie dowodzą, że najpopularniejsze w szkołach podstawowych są proste programy narzędziowe np.: edytory tekstów i bazy danych.

Następnie został „wywołany do tablicy” doradca ministra (niestety nie był przedstawiony) ds. edukacji informatycznej (wspomniany wcześniej pan z ministerstwa). Stwierdził, iż Meritum III jest szansą dla szkół, bowiem produkcja (planowana) 400 sztuk miesięcznie niemal bilansuje, w skali 2 lat, potrzeby szkolnictwa. Komputer ten ma mieć możliwość pracy pod CPM, a nawet pod DOS-em!

Podnoszona była także kwestia praw autorskich dla twórców oprogramowania. Brak

W NUMERZE:

Komputery na start	— str. 4
Programowanie strukturalne	— str. 5
W szponach Atari	— str. 7—11
Informatyka w wojsku	— str. 14
Prognoza pogody	— str. 16—18
Lilavati	— str. 19
Basic	— str. 21—26
Słownik	— str. 29



— Jeżeli będę mówił z sensem, dajcie mi jakiś znak.

tych praw zniechęca ambitnych programistów. Podobno PTI podjęło w tej kwestii odpowiednie kroki. Ponadto, w tym kontekście, wspomniano, że na Zachodzie kupujący nasze oprogramowanie żądają dowodów legalnego zakupu programów narzędziowych, które służyły do budowy przedmiotu transakcji. Na dodatek nasi informatycy mają coraz więcej kłopotów z otrzymywaniem wizy do krajów dysponujących atrakcyjnymi programami.

Pan dr O. Gajl wygłosił ciekawy wykład, zgodny z tytułem konferencji, dokonując między innymi wielu uogólnień oraz klasyfikacji (bariery, co robić, dlaczego, formy działania, warunki realizacji, standardy itp.).

Następnie wystąpił pan profesor, który co kilka lat jeździ za granicę — najczęściej do Kalifornii. Dostrzegł, że kiedyś informatyka (tam) była zajęciem elitarnym, a teraz komputery stoją (tam) na korytarzach. Osobiście sądzi, że proces ten można dostrzec także dużo bliżej — nad Wisłą.

Następnie pan profesor uzasadnił autentyczność komputerowego wspomaganie zdolności intelektualnych człowieka. Komputer ma wyraźny wpływ na mentalność współczesnego człowieka — zmusza do większej precyzji myślenia, ułatwia zobrazowanie (dotychczas tylko werbalnych) informacji. Pytań nie było, tylko jeden z uczestników wyraził w sposób bardzo kategoryczny swoje zdziwienie nad przerostem treści nad formą wystąpienia pana profesora. Większość informacji wyświetlanych na ekranie oraz na monitorach była nieczytelna, przeczyło to zamierzeniu, które wcześniej pan profesor sprecyzował, a także wykazało brak przydatności nowoczesnej techniki w procesie nauczania — wszak to był wykład! Adwersarz pana profesora widział w tym (wystąpieniu) nieświadome odszranianie potencjalnych użytkowników informatyki. Potwierdzało to (ilościowo) ciągłe ubywanie początkowo licznego audytorium.

Konferencję zakończyła oferta Zakładu Edukacji Komputerowej. Jednak i ona, podobnie jak wstęp wygłoszony przez red. Majewskiego nie tylko nie zmieniła, ale utwierdziła garstkę słuchaczy (około 5 proc. początkowej frekwencji) w przekonaniu, że informatyka w naszych szkołach jest przed długą, bardzo długą i wyboistą drogą.

UCZESTNIK

Od kilku lat NASA planuje wysłanie specjalnej sondy międzyplanetarnej o nazwie GALILEUSZ w kierunku Jowisza. Nie posiadając wystarczająco silnej rakiety nośnej przewidywano wstępnie wykorzystanie wahadłowca kosmicznego, lecz katastrofa CHALLENGERA odsunęła realizację tego pomysłu. Ostatnio jednak w laboratorium NASA w Pasadenie (Kalifornia), za pomocą komputera, wytyczono bardzo złożoną i oryginalną trasę lotu, na której w celu rozpędzenia aparatu do niezbędnej prędkości, po wprowadzeniu go w kosmos, wykorzystano pola grawitacyjne Wenus i Ziemi. Podróż na Jowisza zajmie 6 lat, z czego połowę — nabieranie prędkości. Aparat ma krążyć wokół Jowisza przez około 22 miesiące przesyłając na Ziemię informacje naukowe i zdjęcia planety oraz jej czterech satelitów. Zdaniem specjalistów lot GALILEUSZA zapoczątkuje nowy "złoty wiek" badania planet.

■ ■ ■

Na amerykańskim rynku samochodowym pojawił się ostatnio najmniejszy model z rodziny Fordów — FESTIVA. Został on opracowany we współpracy z japońską firmą MAZDA, a produkowany jest w zakładach Kia Motors w Korei Płd. Samochód napędzany jest 4-cylindrowym silnikiem o pojemności 1,3 l i mocy 43 kW. FESTIVA jest kilka centymetrów krótsza od znanej, także na polskim rynku, FIESTY, ma taki sam rozstaw osi i jest nieco szersza.

■ ■ ■

W Stanach Zjednoczonych, po ponad 30 lat trwających pracach badawczych, udało się zbudować maszynę, która łączy w sobie zalety helikoptera i normalnego, napędzanego silnikami turbośmigłowymi samolotu. Ten nowy statek powietrzny, który nazwano OSPREY ma być produkowany w dużych seriach najpierw na potrzeby amerykańskiego lotnictwa wojskowego, a później, co się już rozważa — cywilnego. Jego realizacja była możliwa przede wszystkim dzięki postępowi w technice obliczeniowej (stosowano CAD), w dziedzinie materiałów a zwłaszcza kompozytów. Konstrukcja OSPREY-a składa się w 60% z kompozytu włókna węglowego — żywica epoksydowa, w 20% z żywicy wzmacnianej włóknem szklanym i tylko 20% stanowią detale metalowe, co znacznie obniża jego masę i zwiększa wytrzymałość. W nowej maszynie nie zrezygnowano oczywiście z hydraulicznych elementów poruszających stery, klapy itp., ale komendy pilota docierają do nich w postaci sygnałów cyfrowych za pośrednictwem pokładowych komputerów. Dostawy ponad 900 maszyn tego typu mają się rozpocząć od 1991 roku.

NAUKA • TECHNIKA • NAUKA

TECHNIKA • NAUKA • TECHNIKA

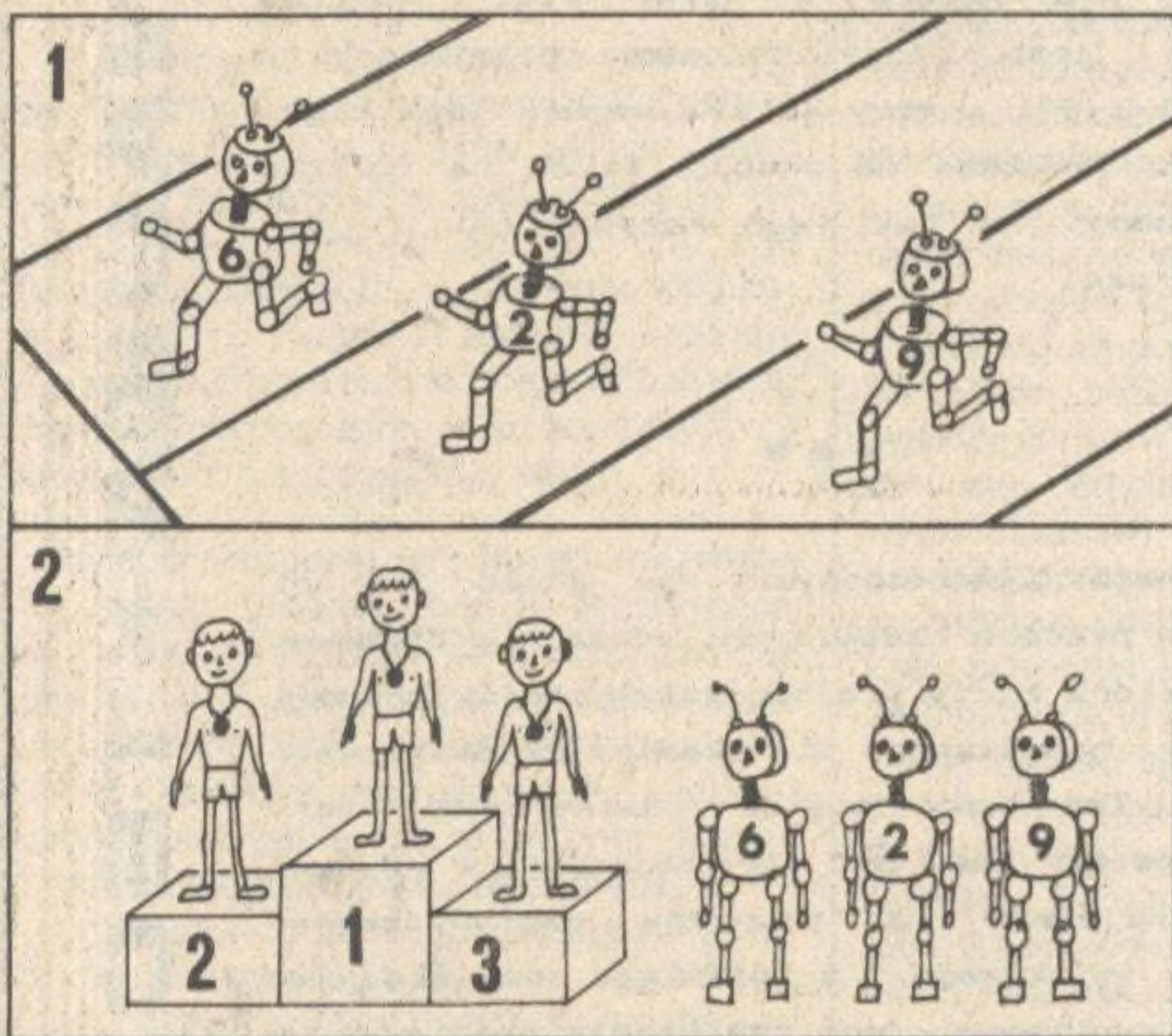
MOIM ZDANIEM

KOMPUTERY NA START

Czyli głos mają Czytelnicy. Pod tym hasłem publikować będziemy opinie i poglądy naszych Czytelników. Nie muszą one być zgodne z poglądami redakcji, ważne jest to, że ich oryginalność zmusza do refleksji, a może nawet pobudzać do twórczego działania.

Porównanie sportu do piramidy jest bardzo trafne: jej podstawa to sport masowy, a wierzchołek — wyczynowy. Nawet nie stosując wzorów matematycznych, łatwo przewidzieć, że im podstawa większa, tym wyżej można wzniesić wierzchołek, nie naruszając stabilności układu. Niestety, w naszym sporcie proporcje nie zostały zachowane, nic więc dziwnego, że w większości dyscyplin piramida runęła lub chyli się ku upadkowi. Jest to widoczne gołym okiem nie tylko dla fachowców i kibiców, błędem jednak jest upatrywanie źródeł słabości w wierzchołku, czyli w sporcie wyczynowym. Bez solidnego zaplecza młodzików i juniorów nie można śnić o potędze opartej na wąskiej grupie zawodników wyczynowych.

Do osiągnięcia sukcesów nie wystarczy też tylko młodzież, nawet najbardziej utalentowana. Niezbędne są jeszcze rzesze solidnie przygotowanych trenerów i instruktorów oraz odpowiednie warunki materialne. Jeśli realnie myślimy o przewyżczeniu sportowego kryzysu, to zarówno trenerom, jak i warunkom materialnym musimy poświęcić dużo uwagi. Na pograniczu tych dwóch elementów znajduje się moim zdaniem, komputer. Na pograniczu, bo chociaż komputer jako taki jest wytworem technicznym, to jednak może, (i w wielu krajach już odgrywa rolę „prawej ręki” trenera. Nie chcę tu gloryfikować techniki komputerowej, uważam, że nawet najlepsza maszyna w pewnych dziedzinach nie dorówna nawet gorszemu wydaniu homo sapiens. Dlatego też komputer jeszcze długo pozostanie tylko technicznym środkiem, wspomagającym pracę trenera, zawodnika i działacza. W czym to wspomaganie się przejawia?



Informatykę w sporcie można wykorzystać w procesie szkoleniowym, pracy naukowo-badawczej, przy organizacji imprez sportowych oraz w zarządzaniu działalnością organizacji sportowych.

Proces szkoleniowy

Mówiąc o procesie szkoleniowym, mam na myśli przygotowania trenerów w Akademiach Wychowania Fizycznego i szkolenie zawodników w klubach. Panuje przy tym opinia, że po pierwsze, bez informatyki trening staje się coraz mniej efektywny, zatem trzeba uczyć studentów pracy z komputerem, po drugie, uważa się, że obecne pokolenie szkoleniowców jest dla informatyki stracone. Tym samym istnieje potrzeba kształcenia studentów AWF z udziałem komputerów i wytworzenia u nich nawyku i umiejętności korzystania z nich w procesie treningowym. Umiejętność tę powinni również zdobyć lekarze, naukowcy i w pewnym stopniu zawodnicy. Do czego może być przydatny komputer w procesie treningowym?

Przede wszystkim do jego doskonalenia oraz do doskonalenia techniki. Komputer umożliwia stworzenie na podstawie istniejącej

wiedzy optymalnego procesu treningowego dla każdej dyscypliny sportowej. Porównanie go z treningiem wybitnych sportowców pozwoli wyciągnąć stosowne wnioski. Więcej, znając psychofizyczne parametry swoich zawodników można opracować dla nich trening zmierzający do osiągnięcia określonych wyników. Prowadzenie ciągłej analizy obciążenia treningowego i uzyskiwanych rezultatów (też za pomocą komputera) pozwala szybko korygować trening, eliminować błędy (przeciążenia, lenistwo itp.) i prognozować wyniki. W zakresie techniki komputer umożliwia jej wzorcowe określenie, porównanie z nią technik poszczególnych zawodników, dokonanie stosownych korekt i określenie odpowiedniego treningu.

Komputer umożliwia również zgromadzenie informacji o zawodnikach, uzyskiwanych przez nich rezultatach, wynikach badań wydolnościowych, testowych i lekarskich. Taki zbiór informacji jest niezbędny do doskonalenia procesu treningowego.

Praca naukowo-badawcza

Konieczność nadszania za postępem w sporcie jest oczywista. Wymaga to stworzenia systemu informatycznego zapewniającego ciągłą aktualizację informacji o stanie nauki, metodach treningu, ewolucji sprzętu sportowego itp. Również prowadzenie własnych badań i doświadczeń bez tej wiedzy jest mało efektywne, a opracowanie ich wyników bez komputera pracochłonne i niekiedy dezaktualizujące je. Niestety, u nas taki system jeszcze nie istnieje.

Organizacja imprez sportowych

Trudno dziś wyobrazić sobie zorganizowanie zawodów sportowych dużej rangi bez wykorzystania komputera. Jego zastosowanie nie tylko ułatwia, ale i przyspiesza ich organizację i prowadzenie. Przyjmowanie zgłoszeń zawodników, tworzenie list startowych, protokołów konkurencji, ustalanie kolejności startów, rejestracja wyników, ustalanie miejsc, wyświetlanie wyników na tablicy, drukowanie końcowych komunikatów, to jedynie niewielki wycinek pracy związanej bezpośrednio z zawodami. Akredytacja dziennikarzy, dysponowanie bazą hotelową, środkami transportu itp., to inne formy działania, w których komputer odgrywa nieocenioną rolę.

Zarządzanie działalnością organizacji sportowych

Zastosowanie informatyki w zarządzaniu związkami sportowymi, klubami, to klasyczne już dziś wykorzystanie komputera.

Przedstawione możliwości zastosowania informatyki w sporcie są rozległe i wymagają wielu przedsięwzięć organizacyjnych. Istniejąca w stosunku do komputerów bariera psychologiczna jest trudna do pokonania. Opory te wynikają głównie z niewiary w możliwość przyswojenia sobie wiedzy i umiejętności posługiwania się komputerem. Bez pokonania tej bariery trudno jest przekonać kogokolwiek o potrzebie stosowania komputerów w procesie treningowym, czy zarządzaniu. Nie mniej istotne są problemy natury organizacyjnej. Wydanie ogólnoresortowych wytycznych w zakresie tworzenia stosownych komórek informatycznych, ujednoczenie sprzętu, szkolenie kadr, to tylko niektóre aspekty, bez których opracowanie jednolity system informatyczny jest niemożliwy do wdrożenia.

Jednak komputer nie jest panaceum na nasze bolączki. Podmiotem działania w sporcie jest, i mam nadzieję, że pozostanie człowiek. Jesteśmy w stanie dokonać komputerowej symulacji nie tylko wybranych fragmentów treningu, ale i czynności psychicznych. Nie oddadzą one jednak atmosfery zawodów, swoistości treningu, a więc tych nieuchwytnych, ale jakże ważnych dla końcowego wyniku czynników. Przeżycia emocjonalne towarzyszą nam na każdym kroku, odzwierciedlają nasz stosunek do otaczającego nas świata i nie pozostają bez wpływu na to, co robimy, i jak robimy. Te procesy psychiczne trudno przedstawić w postaci algorytmu, jeszcze trudniej je przewidzieć i zaprogramować. I dopóki ten stan rzeczy będzie istniał, dopóty walka sportowa będzie miała swój sens i urok.

Ryszard RADZIEJEWSKI

Programowanie strukturalne

Janusz MORBITZER

Koncepcja programowania strukturalnego po raz pierwszy pojawiła się w 1968 r. w artykule holenderskiego informatyka E. W. Dijkstry, pt. „Szkodliwe GOTO” (2). Rozwinięciem idei Dijkstry były późniejsze prace N. Wirtha (5), (6), (7).

Znajomość zasad programowania strukturalnego jest niezbędna każdemu, kto pragnie samodzielnie rozwiązywać problemy i pisać dobre programy.

Programowanie strukturalne często utożsamiane jest z jednym z jego elementów — kodowaniem strukturalnym. Jest to podstawowy błąd, programowanie strukturalne oznacza bowiem pewną filozofię, podstawę przy rozwiązywaniu problemów i zapisywaniu tych rozwiązań w postaci programów. Podstawowymi założeniami są:

- programy mają być w pełni zrozumiałe przede wszystkim dla człowieka, a następnie dopiero dla maszyny,
- struktura programu powinna być odzwierciedleniem struktury problemu, który jest przez ten program rozwiązywany,
- program ma się dać czytać „z góry na dół” (unikanie instrukcji skoku).

Stosowanie zasad programowania strukturalnego przyczynia się w znacznym stopniu do tworzenia przejrzystych i czytelnych programów o mniejszej liczbie i łatwiej wykrywalnych w porównaniu z innymi metodami błędów logicznych. Zasady te mogą być stosowane w odniesieniu do każdego języka wyższego poziomu, także do języka „niestrukturalnego”, jakim jest np. BASIC.

Programowanie strukturalne obejmuje trzy elementy:

- projektowanie zstępujące (ang. top-down),
- programowanie modularne,
- kodowanie strukturalne.

Projektowanie zstępujące — zwane również projektowaniem metodą kolejnych uściśleń (ang. step-wise refinement) — polega na tworzeniu programu w najogólniejszym zarysie, a następnie stopniowym uściśleniu struktury w celu wypełnienia jej szczegółami. Prace rozpoczyna się od spisania celu i założeń projektu.

Jednym zdaniem określa się, co program ma robić. W każdym następnym kroku zadanie rozkładane jest na szereg podzadań, aż do momentu, gdy prostota podzadania pozwala na łatwe napisanie go w postaci fragmentu programu — odrębnego modułu. Każdy moduł powinien zawierać opis wykonywanej funkcji i użytych struktur danych oraz opis danych służących do testowania modułu. Zaletą takiego podejścia jest wczesne tworzenie dokumentacji programu. Bardzo ważne jest również stworzenie projektu przed rozpoczęciem kodowania. Napisanie fragmentu programu stwarza bowiem barierę psychologiczną uniemożliwiającą ulepszenie projektu; łatwiej jest zmienić projekt niż napisany już fragment programu. Wykonując projekt systemu metodą zstępującą mamy pewność, że oprogramowanie będzie odpowiadać celom oraz że popełnione błędy logiczne zostaną wczesnie wykryte. Równoległe sporządzenie dokumentacji zmusza do starannego przemyślenia struktury programu i danych.

Programowanie modularne polega na podzieleniu programu na niezależne części zwane modułami lub segmentami. Podział na moduły staje się widoczny już w fazie projektowania zstępującego — poszczególne podzadania wyznaczają kolejne moduły. Podział dużego programu na mniejsze jednostki służy do ograniczenia złożoności programu, ułatwia jego analizę i zrozumienie.

Poszczególne moduły to „cegielki”, z których budowany jest program. Najważniejszymi cechami modułów są:

- wielkość modułu,
- niezależność.

Nie istnieją ściśle wytyczne, określające wielkość modułu, panuje jednak zgodne przekonanie, że moduły mniejsze są lepsze od dużych. Wielkość modułu powinna być taka, aby łatwo można było jego treść pojąć i zapamiętać. **Dla mikrokomputerów rozsądne jest założenie, że moduł powinien mieścić się w całości na ekranie.** Ułatwia to znacznie analizę i wyszukiwanie błędów.

Przez niezależność modułu należy rozumieć brak ubocznego wpływu na inne części programu, tj. jego niezależność od pochodzenia danych, od przeznaczenia wyników i od wcześniejszego wykorzystania modułu.

Nieprzestrzeganie postulatów modułów prowadzi do tzw. efektu falowego (ang. ripple effect): zmiana w jednym module pociąga za sobą konieczność dokonania zmian w innych częściach programu. Prosty sposób ograniczenia efektu falowego jest unikanie zmiennych globalnych i używanie małych modułów. W języku BASIC wszystkie zmienne są zmiennymi globalnymi. W celu uniknięcia kolizji warto założyć katalog używanych w programie nazw, a w razie potrzeby stosować nazwy dłuższe, mnemoniczne (tj. kojarzące się z pełnią funkcją, np. zmienna o nazwie LICZ używana jest jako licznik itp.).

Dobry moduł powinien mieć cechy podobne do standardowych funkcji matematycznych, takich jak EXP, SQR czy SIN. Są to następujące cechy:

- algorytm rozwiązania zadania,
- zakres dopuszczalnych wartości wejściowych (dziedzina),
- zakres dopuszczalnych wartości wyjściowych (przeciwdziedzina),
- efekty uboczne (komunikaty o błędzie, gdy argument nie należy do dziedziny).

Dobry moduł powinien sprawdzać, czy przekazane mu dane wejściowe należą do dziedziny. W razie rozpoznania błędu moduł nie powinien przerywać wykonywania programu, lecz przekazać odpowiednią informację do modułu bezpośrednio go wywołującego. Każdy moduł powinien mieć jeden punkt wejścia i jeden punkt wyjścia. Należy pamiętać, że czytelność poszczególnych modułów decyduje o czytelności i przejrzystości całości programu.

Kodowanie strukturalne oznacza technikę pisania programów opartą na podstawowych strukturach programowania. Zgodnie z postulatem programowania strukturalnego o pisaniu programów dla ludzi, a nie dla maszyn, wśród podstawowych struktur nie ma instrukcji skoku. Człowiek przetwarza bowiem informacje sekwencyjnie. Im większa liczba odstępstw od sekwencyjności, tym trudniej jest śledzić kompozycję programu. Szczególnym utrudnieniem ze względu na różnorodność możliwych dróg, są skoki warunkowe.

Matematyczną podstawą kodowania strukturalnego jest następujące twierdzenie o strukturze:

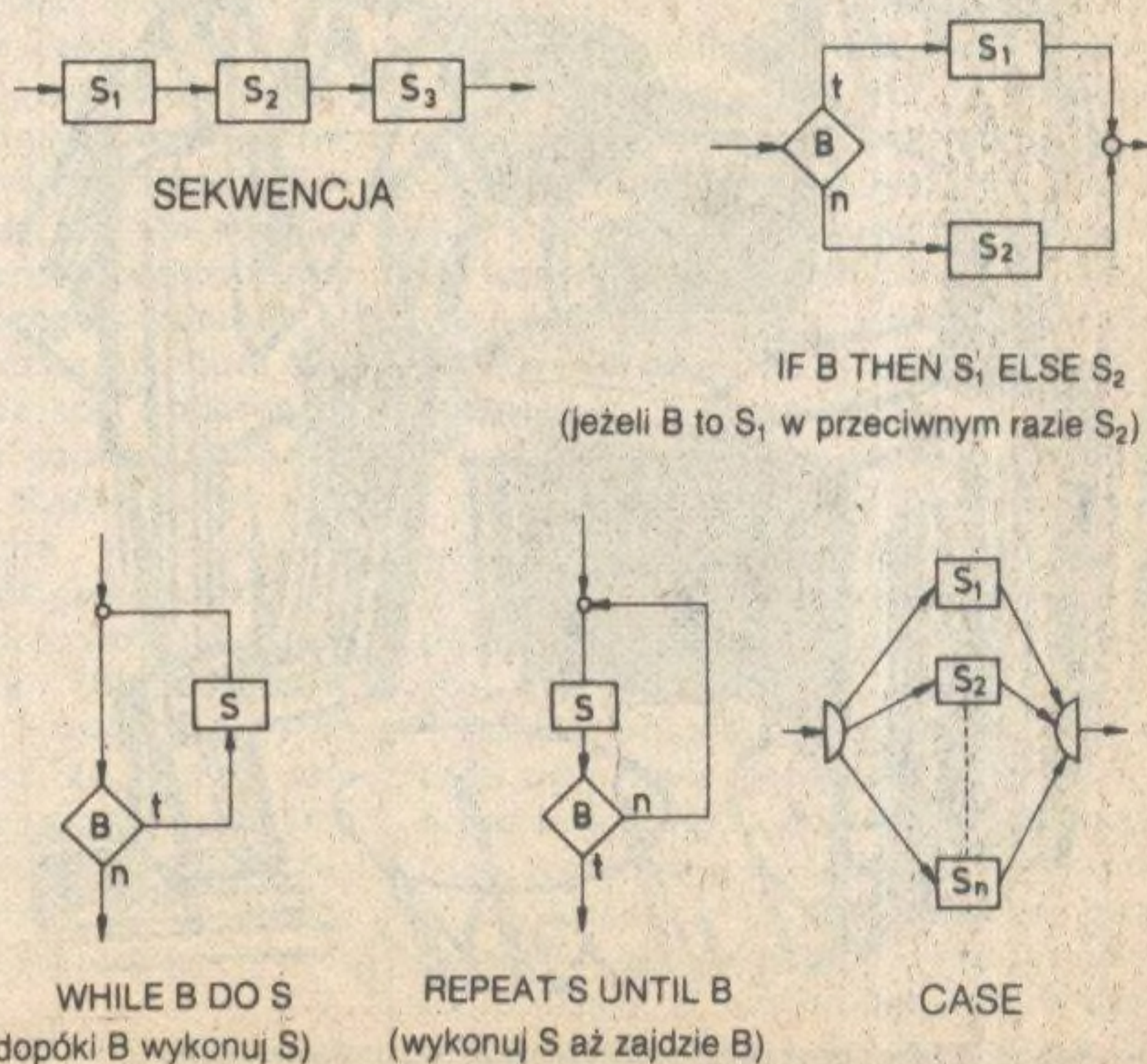
każdy program mający jedno wejście i jedno wyjście, nie zawierający nieskończonych pętli i nieosiągalnych fragmentów, można zapisać przy użyciu tylko trzech podstawowych struktur:

- sekwencji,
- selekcji (IF THEN ELSE),
- iteracji (WHILE DO).

Zwykle w celu ułatwienia programowania dopuszcza się dwie dodatkowe struktury:

- iteracji REPEAT UNTIL
- operacji selektywnego wyboru (CASE).

Wprowadzone tu instrukcje strukturalne przedstawione zostały na poniższym rysunku.



Poprawność programu zostaje zachowana, gdy dowolny prostokąt symbolizujący operację zostanie zastąpiony inną strukturą podstawową. Pętle WHILE DO i REPEAT UNTIL są podobne do siebie. Różnica polega na tym, że w pętli WHILE DO warunek badany jest przed wykonaniem instrukcji, a w pętli REPEAT UNTIL po jej wykonaniu oraz że inny jest warunek wyjścia z pętli — pętla WHILE DO kończy się, gdy warunek staje się fałszywy, natomiast pętla REPEAT UNTIL, kiedy warunek staje się prawdziwy. Wynika z tego, że pętla REPEAT UNTIL wykonana będzie — niezależnie od warunku — przynajmniej jeden raz, podczas gdy pętla WHILE DO może nie zostać wykonana ani razu.

Instrukcja selektywnego wyboru CASE służy do wykonania jednej z operacji S_1, S_2, \dots, S_n w zależności od wartości (całkowitej) wyrażenia zwanego selektorem.

Wszystkie przedstawione konstrukcje programowania strukturalnego występują jedynie w języku PASCAL. W innych językach można — jeśli dana konstrukcja nie występuje — symulować ją. W artykule przedstawione zostaną sposoby zapisywania konstrukcji strukturalnych w języku BASIC dla mikrokomputera ZX Spectrum. Niezbędne tu będzie użycie instrukcji skoku — ale wyłącznie skoku w przód. Nie narusza to postulatu czytelności programu z „góry na dół”.

Konstrukcję strukturalną IF THEN ELSE można symulować w BASIC-u za pomocą instrukcji IF THEN:

```
10 REM konstrukcja REPEAT UNTIL
20 FOR i=1 TO 2 STEP 0
30 ...tu instrukcja (ciąg) S...
40 IF warunek THEN GO TO 60
50 NEXT i
60 REM koniec REPEAT UNTIL
```

W wypadku, gdy po THEN i ELSE wykonywane są pojedyncze instrukcje (lub sekwencje pisane po dwukropku), konstrukcję IF THEN ELSE można zapisać przy użyciu dwóch instrukcji IF THEN:

```
10 IF warunek THEN S1
20 IF NOT warunek THEN S2
```

Konstrukcje strukturalne WHILE DO i REPEAT UNTIL można symulować przy użyciu pętli FOR z bardzo dużą wartością końcową lub — lepiej — z krokiem 0 (zapewnia to nieskończone wykonywanie pętli).

Test opuszczenia pętli dla konstrukcji WHILE DO umieszczony jest na początku, a warunek jest zanegowany:

```
10 REM konstrukcja WHILE DO
20 FOR i=1 TO 2 STEP 0
30 IF NOT warunek THEN GO TO 60
40 ...tu instrukcja (ciąg) S...
50 NEXT i
60 REM koniec WHILE DO
```

W konstrukcji REPEAT UNTIL test znajduje się na końcu, a warunek jest prosty (nie zanegowany):

```
10 REM konstrukcja IF THEN ELSE
20 IF warunek THEN GO TO 60
30 ...wykonywane, gdy fałsz (S2)
40 ...
50 GO TO 80
60 ...wykonywane, gdy prawda (S1)
70 ...
80 REM koniec IF THEN ELSE
```

Do zapisania ostatniej z konstrukcji strukturalnych (CASE) można wykorzystać instrukcję GOTO (w niektórych wersjach języka BASIC znajduje się bardzo wygodna instrukcja ON GOTO). Istotne jest sprawdzenie, czy wartość selektora S_1 nie wykracza poza dopuszczalny zakres.

```
1 REM konstrukcja CASE
5 IF s1<1 OR s1>4 THEN LET s1=5
7 GO TO 10*s1
10 ...tu instrukcja (ciąg) S1...
15 GO TO 60
20 ...tu instrukcja (ciąg) S2...
25 GO TO 60
30 ...tu instrukcja (ciąg) S3
35 GO TO 60
40 ...tu instrukcja (ciąg) S4...
45 GO TO 60
50 ...selektor spoza zakresu...
55 ...(sytuacja awaryjna).....
60 REM koniec CASE
```

Konsekwentne stosowanie zasad programowania strukturalnego prowadzi do tworzenia czytelnych, dobrze przemyślanych programów. Daje możliwość tzw. strukturalnej przechadzki (ang. structured walkthrough), tj. czytanie kodu z góry na dół, bez konieczności skakania po tekście. Program podzielony jest na mniejsze jednostki (moduły), podobnie jak książka podzielona jest na rozdziały. Stosowanie wcięć w tekście programu (akapitowanie) w celu uwidocznienia struktur sterowania dodatkowo zwiększa czytelność. Dopuszczalne jest (oszczędnie!) używanie instrukcji skoku — tylko do przodu. Skok w tył oznacza pętlę — należy wówczas użyć konstrukcji pętli. Należy pamiętać, że najwięcej skoków pojawia się w programie przy nanoszeniu poprawek i uzupełnień. Zapobiega temu konieczność sporządzenia szczegółowego projektu (uwzględniającego wszystkie możliwe sytuacje) przed przystąpieniem do kodowania.

Jak wynika z przedstawionych rozważań, programowanie strukturalne składa się z trzech elementów, z których dwa (projektowanie zstępujące i programowanie modularne) są mało zależne od języka programowania. Trzeci element — kodowanie strukturalne — może być, jak pokazano w artykule, symulowane w innych językach.

Programowanie strukturalne koncentruje się na kompozycji programu. Warto znać i stosować przedstawione zasady. Unikniemy wówczas mozolnej analizy własnego programu, kiedy po miesiącu od napisania go, tracimy w nim orientację (któż z nas robi porządną projekt i dokumentację?!), a nasze programy staną się lepsze, bardziej czytelne i zrozumiałe — dla nas i dla innych.

Janusz MORBITZER

Literatura

- (1) Dahl O.J., Dijkstra E.W., Hoare C.A.R. „Structured programming”. Londyn, Academic Press 1972.
- (2) Dijkstra E.W. „GOTO Statement considered harmful”. CACM, 11/1968.
- (3) Myers G.J. „Projektowanie niezawodnego oprogramowania”. WNT 1980
- (4) Tassel D. „Praktyka programowania”. WNT 1982
- (5) Wirth N. „Program development by step — wise refinement”. CACM 14/1971
- (6) Wirth N. „Wstęp do programowania systematycznego”. WNT 1979
- (7) Wirth N. „Algorytmy i struktury danych”. WNT 1980



— Panno Zlutko, proszę zakodować moje uczucie do pani.

Jak wykorzystywać przerwania?

Tomasz MROWIEC, Ludwik PIELA

Elastyczność działania współczesnych komputerów, również domowych, wynika z ich zdolności selektywnego reagowania na specjalne zdarzenia układowe i programowe, zwane przerwaniami. Spośród kilku rodzajów przerw występujących w Atari, dwa z nich są szczególnie interesujące: VBI — czyli przerwanie powrotu pionowego oraz DLI — przerwanie programu wyświetlania. Pierwsze z nich służy głównie do zsynchronizowania działania układów komputera z sygnałem telewizyjnym wysyłanym do monitora (telewizora) i pojawia się regularnie 50 razy na sekundę. Występuje ono niezależnie od woli użytkownika. Drugie, DLI, jest przerwaniami pochodzącym od specjalizowanego procesora graficznego — ANTIC'u. O jego pojawieniu się i momencie wystąpienia decyduje użytkownik.

Przerwanie powrotu pionowego

W celu uzyskania czystego i wyraźnego obrazu na ekranie monitora lub telewizora niezbędne jest stałe jego odnawianie. Dlatego w pamięci RAM znajduje się obszar, zwany pamięcią ekranu, w którym przechowywane są informacje aktualnie wyświetlane na ekranie. Samo wyświetlanie realizowane jest przez procesor ANTIC, zgodnie z dostarczonym programem wyświetlania, niezależnie od mikroprocesora 6502.

Dla jednoznaczności opisu przypomnijmy definicje pewnych pojęć. Pojedyncze przejście strumienia elektronów przez ekran nazywamy „poziomą linią ekranu”. Okres, w którym strumień wraca od prawego do lewego brzegu, nazywamy „powrotem poziomym”. Okres, w którym strumień elektronów wraca na górę ekranu (bez kreślenia), nazywamy „powrotem pionowym”. Rozkazy programu wyświetlania określają sposób wyświetlania jednej linii. Nie jest to linia ekranu, ale tak zwana linia obrazu, która w zależności od używanego trybu składa się z kilku linii ekranu (w trybie 8 BASIC'u z jednej).

Proces kreślenia obrazu (wykonywany 50 razy na sekundę) zajmuje około 20 000 mikrosekund. W tym czasie następuje wykreślenie 312 linii, od góry do dołu ekranu, tworzących półobraz oraz przemieszczenie strumienia elektronów (po jego uprzednim wygaszeniu) na górę ekranu. Bezpośrednio po wykreśleniu ostatniej linii ekranu i wygaszeniu strumienia elektronów powstaje przerwanie.

Po rozpoznaniu przyczyny przerwania system operacyjny umieszcza na stosie zawartość rejestrów A, X i Y oraz przechodzi do odpowiedniej procedury według adresu umieszczonego w komórce VVBLKI (546,547). Procedura obsługi przerwania, rozpoczynająca się od adresu \$E45F, zwiększa stan zegara (komórka o adresie 20), zmniejsza zawartości liczników (zega-

rów systemowych), dokonuje zmiany kolorów, kopiuje zawartości rejestrów „cieni” do rejestrów układowych i aktualizuje wartości na wyjściach jednostek sterujących. Jej działanie kończy się przejściem do wektora VVBLKD, umieszczonego w komórkach o adresach 548,549. Wskazuje on adres procedury zakończenia przerwania.

Adresy komórek VVBLKI oraz VVBLKD umożliwiają nam śledzenie procedury obsługi przerwania powrotu pionowego oraz wykorzystanie przerwania do własnych celów. W tych dwóch miejscach możliwe jest dołączenie własnej procedury. Będzie ona wtedy realizowana albo przed (natychmiastowa) albo po (opóźniona) wykonaniu systemowej procedury obsługi przerwania. W pewnych przypadkach nie jest obojętne miejsce podłączenia własnej procedury obsługi przerwania. Pierwszy z nich ma miejsce, kiedy nasza procedura odczytuje lub zmienia zawartości rejestrów, które mają swoje „cienie” w pamięci. Może być konieczna zmiana ich zawartości po wykonaniu tego przez procedurę GO. O zawartości decyduje ostatni zapis.

Drugi przypadek występuje, gdy nasza procedura zużywa zbyt dużo czasu procesora. Realizacja procedury obsługi przerwania VBI może być opóźniona poza koniec okresu powrotu pionowego. Lecz wtedy może się zdarzyć, że pewne rejestry związane z grafiką będą zmieniane podczas kreślenia obrazu na ekranie. Można uzyskać nieoczekiwane wyniki. W takim przypadku musi być realizowana po wykonaniu systemowej procedury obsługi, jako tak zwana opóźniona. Limit czasowy dla natychmiastowej procedury obsługi VBI wynosi około 2000 cykli maszynowych, dla opóźnionej — około 20 000 cykli. Jednak wiele z nich realizowanych jest w czasie, gdy strumień elektronów kreśli obraz na ekranie. Dlatego procedury obsługi VBI, związane ze zmianą parametrów obrazu nie powinny być zbyt długie. **Ponadto musimy pamiętać, że czas obsługi przerwania skraca czas procesora dostępny dla realizacji podstawowego programu.**

Po zadecydowaniu, czy procedura VBI ma być natychmiastowa, czy opóźniona, musimy umieścić ją w pamięci (dobrym miejscem jest strona 6 RAM), połączyć z odpowiednią procedurą systemową i zmodyfikować zawartość pewnych komórek pamięci w celu jej wywołania. Natychmiastową procedurę VBI kończymy rozkazem JMP \$E45F, a opóźnioną JMP \$E462. Może się zdarzyć, że chcemy całkowicie ominąć systemową procedurę obsługi VBI (na przykład w celu połączenia czasów przetwarzania), wtedy procedurę natychmiastową kończymy rozkazem JMP \$E462.

Podczas dołączania własnej procedury obsługi przerwania konieczna jest zmiana

zawartości wektora przerwania. Jest to wielkość dwubajtowa, zatem potrzebne są dwa rozkazy przechowania. Mogłoby się zdarzyć wystąpienie przerwania po zmianie pierwszego bajtu, przed aktualizacją drugiego. Może to spowodować upadek systemu. Dlatego należy zabezpieczyć się przed taką sytuacją poprzez wykorzystanie procedury SO SETVBV od adresu \$E45C. W tym celu wprowadzamy do rejestru Y młodszy bajt adresu, do rejestru X — starszy bajt, a do akumulatora 6 dla natychmiastowej procedury VBI lub 7 dla opóźnionej. Następnie wykonujemy JSR SETVBV. Nowa procedura VBI rozpocznie działanie w ciągu 0,02 s.

Własna procedura obsługi przerwania VBI umożliwia wykonanie różnorodnych operacji:

1. W czasie powrotu pionowego strumienia elektronów mogą być wykonywane różnorodne manipulacje ekranem. Mamy pewność, że nie wystąpią one w trakcie kreślenia obrazu, lecz dotyczyć będą całego ekranu. W szczególności możliwa jest rytmiczna animacja, w której zmiany muszą występować w tempie niezależnym od innego przetwarzania.

2. Przerwania powrotu pionowego umożliwiają efektywne sterowanie częstotliwością, głośnością i zniekształceniami dźwięku. Dzięki temu możemy dodać do dźwięku fazy nabrzmiewania i wybrzmiewania.

3. Wykorzystanie VBI umożliwia stałe kontrolowanie wejść użytkownika (50 razy na sekundę) bez zakłócania realizacji programu. Wejścia te wymagają zwykle mało przetwarzania, lecz stałej uwagi. Jest to idealne rozwiązanie problemu zachowania ciągłości obliczeń bez ignorowania użytkownika.

4. Wykorzystanie VBI umożliwia uzyskanie wielozadaniowości w najczystszej postaci. Program pierwszoplanowy może działać pod VBI, podczas gdy program drugoplanowy działa w głównej linii kodu.

Program nr 1 jest przykładem praktycznego wykorzystania przerwania. Umożliwia on uzyskanie dziesięciu kolorów w drugim trybie graficznym (zamiast standardowych czterech). Uzyskane kolory, podobnie jak wszystko na ekranie, są w rzeczywistości iluzją. Następuje mieszanie się kolorów ponieważ obrazy zmieniane są tak szybko, że nasze oczy nie mogą zauważyć zmian. Dlatego widzimy tylko jeden kolor, który jest mieszaniną kolorów we wszystkich obrazach. Możemy mieszać więcej niż dwa kolory jednocześnie, lecz w miarę wzrostu liczby obrazów, wzrasta także ilość migotań ekranu. Praktyczną granicą jest mieszanie czterech kolorów.

Efekt ten uzyskujemy poprzez użycie kilku obszarów RAM ekranu i zmianę bajtów LMS w programie wyświetlania w czasie obsługi przerwania powrotu pionowego.

Szybka zmiana bajtów adresu rozkazu LMS umożliwia przełączenie z jednego obrazu na drugi i z powrotem. Nie można tego zrobić w języku BASIC — jest zbyt wolny. Bajty LMS zmieniane są przez krótką procedurę w języku maszynowym, która działa 50 razy na sekundę, w czasie przejścia strumienia elektronów na górę ekranu.

```

FK 1 REM *****
FJ 2 REM *
XA 3 REM * Program nr 1 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
KI 10 ? CHR$(125):GRAPHICS 2+16:B
    REAK=1000
BI 15 REM
IE 20 FOR I=0 TO 39:READ A:POKE 1
    536+I,A:NEXT I
LI 30 DATA 173,51,6,73,1,141,51,6
    ,240,15,173,40,6,141,49,6,173,
    41,6,141,50,6,76,98,228
XE 40 DATA 173,42,6,141,49,6,173,
    43,6,141,50,6,76,98,228
BL 45 REM
SX 50 DL=PEEK(560)+256*PEEK(561)
BM 55 REM
ZC 60 BYTE=DL+4:GOSUB BREAK
NM 70 POKE 1550,LOW:POKE 1565,LOW
IY 80 POKE 1551,HIGH:POKE 1566,HI
GH
ZR 90 BYTE=DL+5:GOSUB BREAK
HS 100 POKE 1556,LOW:POKE 1571,LO
W
GS 110 POKE 1557,HIGH:POKE 1572,H
IGH
NZ 120 BYTE=DL+20:GOSUB BREAK
EE 130 POKE 1577,HIGH:POKE 1576,L
OW
ZK 140 BYTE=DL-250:GOSUB BREAK
HM 150 POKE 1579,HIGH:POKE 1578,L
OW
RL 155 REM
QZ 160 FOR I=0 TO 10:READ A:POKE
    1600+I,A:NEXT I
GK 170 DATA 104,162,6,160,0,169,7
    ,32,92,228,96
ZA 180 X=USR(1600)
QR 220 REM
QC 230 BYTE=DL+20:GOSUB BREAK
TQ 235 POKE 88,LOW:POKE 89,HIGH
IQ 240 POSITION 0,4
VC 250 ? #6;"  InformaTYka      "
PM 260 ? #6;"  KOMPuTEry       "
LP 270 ? #6;"  sYSTEMy         "
ZT 280 BYTE=DL-250:GOSUB BREAK
TN 290 POKE 88,LOW:POKE 89,HIGH
RD 305 REM
DK 312 SETCOLOR 0,12,6
VL 313 SETCOLOR 1,4,6
ID 314 SETCOLOR 2,15,8
ZF 315 SETCOLOR 3,8,6
IN 320 POSITION 0,4
JT 330 ? #6;"  InformaTYka      "
ZF 340 ? #6;"  KOMPuTEry       "
PY 350 ? #6;"  sYSTEMy         "
KQ 359 REM X=USR(1600)
PC 360 GOTO 360
SN 999 REM
ES 1000 HIGH=INT(BYTE/256)
OK 1010 LOW=BYTE-HIGH*256
AF 1020 RETURN

```

Program nr 2 umożliwia zainstalowanie i wykorzystanie procedury, która wygrywa melodię bez naruszania ciągłości działania programu. Generatory dźwięku wysterowane są w czasie obsługi przerwania VBI na podstawie informacji zawartych w zmiennych tekstowych. Każda ze zmiennych zawiera dane, sterujące jednym generatorem, w postaci par znaków. Kod pierwszego znaku określa wysokość dźwięku, a drugiego — czas trwania w jednostkach równych 0,02 s. Na przykład para: CHR\$(121) i CHR\$(25) zdefiniuje nutę C o czasie trwania 0,5 s. Każdy z ciągów par musi być zakończony dwoma znakami o kodzie 0(CHR\$(0)).

W celu uzyskania dźwięku musimy użyć następującej instrukcji:
 $x = \text{USR}(1536, \text{ADR}(M1\$), \text{ADR}(M2\$), \text{ADR}(M3\$), \text{ADR}(M4\$))$
 gdzie 1536 jest adresem początku procedury (umieszczonej na stronie 6), a M1\$, M2\$, M3\$ i M4\$, są nazwami zmiennych

tekstowych zawierających dane sterujące poszczególnymi generatorami. W instrukcji wywołania używamy tyłu nazw zmiennych, ilu generatorów dźwięku chcemy używać. Na przykład, dla dwóch generatorów instrukcja wywołania ma następującą postać:

$x = \text{USR}(1536, \text{ADR}(M1\$), \text{ADR}(M2\$))$

Sposób użycia opisywanej procedury dźwiękowej stanie się oczywisty po przeanalizowaniu tekstu programu.

W celu uzyskania pauzy należy podać wysokość tonu i (nie może być 0) i żądany czas trwania.

Przerwania programu wyświetlania

Przerwania programu wyświetlania są jedną z bardziej przydatnych właściwości komputera Atari. Wprowadzanie same nie dostarczają dodatkowych możliwości, lecz używane łącznie z innymi właściwościami systemu, jak: grafika gracz—pocisk, wymiana zestawów znaków lub zmiana zawartości rejestrów kolorów, umożliwiają pełne ich wykorzystanie.

```

FK 1 REM *****
FJ 2 REM *
XU 3 REM * Program nr 2 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
QC 10 GRAPHICS 9
TU 15 GOTO 70
JO 20 C=INT(RND(0)*16)
FT 30 COLOR C
OL 40 PLOT INT(RND(0)*80),INT(RND
    (0)*192)
RV 50 GOTO 20
NV 70 GOSUB 32000
RY 75 REM SPOSOB WYWOŁANIA
FU 77 REM X=USR(1536,(ADR(M1$),ADR
    (M2$),ADR(M3$),ADR(M4$))
KN 80 DIM M1$(255),M2$(255):I=0
JI 85 RESTORE 110
LL 90 READ D:I=I+1
OB 95 IF D=-1 THEN 100
QR 97 M1$(I,I)=CHR$(D)
VC 99 GOTO 90
FC 100 I=0
UQ 102 READ D:I=I+1
JT 104 IF D=-1 THEN 100
GA 105 M2$(I,I)=CHR$(D)
MW 106 GOTO 102
TR 108 X=USR(1536,ADR(M1$),ADR(M2
    $))
KF 109 GOTO 20:REM GLOS PIERWSZY
JG 110 DATA 96,25,108,25,121,25,1
    28,25,144,25,128,25,121,25,128
    ,25,144,25,128,25
SL 111 DATA 121,25,128,25,144,25,
    128,25,121,25,108,25,96,25,108
    ,25
DN 112 DATA 121,25,128,25,144,25,
    128,25,121,25,128,25,144,25,12
    8,25
SR 113 DATA 121,25,128,25,144,25,
    128,25,121,25,108,25,96,25,108
    ,25
DT 114 DATA 121,25,128,25,144,25,
    128,25,121,25,128,25,144,25,12
    8,25
ID 115 DATA 121,25,128,25,144,25,
    128,25,121,25,108,25,96,25,91,
    25
NL 116 DATA 96,25,108,25,121,25,1
    28,25,144,25,153,25,144,50,193
    ,150,96,25,91,25
IU 117 DATA 96,50,108,50,121,50,1
    28,50,121,50,144,25,128,25,121
    ,25,108,25,96,25,91,25
KU 118 DATA 81,50,91,50,96,50,108
    ,50,96,50,121,25,108,25,96,25,
    85,25,81,25,72,25
LS 119 DATA 64,50,72,50,81,50,85,
    50,81,50,96,50,60,50,72,50,72,
    25,81,25

```

```

EA 120 DATA 85,25,96,25,64,50,128
    ,50,96,50,128,50,193,50,1,300,
    0,0,-1

```

```

MG 129 REM GLOS DRUGI

```

```

NA 130 DATA 60,25,53,25,47,50,45,
    50,47,50,53,50,47,50,72,100,76
    ,50,72,50,45,50,47,50,53,50

```

```

MG 131 DATA 47,50,72,100,76,50,72
    ,50,45,50,47,50,53,50,35,50,40
    ,25,45,25,47,50,53,50

```

```

KN 132 DATA 60,50,64,25,60,25,53,
    25,64,25,60,25,72,25

```

```

FX 133 DATA 64,25,72,25,76,25,72,
    25,64,25,60,25,53,25,64,25

```

```

ZD 134 DATA 60,25,53,25,47,25,45,
    25,47,25,53,25,60,25,53,25

```

```

YH 135 DATA 47,25,53,25,60,25,64,
    25,72,25,64,25,60,25,53,25

```

```

YJ 136 DATA 47,25,45,25,40,25,35,
    25,40,25,45,25,47,25,45,25

```

```

WN 137 DATA 40,25,45,25,47,25,53,
    25,60,25,53,25,47,25,42,25

```

```

IH 138 DATA 40,25,35,25,31,25,29,
    25,31,25,35,25,40,25,35,25

```

```

PW 139 DATA 31,25,35,25,40,25,42,
    25,47,25,50,25,47,25,42,25

```

```

HI 140 DATA 64,50,40,50,40,25,42,
    25,47,25,50,25

```

```

BB 141 DATA 47,150,1,300,0,0,-1

```

Wprowadzanie obrazu kreślony jest na ekranie 50 razy na sekundę, niebywale szybko jak na nasze możliwości, jednak komputer ma mnóstwo czasu na zmianę parametrów wyświetlania. Istotne jest, aby dokonywać zmiany w czasie każdego kreślenia obrazu, to znaczy 50 razy na sekundę oraz zmieniać interesujący nas parametr dokładnie w tym samym momencie kreślenia obrazu. Oznacza to, że cykl modyfikowania parametrów musi być zsynchronizowany z cyklem kreślenia. Doskonałym sposobem rozwiązania tego problemu może być pojawienie się sygnału przerwania tuż przed momentem zmiany parametrów obrazu. Mikroprocesor rozpoznaje przerwanie, zmienia odpowiednie parametry i wraca do swojej normalnej pracy. Moment pojawienia się przerwania musi być precyzyjnie odmierzony, aby występowało ono dokładnie w tej samej chwili w czasie kreślenia obrazu. Takie specjalnie taktowane przerwanie dostarczane jest przez ANTIC i nazywa się przerwaniem programu wyświetlania (DLI).

Jak wiadomo, ANTIC posiada własny program wyświetlania, którego rozkazy kolejno pobiera i realizuje. Pod koniec wyświetlania linii obrazu pobiera następny rozkaz wyświetlania. Jeśli stwierdzi, że jest w nim ustawiony bit przerwania (D7), to po rozpoczęciu wyświetlania ostatniej linii ekranu aktualnej linii obrazu sprawdza, czy w rejestrze NMIEN (o adresie 54286) dozwolone jest przerwanie programu wyświetlania. Jeśli bit zezwolenia (D7) jest wyzerowany, ANTIC ignoruje żądanie przerwania i kontynuuje swoje normalne zadania. Jeśli bit zezwolenia jest ustawiony, ANTIC wysterowuje linię NMI mikroprocesora 6502, który przechodzi do procedury obsługi przerwania według wektora NMI. Po określeniu przyczyny przerwania przechodzi do procedury obsługi DLI (na podstawie zawartości komórek \$200, \$201 tj. 512 i 513), która może zmienić zawartość jednego lub kilku rejestrów graficznych sterujących wyświetlaniem. Następnie 6502 wraca do realizacji programu głównego.

Jak widać z tego krótkiego opisu, przed wykonaniem procedury obsługi przerwania DLI należy wykonać następujące czynności:

1. Wprowadzić procedurę obsługi przerwania do bezpiecznego obszaru pamięci (idealnym miejscem jest strona 6 RAM).
2. Zmodyfikować pewne bajty programu wyświetlania w celu wykonania przerwania.
3. Do komórek o adresach 512 i 513 wstawić adres pierwszego rozkazu procedury obsługi przerwania programu wyświetlania.
4. Zezwolić na pojawienie się przerwania za pomocą POKE 54286, 192.

Z kolei procedura obsługi przerwania powinna wykonywać następujące czynności:

1. Przechować zawartości rejestrów, które będą używane, poprzez umieszczenie ich na stosie 6502.
2. Wykonać odpowiednie działania.
3. Odtworzyć zawartości przechowywanych rejestrów poprzez zdjęcie ich ze stosu i umieszczenie w odpowiednich rejestrach.
4. Wykonać rozkaz RTI (powrót z przerwania).

Powyższe czynności muszą być wykonywane podczas każdej realizacji procedury obsługi przerwania. W przeciwnym przypadku mogą być zniszczone zawartości rejestrów mikroprocesora (i prawdopodobnie nasz program także).

Podobnie, jak w każdej procedurze obsługi przerwania musimy uwzględnić zależności czasowe. ANTIC nie wysyła sygnału przerwania do 6502 bezpośrednio po napotkaniu odpowiedniego rozkazu; opóźnia to do chwili rozpoczęcia kreślenia ostatniej linii ekranu aktualnej linii obrazu, która ma być przerwana. Dalsze opóźnienia wynikają z czasów działania procedury rozpoznawania przyczyny przerwania oraz przejścia do naszej procedury. Rozpocznie ona zatem działanie, gdy strumień elektronów przebył już część drogi po ekranie. Jeśli procedura DLI zmienia zawartość rejestru koloru, to stary kolor będzie wyświetlony w lewej połowie ekranu, a nowy kolor pojawi się w prawej połowie ekranu w tej samej linii. Z powodu niezbyt precyzyjnego odmierzenia czasu odpowiedzi na przerwanie, granica między nimi nie będzie ostra, lecz denerwująco drgająca.

W celu rozwiązania tego problemu musimy wykorzystać rejestr WSYNC (czekanie na synchronizację poziomą). Jeśli wstawimy rozkaz STA WSYNC tuż przed rozkazem, który zmienia zawartość rejestru koloru, to kolor ulegnie zmianie wtedy, gdy strumień elektronów jest wyłączony (powrót poziomy). Zmiana koloru wystąpi o jedną linię ekranu niżej, lecz będzie czysta i ostra.

Właściwym użyciem programu wyświetlania jest ustawienie bitu D7 w rozkazie programu wyświetlania poprzedzającym linię obrazu, w której chcemy dokonać zmiany.

Prosty przykład procedury obsługi przerwania programu wyświetlania prezentujemy poniżej:

PHA	przechowanie zawartości akumulatora
TXA	
PHA	przechowanie zawartości rejestru X

LDA #\$50	ciemny kolor dla znaków
LDX #\$58	różowy kolor dla tła
STA WSYNC	czekanie
STA COLPF1	zmiana zawartości rejestrów kolorów
STX COLPF2	
PLA	
TAX	odtworzenie zawartości rejestrów X i A
PLA	
RTI	powrót z obsługi przerwania

Powyższa procedura, zrealizowana w programie nr 3, zmienia kolor tła z niebieskiego na różowy. Zmienia także kolor znaków. Okazuje się, że po jej wykonaniu górna część ekranu pozostaje niebieska, pomimo tego, że procedura ustawia kolor różowy w odpowiednim rejestrze. Dzieje się tak dlatego, że procedura obsługi przerwania powrotu pionowego (VBI) wstawia do tego rejestru kod koloru niebieskiego. Pochodzi on z rejestru „cieni” tego rejestru koloru. Każdy z układowych rejestrów koloru ma swój „cień” w komórkach RAM o adresach 708 do 712. W większości przypadków wygodnie jest zmieniać kolory poprzez zmianę zawartości rejestrów „cieni”. Jeśli zmienimy zawartości rejestrów układowych, to w ciągu 1/50 s ustawione kolory zostaną zmienione przez system operacyjny. Jednak w procedurze obsługi DLI musimy zmieniać zawartości rejestrów układowych. Zauważmy również, że nie możemy zmieniać koloru pierwszej wyświetlanej linii obrazu.

```

FK 1 REM *****
FJ 2 REM * *
YO 3 REM * Program nr 3 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
ST 10 DL=PEEK(560)+256*PEEK(561)
FJ 20 POKE DL+15,130
LH 30 FOR I=0 TO 19
UI 40 READ A:POKE 1536+I,A:NEXT I
MA 50 DATA 72,138,72,169,80,162,89
VV 60 DATA 141,10,212,141,23,208
FN 70 DATA 142,24,208,104,170,104,64
TX 80 POKE 512,0:POKE 513,6
CX 90 POKE 54286,192

```

Wstawianie kodów kolorów bezpośrednio do rejestrów układowych powoduje pojawienie się innego problemu: kasujemy automatyczny tryb zwracania uwagi (attract mode), który jest własnością dostarczaną przez system operacyjny. Po dziesięciu minutach bez naciśnięcia jakiegokolwiek klawisza, kolory na ekranie zaczynają się zmieniać, przyjmując przypadkowe barwy przy obniżonej jasności. Gwarantuje to, że komputer pozostawiony bez kontroli na kilka godzin nie wypali obrazu na ekranie telewizora. Musimy zatem rozważyć możliwość utrzymania tej własności. Można to zrobić poprzez wstawienie do procedury obsługi przerwania programu wyświetlania dwóch dodatkowych rozkazów:

stare	nowe
LDA NEWCOL	LDA NEWCOL
STA WSYNC	EOR COLRSH
STA COLPF2	AND DRKMSK
	STA WSYNC
	STA COLPF2

DRKMSK i COLRSH są komórkami pamięci na stronie zerowej (78 i 79) ustawia-

nymi i aktualizowanymi przez system operacyjny w czasie obsługi przerwania powrotu pionowego. Gdy nie ma trybu zwracania uwagi, COLRSH przyjmuje wartość 00a DRKMSK — \$FF. Po wymuszeniu tego trybu, co 4 sekundy do COLRSH wstawiana jest nowa wartość losowa, a DRKMSK przechowuje wartość \$F6. Zatem COLRSH, zmienia kolor, a DRKMSK obcina najstarszy bit jasności.

Wprowadzenie trybu zwracania uwagi do procedury obsługi DLI zaostroża istniejący już problem czasu działania procedury. Szczegółowe rozważania czasowe uczynią go bardziej oczywistym. W realizacji przerwania programu wyświetlania można wyróżnić trzy fazy. Faza 1 obejmuje okres od pojawienia się przerwania do rozkazu STA WSYNC. W czasie jej trwania strumień elektronów kreśli ostatnią linię ekranu przerywanej linii obrazu. Faza 2 obejmuje okres od rozkazu STA WSYNC do momentu pojawienia się strumienia elektronów na ekranie. Odpowiada to przerwie poziomej; wszystkie zmiany obrazu powinny być robione w tym czasie. Faza 3 obejmuje okres od pojawienia się strumienia na ekranie do zakończenia procedury obsługi DLI.

Kreślenie jednej linii ekranu trwa 114 cykli zegara procesora. Przerywanie dochodzi do mikroprocesora w cyklu numer 2. Zużywa on od 8 do 14 cykli na rozpoznanie przyczyny przerwania. Procedury: obsługi przerwania SO i skierowania go do procedury obsługi DLI zużywa 11 cykli maszynowych. W tym czasie od 1 do 3 cykli zostanie zabranych na odświeżenie pamięci DMA. Zatem procedura obsługi DLI nie jest osiągnięta przed upływem 28—36 cyklu zegara. Dla celów planowania musimy przyjąć najgorszy przypadek i programować tak, jakby procedura obsługi DLI osiągnięta była w 36 cyklu. Dodatkowo rozkaz STA WSYNC musi być osiągnięty w cyklu numer 100; redukuje to czas dostępny w fazie 1 o 14 cykli. Ponadto DMA ANTIC’u będzie zabierał pewne cykle maszynowe. Dziewięć cykli będzie straconych na odświeżanie pamięci DMA. Pozostawia to maksymalnie 55 cykli dostępnych dla fazy 1. To maksimum uzyskiwane jest tylko w przypadku pustej linii obrazu. Rozkazy trybu znakowego i graficznego powodują utratę jednego cyklu dla każdego bajtu wyświetlanych danych. Najgorszy przypadek występuje dla trybów BASIC’u 0, 7, 8, 14 i 15, które wymagają 40 bajtów na linię. Zatem procedura fazy 1 będzie miała na realizację od 15 do 55 cykli maszynowych.

Faza 2, krytyczna, rozszerza się ponad 27 cykli zegara. Jak w fazie 1, pewne z nich są tracone na rzecz DMA. Jeśli używana jest grafika P-M, tracimy dodatkowe 5 cykli. Rozkaz wyświetlania zużywa jeden cykl, jeśli używana jest opcja LMS, to tracimy dwa dodatkowe cykle. W końcu, jeden lub dwa cykle mogą być stracone na odświeżenie pamięci lub wyszukiwanie wyświetlanych danych. Zatem dla fazy 2 dostępnych jest od 17 do 26 cykli maszynowych.

Problemy czasowe obsługi przerwania programu wyświetlania stają się teraz oczywiste. Pobranie, zmiana i przechowanie pojedynczego koloru zużywa 4 cykle.

Przechowanie na stosie zawartości rejestrów A, X i Y, a następnie wprowadzenie, zmiana i przechowanie trzech kolorów w A, X i Y zużywają 47 cykli, to jest większość, jeśli nie wszystkie, fazy 1. Programista, który chce używać DLI dla rozbudowanych zmian graficznych, zużywa sporo czasu na kalkulacje czasowe. Na szczęście początkujący programista nie musi koncentrować się na obliczeniach czasowych. Jeśli zmienia się tylko jeden kolor lub wykonywane są proste operacje graficzne, zliczanie cykli i optymalizacja procedury nie są konieczne.

Nie ma żadnych prostych opcji dla programisty, który potrzebuje zmienić więcej niż trzy rejestry koloru w pojedynczym przerwaniu. Może być możliwa zmiana czwartego koloru na początku fazy 3, jeśli nie jest on wyświetlany na lewym brzegu ekranu. Podobnie, kolor, który nie występuje przy prawym brzegu ekranu, może być zmieniony w fazie 1. Innym sposobem podejścia jest rozdzielenie jednej procedury obsługi DLI o nadmiernej aktywności na dwie mniej ambitne, z których każda wykonuje połowę pracy oryginału. Drugie DLI można uzyskać przez wstawienie pojedynczego rozkazu pustej linii ekranu (z ustawionym bitem DLI) do programu wyświetlania tuż poniżej przerywanej linii obrazu. Zajmuje to jednak pewien obszar ekranu.

Innym częściowym rozwiązaniem jest dokonanie zmiany kolorów w czasie przerwy pionowej. W tym celu w RAM muszą być przechowywane dwie tablice kolorów. Pierwsza z nich zawiera kody kolorów, które mają być wyświetlane przez procedurę obsługi DLI, druga zawiera wartości umożliwiające zmianę kolorów. W czasie powrotu pionowego dostarczona przez nas procedura obsługi przerwania pobiera kod koloru z pierwszej tablicy, zmienia go i przechowuje w drugiej tablicy. Procedura obsługi DLI pobiera wtedy wartości bezpośrednio z drugiej tablicy, bez straty dodatkowego czasu na zmianę.

Często pożądane jest posiadanie kilku procedur obsługi przerwania programu wyświetlania występujących w kilku miejscach ekranu. Niestety, istnieje tylko jeden wektor DLI. Jeśli chcemy zastosować kilka procedur obsługi przerwania, to w samej procedurze musimy umieścić kierowanie do odpowiedniej obsługi. Można to zrobić na kilka sposobów. Jeśli procedura obsługi DLI wykonuje taki sam proces z różnymi wartościami, wtedy może być sterowana tablica. Po każdej realizacji procedury zwiększany jest licznik i używany jako indeks tablicy wartości. Poniższy przykład prezentuje ten sposób.

PHA	
TXA	przechowanie zawartości rejestrów
PHA	
INC LICZ	zwiększenie licznika
LDA TABCKOL, X	użycie strony 2 dla tablicy kolorów
	oczekiwanie
STA WSYNC	
STA COLBAK	
CPX #4F	czy ostatnia linia?
BNE KON	nie, wyjdźcie
LDA #500	tak, wyzerowanie licznika

STA LICZ
KON PLA
TAX
PLA
RTI

odtworzenie zawartości rejestrów

W celu wykorzystania powyższej zasady należy wprowadzić program nr 4, który umieszcza na ekranie 80 kolorów jednocześnie.

Innym sposobem zastosowania wielokrotnych DLI jest użycie licznika przerwania jako testu dla przejścia do odpowiedniej procedury obsługi. Jednak lepszym będzie rozwiązanie, w którym każda procedura obsługi DLI wpisuje adres następnej do komórek \$200, \$201. Może to być wykonywane w fazie 3. Jest to najbardziej uniwersalne rozwiązanie problemu wielokrotnych DLI. Ma tę dodatkową zaletę, że rozkazy sterowania wyborem wykonywane są po części krytycznej obsługi, a nie przed.

```
FK 1 REM *****
FJ 2 REM * *
ZI 3 REM * Program nr 4 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
PC 10 GRAPHICS 7
SU 20 DL=PEEK(560)+256*PEEK(561)
OW 30 FOR J=6 TO 84
KH 40 POKE DL+J,141
JC 50 NEXT J
HI 60 FOR J=0 TO 30
WK 70 READ A:POKE 1536+J,A:NEXT J
QM 80 DATA 72,138,72,238,32,6,175,32,6
VZ 90 DATA 189,0,240,141,10,212,141,26,208
IJ 100 DATA 224,79,208,5,169,0
UM 110 DATA 141,32,6,104,170,104,64
VP 120 POKE 512,0:POKE 513,6
OJ 130 POKE 54286,192
```

Procedura systemowa trzasku (klekotu) klawiatury interferuje z procedurą obsługi DLI. Po naciśnięciu i rozpoznaniu klawisza tworzony jest trzask. Jego czas trwania odmierzany jest przez kilka rozkazów STA WSYNC. Może to naruszyć taktowanie procedury obsługi DLI i spowodować, że kolory przeskoczą w dół (o jedną linię ekranu) na ułamek sekundy. Rozwiązanie tego problemu nie jest łatwe. Jeden ze sposobów wykorzystuje rejestr VCOUNT, który informuje, która linia ekranu jest aktualnie wyświetlana przez ANTIC. Procedura obsługi DLI może badać jego zawartość w celu podjęcia decyzji kiedy zmienić kolor. Innym rozwiązaniem może być wyłączenie systemowej procedury obsługi klawiatury i dostarczenie własnej, lecz wymaga to sporo pracy. Ostatecznym rozwiązaniem jest brak akceptacji wejść z klawiatury. Jeśli naciśnięty klawisz nie jest rozpoznany, wstrząsy ekranu nie występują.

Zastosowania

Dopiero teraz staje się oczywista ogromna wartość grafiki pośredniej i rejestrów układowych, których zawartość możemy zmieniać. Za pomocą przerwania programu wyświetlania każdy z tych rejestrów może być zmieniany „w locie”. Najbardziej oczywistym zastosowaniem DLI jest umieszczenie na ekranie większej liczby kolorów. Każdy rejestr koloru może być zmieniany po wystąpieniu przerwania. Dotyczy to zarówno rejestrów koloru pola gry, jak i rejestrów koloru graczy. Mamy zatem

dziewięć rejestrów, z których każdy może wyświetlać jeden z 256 różnych kolorów. Oczywiście normalny program nie będzie używał wszystkich kolorów. Z kolei zbyt wiele przerwania spowalnia cały program. W praktyce tuzin kolorów jest łatwy do uzyskania, dwa tuziny wymagają uważnego planowania, a więcej — wymaga wymyślonej sytuacji.

Przerwanie programu wyświetlania mogą także zwiększyć możliwości grafiki P-M. Pozycja pozioma gracza może być zmieniana podczas DLI. W ten sposób gracz może mieć kilka wcieleń (różnych kształtów) na ekranie.

Innym sposobem wykorzystania DLI jest zmiana szerokości lub priorytetu gracza. Ostatnim zastosowaniem przerwania programu wyświetlania jest zmiana zestawu znaków w dolnej części ekranu. Umożliwia to używanie grafiki znakowej w dużym oknie i zwykłego tekstu w oknie tekstowym. Możliwe są wielokrotne zmiany zestawu znaków; program może używać jednego zestawu znaków u góry ekranu, innego w środku i normalnego zestawu na dole.

Właściwe wykorzystanie przerwania programu wyświetlania wymaga uważnego zaprojektowania obrazu. Musimy pamiętać, że obraz ma „bardziej” pionową strukturę niż poziomą. Wynika to z systemu tworzenia obrazu, w którym tempo kreślenia w poziomie jest 262 razy szybsze od tempa kreślenia w pionie. Komputerowy obraz nie jest czystą kartką papieru, na której kreślimy, jest to stos cienkich paszków, z których każdy może być inny. I musi to być brane pod uwagę podczas projektowania obrazu.

Prezentowane pogromy demonstrują kilka z możliwych zastosowań przerwania programu wyświetlania.

Program nr 5 umożliwia dokonanie zmiany koloru tła w dowolnej części ekranu. Ze względu na to, że zmieniamy zawartość rejestru układowego, zmienia się kolor części ekranu poniżej miejsca przerwania. Górna część ekranu pozostaje bez zmian, ponieważ rejestr układowy aktualizowany jest zawartością rejestru „cienia” podczas realizacji procedury obsługi VBI.

Dla dolnej części ekranu możemy wybrać dowolny kolor. W tym celu określamy kod koloru i umieszczamy jako trzecią liczbę w instrukcji DATA (linia 150). Możemy również zmieniać ten kolor bezpośrednio, za pomocą POKE z adresem 1538.

Możemy również zmienić kolor tła w górnej części ekranu, poprzez wstawienie odpowiedniej wartości do rejestru „cienia” o adresie 712.

Możliwa jest również zmiana koloru dowolnego z kreślonych prostokątów. W tym celu musimy wstawić żadaną wartość koloru do odpowiedniego rejestru „cienia” (dla górnej części prostokąta) lub zmienić zawartość odpowiedniego rejestru układowego (dla dolnej części prostokąta). Lewy prostokąt kreślony jest przy użyciu COLOR 1, środkowy — przy użyciu COLOR 2, a prawy — COLOR 3.

Jeśli chcemy dokonać tych zmian w samym programie musimy zmienić ósmą liczbę w instrukcji DATA w linii 150 dla młodszego bajtu adresu odpowiedniego rejestru układowego. Dla prostokąta uży-

wającego COLOR 1 wstawimy 22, dla COLOR 2 — 23 i dla COLOR 3 — 24.

W celu wybrania różnych pozycji pionowych dla zmiany koloru, dodajmy 128 do różnych rozkazów programu wyświetlania. Wykorzystamy do tego linię 240, która umieszcza żądanie przerwania programu wyświetlania w odpowiedniej linii. W używanym trybie 5 BASIC'u otrzymujemy 40 linii obrazu, możemy zatem umieszczać żądania przerwania w komórkach od START+6 do START+44.

```
FK 1 REM *****
FJ 2 REM *
AC 3 REM * Program nr 5 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
QM 100 REM
QC 110 FOR I=0 TO 10
SK 120 READ A
RB 130 POKE 1536+I,A
FZ 140 NEXT I
ZY 150 DATA 72,169,6,141,10,212,1
41,26,208,104,64
QY 160 REM
VZ 170 POKE 512,0:POKE 513,6
VB 210 GRAPHICS 5
FI 220 START=PEEK(560)+256*PEEK(5
61)
BC 240 POKE START+24,10+128
RQ 310 FOR X=1 TO 3
OC 320 COLOR X
LF 330 PLOT 20*X+5,30:DRAWTO 20*X
+5,10
NY 340 DRAWTO 20*X-5,10:POSITION
20*X-5,30
GM 350 POKE 765,X
SJ 360 XIO 18,#6,0,0,"S:"
MB 370 NEXT X
RE 380 REM
JT 390 FOR D=1 TO 300:NEXT D
QP 400 REM
OI 410 POKE 54286,192
NX 420 END
```

Program nr 6 umożliwia zademonstrowanie reklamowanych możliwości komputera Atari, to znaczy uzyskanie na ekranie 128 kolorów jednocześnie. Oprócz dostarczenia procedury obsługi przerwania, wymagało to także zmodyfikowania programu wyświetlania w celu otrzymania odpowiedniej liczby wyświetlanych linii.

Procedura obsługi przerwania tak została napisana, że za każdym jej użyciem wartość kolor/jaskrawość zwiększana jest o 2. Na końcu obrazu otrzymamy 128 kolorów.

```
FK 1 REM *****
FJ 2 REM *
AW 3 REM * Program nr 6 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
UR 110 FOR I=0 TO 17
SU 120 READ B
RS 130 POKE 1536+I,B
FZ 140 NEXT I
VX 160 POKE 512,0:POKE 513,6
WR 210 GRAPHICS 8
FQ 220 DL=PEEK(560)+256*PEEK(561)
XQ 230 COLOR 1
NW 240 FOR I=1 TO 5
TH 250 READ X1,Y1,X2,Y2,X3,Y3,X4,
Y4
FD 260 PLOT X1,Y1:DRAWTO X2,Y2:DR
AWTO X3,Y3:POSITION X4,Y4
LP 270 POKE 765,1
SM 280 XIO 18,#6,0,0,"S:"
GK 290 NEXT I
JW 310 POKE DL+3,78
WK 320 FOR I=6 TO 33:POKE DL+I,14
:NEXT I
CN 330 FOR I=34 TO 98:POKE DL+I,1
4+128:NEXT I
RA 340 POKE DL+99,78+128
```

```
CT 350 FOR I=102 TO 164:POKE DL+I
,14+128:NEXT I
FC 360 FOR I=165 TO 198:POKE DL+I
,14:NEXT I
EC 370 POKE DL+199,65
SM 380 POKE DL+200,80
EW 390 POKE DL+201,128
OI 410 POKE 54286,192
QT 800 REM
HJ 810 DATA 72,173,198,2,24,105,1
KN 820 DATA 141,10,212,141,198,2,
141,24,208,104,64
QZ 830 REM
CD 840 DATA 139,155,116,27,77,27,
100,155
GA 850 DATA 219,155,242,27,203,27
,179,155
LO 860 DATA 189,99,160,35,158,35,
129,99
TZ 870 DATA 140,155,159,99,130,99
,139,155
QU 880 DATA 179,155,189,99,159,99
,178,155
```

Program nr 7 demonstruje sposób wydrukowania komunikatu, w trybie GR. 1 lub 2, zawierającego wielokolorowe znaki (do siedmiu kolorów jednocześnie). Każdy kolor występuje w jednej linii ekranu.

```
FK 1 REM *****
FJ 2 REM *
BO 3 REM * Program nr 7 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
WR 40 DIM MSS$(120)
GK 50 TRAP 1000
OP 60 ? "PODAJ TRYB 1 LUB 2":INP
UT A:IF A<>1 AND A<>2 THEN 60
SJ 70 GOSUB 100:GRAPHICS A+16
DA 80 DL=PEEK(560)+256*PEEK(561)+
6
FJ 90 FOR I=0 TO 6:POKE DL+I,133+
A:NEXT I:GOTO 200
QM 100 REM
WP 110 POKE 82,10:POKE 83,29:?
QQ 120 REM
QP 130 ? :? "PODAJ TEKST"
UQ 140 INPUT MSS$:RETURN
QN 200 REM
FF 210 FOR I=0 TO 42:READ B:POKE
1553+I,B:NEXT I
OE 220 DATA 72,138,72,152,72,162,
0,141,10,212
MD 230 DATA 189,1,6,188,9,6,141,2
4,208,140,25,208
CE 240 DATA 232,173,0,6,201,1,144
,3,141,10,212
XZ 250 DATA 224,6,144,226,104,168
,104,170,104,64
QQ 300 REM
NF 310 FOR I=0 TO 73:READ C:POKE
1599+I,C:NEXT I:POKE 1596,0
IX 320 DATA 72,138,72,173,60,6,20
1,4,144,55,162,7
KO 330 DATA 173,8,6,141,61,6,173,
9,6,141,62,6
DU 340 DATA 202,189,1,6,157,2,6
TZ 350 DATA 224,0,208,245,173,61,
6
QQ 360 DATA 141,1,6,162,0,189,10,
6,157,9,6,232,224,7,208,245,17
3,62,6,141,16,6,169,0
BD 370 DATA 141,60,6,238,60,6,104
,170,104,76,98,228
QP 400 REM
WW 410 FOR I=0 TO 7:POKE 1537+I,(
8+I)*16+I*2:NEXT I
ND 420 FOR I=0 TO 7:POKE 1545+I,(
1+I)*16+I*2:NEXT I
VN 430 POKE 1536,A
MI 490 POSITION 0,0:? #6;" "
QQ 500 REM
MS 510 POKE 512,17:POKE 513,6:POK
E 54286,192
ZK 520 POKE 548,63:POKE 549,6:POK
E 54286,192
ZS 600 POSITION 1,2:? #6:MSS$
NW 610 GOTO 610
VY 1000 TRAP 40000:GOTO 50
```

Po wprowadzeniu programu i uruchomieniu go będziemy proszeni o wybranie trybu (1 lub 2). Następnie musimy podać komunikat, który chcemy wyświetlać. Wielkie i małe litery będą miały swoje normalne kolory: pomarańczowy i jasnozielony. Negatywy wielkich i małych liter będą wielokolorowe. Poszczególne kolory będą przesuwane się w przeciwnych kierunkach. Maksymalna długość komunikatu wynosi 120 znaków, lecz może być zmieniona (zmienna MSS\$). Podczas wprowadzania marginesy są tak ustawione, symulują szerokość obrazu w trybie 1 lub 2, aby łatwo było розміścić komunikat.

Spójrzmy na linię 400. Dwie pętle FOR...NEXT przechowują różne wartości kolorów w tablicy, umieszczonej na stronie szóstej, dla użycia ich przez procedury obsługi DLI i VBI. Zauważmy, zawartości koloru i jaskrawości wzrastają dla każdej komórki, w rezultacie kolor i jaskrawość zmieniają się dla każdej linii ekranu.

Jeśli chcemy, łatwo możemy zmienić te wartości dla otrzymania innych efektów. Na przykład, zachowując tę samą wartość jaskrawości, uzyskamy ostry kontrast między kolorami. A dla otrzymania większego zakresu kolorów należy zmieniać numer koloru o więcej niż jeden dla każdej linii ekranu (np.: I*16+1.2). Jest duża liczba możliwości, które możemy brać pod uwagę dla otrzymania pożądanego efektu.

Procedura obsługi VBI zawarta jest w instrukcjach Data od linii 300. Jej funkcją jest cykliczna zmiana dwóch tablic kolorów, w różnych kierunkach, dla otrzymania efektu ruchomej kurtyny. Jeśli chcemy mieć obraz z nieruchomymi kolorami, możemy wyeliminować procedurę obsługi VBI przez usunięcie linii 520 i dodanie POKE 54286, 192 do linii 510.

Inną łatwą modyfikacją jest zmiana szybkości rotacji kolorów. Liczba 4 po liczbie 201 w linii 320 steruje szybkością zmian. Zmniejszenie tej liczby zwiększa szybkość rotacji; zwiększenie jej — zmniejsza. Zauważmy, że maksymalną liczbą jest 255. Jeśli przekroczymy tę wielkość, zostanie zarejestrowany błąd, który zostanie przechwycony przez TRAP (linia 50); spowoduje to rozpoczęcie działania programu od początku. I tak będzie do chwili poprawienia błędnej wartości dla POKE.

Podczas przygotowywania tego artykułu korzystaliśmy z opisów i przykładów zawartych w „De Re Atari”, „Second book of Atari graphics” oraz „Second book of Atari”.

UWAGA!

Dwa znaki z lewej strony numeru linii nie są częścią programu, lecz sumą kontrolną linii dla Edytora BASIC'u („IKS” nr 4 z 87 r.). Nie wprowadzamy ich do komputera.

Występujące w programach podkreślone znaki należy wprowadzić w negatywie.

Tomasz MROWIEC
Ludwik PIELA

Grafika trójwymiarowa

```

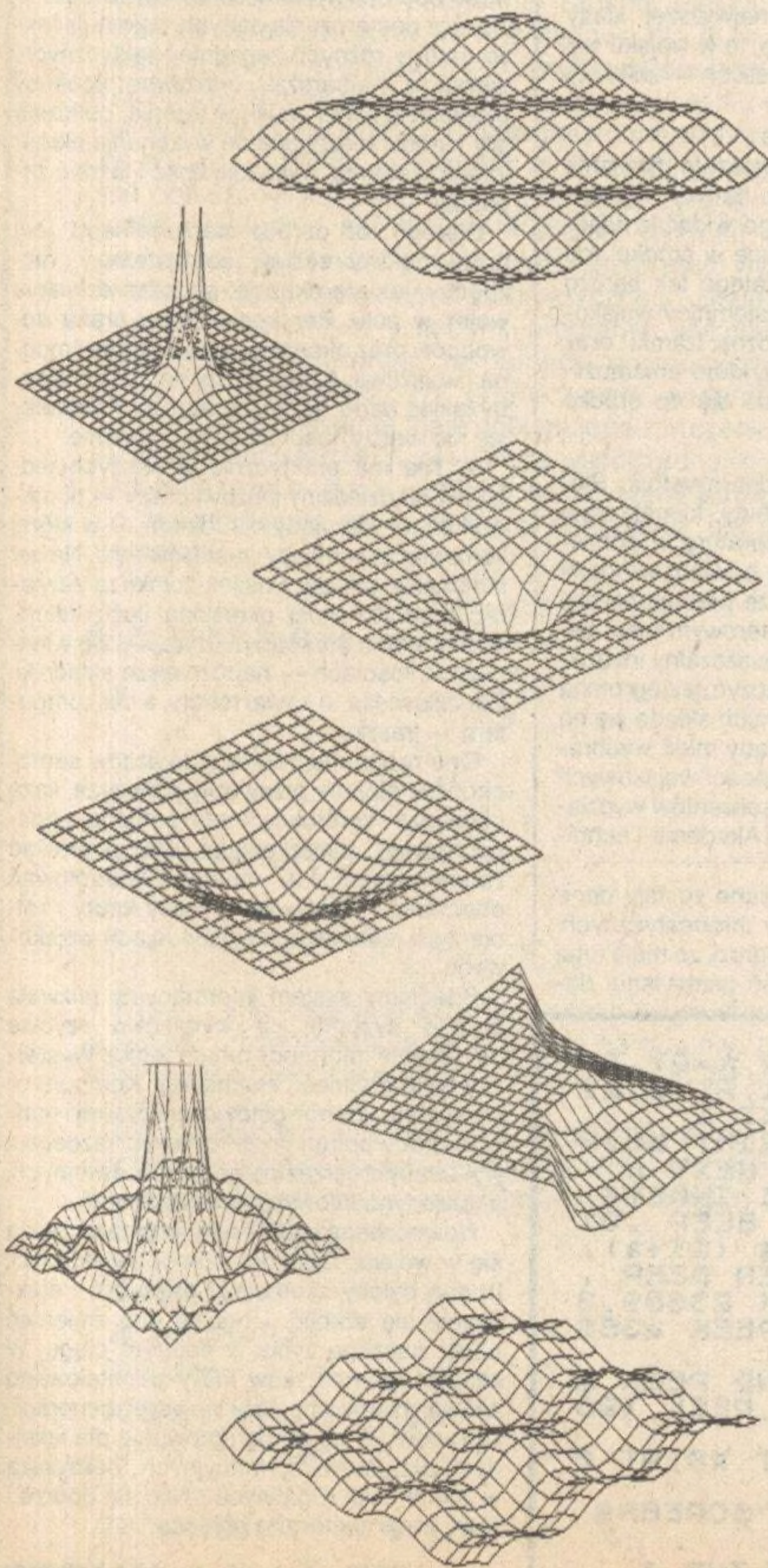
40 REM *****
45 REM grafika trojwymiarowa
47 REM *****
50 BORDER 0:INK 0,1:INK 1,26:INK 2,21:INK
K 3,17:MODE 1:DEG
60 sc=18:tilt=0.45:mag=400/sc:da=43:hlr=
7:vex=10
70 DIM z(sc+1,sc+1),scred(sc+1,sc+1,2)
80 PRINT" obliczanie wspolrzecznych baz
owych"
90 FOR x=sc TO 0 STEP -1:LOCATE 19,12:PE
N 1:PRINT x:LOCATE 15,25:PEN 2:PRINT "PR
DSZE CZEKAC":FOR y=0 TO sc
100 a=ATN((x+1)/(y+1)):IF SGN(a)=-1 THEN
a=180-a
110 r=SQR((x+1)^2+(y+1)^2)
120 scred(x,y,1)=mag*r*COS(da+a)
130 scred(x,y,2)=mag*r*SIN(da+a)*tilt
140 NEXT y,x
150 REM
160 WHILE 1:MODE 1:BORDER 0
170 ERASE z:DIM z(sc+1,sc+1)
180 PEN 2:LOCATE 9,2:PRINT CHR$(24);" GR
AFIKA TROJWYMIAROWA ";CHR$(24)
190 INK 1,0:WINDOW 11,30,4,21:PAPER 1:CL
S:WINDOW 12,29,5,20:PAPER 0:CLS:INK 1,26
200 PRINT:PEN 3
210 PRINT" WYBIERZ FUNKCJE"
220 PRINT
230 PRINT
240 PRINT:PEN 1
250 PRINT" 1-czarna dziura"
260 PRINT" 2-biala dziura"
270 PRINT" 3-falowanie"
280 PRINT" 4-uderzenia"
290 PRINT" 5-plusk"
300 PRINT" 6-zalamania"
310 PRINT" 7-wzgorze"
320 PRINT" 8-falda"
330 INK 1,26:WHILE INKEY$<>"":WEND
340 j=0:WHILE j<1 OR j>8 :j=VAL(INKEY$):
WEND
350 PEN 3:PRINT:PRINT" obliczenia...":PE
N 1
360 ON j GOSUB 610,670,710,760,810,870,1
080,1130
370 REM
380 MODE 2:DRAW 639,0,1:DRAWR 0,399:DRAW
0,399:DRAW 0,0
390 BORDER 10 :ORIGIN 320,4,1,638,2,396
400 REM
410 FOR x=sc TO 0 STEP -1:FOR y=sc TO 0
STEP -1
420 sx0=scred(x,y,1):sy0=scred(x,y,2)+vex*
z(x,y)
430 sx1=scred(x+1,y,1):sy1=scred(x+1,y,2)+
vex*z(x+1,y)
440 sx2=scred(x,y+1,1):sy2=scred(x,y+1,2)+
vex*z(x,y+1)
450 IF x<>sc THEN GOSUB 520
460 IF y<>sc THEN GOSUB 560
470 NEXT y,x
480 WINDOW 1,1,1,25:PRINT CHR$(24);"naci
sni; dowolny klawisz ";CHR$(24);
490 WHILE INKEY$ <> "":WEND:CALL &BB18
500 WEND
510 REM
520 PLOT 1000,1000,0
530 FOR n=1 TO hlr:MOVE sx0,sy0-n*2:DRAW
sx1,sy1-n*2:NEXT
540 MOVE sx0,sy0:DRAW sx1,sy1,1
550 RETURN
560 PLOT 1000,1000,0
570 FOR n=1 TO hlr:MOVE sx0,sy0-2*n:DRAW
sx2,sy2-n*2:NEXT
580 MOVE sx0,sy0:DRAW sx2,sy2,1
590 RETURN
600 REM
610 REM
620 FOR x=0 TO sc:FOR y=0 TO sc
630 z(x,y)=-((20/((x-sc/2+0.25)^2+(y-sc/
2+0.25)^2)))
640 NEXT y,x
650 RETURN
660 REM
670 FOR x=0 TO sc:FOR y=0 TO sc
680 z(x,y)=(15/((x-sc/2+0.25)^2+(y-sc/2+
0.25)^2))
690 NEXT y,x
700 RETURN
710 REM
720 FOR x=0 TO sc:FOR y=0 TO sc
730 z(x,y)=COS((x+y)*600/sc)
740 NEXT y,x
750 RETURN
760 REM
770 FOR x=0 TO sc:FOR y=0 TO sc
780 z(x,y)=COS(x*1000/sc)+SIN(y*1000/sc)
/90 NEXT y,x
800 RETURN
810 REM
820 FOR x=0 TO sc:FOR y=0 TO sc
830 q=(x-sc/2)^2+(y-sc/2)^2
840 z(x,y)=COS(q*8)/(0.01*(q+0.1))
850 NEXT y,x
860 RETURN
870 REM
880 RANDOMIZE TIME:FOR i=1 TO 35
890 s1=RND*3+1:s2=RND*3+1
900 IF s2=s1 THEN 890
910 d1=RND*sc:d2=RND*sc
920 d0=d1:ON s1 GOSUB 1040,1050,1060,107
0:ec1(0)=a:ec1(1)=b
930 d0=d2:ON s2 GOSUB 1040,1050,1060,107
0:ec2(0)=a:ec2(1)=b
940 IF ec1(0)=ec2(0) AND ec1(1)=ec2(1) T
HEN 910
950 IF ec1(0)>ec2(0) THEN t0=ec1(0):t1=
ec1(1):ec1(0)=ec2(0):ec1(1)=ec2(1):ec2(0)
=t0:ec2(1)=t1
960 xd=ec2(0)-ec1(0):IF xd=0 THEN m=0:GO
TO 980
970 m=(ec2(1)-ec1(1))/xd
980 c=ec1(1)-m*(ec1(0))

```

```

990 d=ROUND(RND)*2-1
1000 FOR x=0 TO sc:FOR y=0 TO sc
1010 IF y>m*x+c THEN z(x,y)=z(x,y)+d
1020 NEXT y,x,i
1030 RETURN
1040 a=d0:b=0:RETURN
1050 a=d0:b=sc:RETURN
1060 a=0:b=d0:RETURN
1070 a=sc:b=d0:RETURN
1080 REM
1090 FOR x=0 TO sc:FOR y=0 TO sc
1100 z(x,y)=MAX((0.4*sc-SQR((x-sc/2)^2+(y-sc/2)^2)),0)
1110 NEXT y,x
1120 RETURN
1130 REM
1140 FOR x=0 TO sc:FOR y=0 TO sc
1150 q=(x-sc/2)^3-(y-sc/2)^3-(sc/6)^3
1160 z(x,y)=MAX(ABS(q)^(1/3)*SGN(q),0)
1170 NEXT y,x
1180 RETURN

```



SPECTRUM

Spectrum pomoże dobrze gospodarować materiałem

Jan RUBINOWICZ

```

100 FOR n=65368 TO 65383: READ
bt: POKE n, bt: NEXT n: RESTORE
DATA 0,72,80,96,64,192,126,0,0,
126,66,66,66,66,126,0: REM ab=AB
110 CLS: INPUT "DŁUGOSC MATERI
AAU (3 do 32) "; dlm
120 IF dlm>32 OR dlm<3 OR dlm<>
INT dlm THEN BEEP 1,-10: GO TO 1
10
130 INPUT "SZTUK MATERIARAU (3 d
o 22) "; sztm
140 IF sztm>22 OR sztm<3 OR szt
m<>INT sztm THEN BEEP 1,-10: GO
TO 130
150 DIM m$(sztm,dlm): DIM r(szt
m)
160 PRINT AT 5,11; FLASH 1; BRI
GHT 1;" CZEKAJ ! "; FOR n=1 TO s
ztm: BEEP .03,26: FOR m=1 TO dlm
: LET m$(n,m)="B": NEXT m: NEXT
n: CLS
170 FOR n=1 TO sztm: LET r(n)=d
lm: NEXT n
180 LET k=1
190 INPUT "DŁUGOSC ELEMENTU "; (
CHR$(64+k)); " (MAX. "; (dlm); "
"; dle
200 IF dle>dlm OR dle<1 OR dle<
>INT dle THEN BEEP .5,2: GO TO 1
90
210 INPUT "SZTUK ELEMENTU "; (CH
R$(64+k)); " "; szte
220 IF szte<1 OR szte<>INT szte
THEN BEEP .5,2: GO TO 210
230 DIM e$(dle)
240 FOR n=2 TO dle-1: LET e$(n)
=CHR$(64+k): NEXT n: LET e$(1)=
"<": LET e$(dle)=">"
250 FOR n=1 TO sztm: IF r(n)>=d
le AND szte>0 THEN LET m$(n,1+dl
m-r(n)) TO dle+dlm-r(n))=e$: LET
r(n)=r(n)-dle: LET szte=szte-1:
GO TO 250
260 NEXT n
270 IF szte>0 THEN BEEP 1,-10:
PRINT AT 3,5; BRIGHT 1; FLASH 1;
" ZABRAKAO MATERIARAU ! "; LET sz
tm=sztm+1: GO TO 140
280 PRINT AT 0,0; FOR n=1 TO s
ztm: BEEP .01,n: PRINT TAB (16-d
lm/2); PAPER 6; BRIGHT 1;m$(n):
NEXT n
290 LET reszta=0
300 FOR n=1 TO sztm: LET reszta
=reszta+r(n): NEXT n: PRINT #0;A
T 0,0;" RESZTA = ";reszta;AT 1,0
;" WYKORZYSTANIE MATERIARAU ";.01
*INT (10000*(dlm*sztm-reszta)/(d
lm*sztm)); "%
305 IF reszta=0 THEN PAUSE 0: G
O TO 100
310 PAUSE 300: LET k=k+1: GO TO
190
800 CLS: PRINT " "
Niniejszy program pozwala na
optymalny rozkroj materialow
o dlugosci do 32 jednostek
na krotsze odcinki z wykorzy
staniem pozostalych odpadow."
810 PAUSE 0: GO TO 100
900 SAVE "Rozkroj" LINE 800: BE
EP .2,6: CLS: PRINT "Verify:
Rozkroj": VERIFY "": GO TO 100

```

Informatyka w wojsku

Wciśnięcie właściwych cyfr zamka szyfrowego sprawia, że ciężkie, obite blachą drzwi, otwierają się automatycznie. Do środka, mogą dostać się tylko ci, którzy znają ten szyfr i... obowiązuje tu język. Odmienny od tych, których uczą w szkołach, rzecz można wyjątkowy — język komputerów.

To prawda już dziś banalna. Współczesna nauka i technika odmieniły nasze życie. Nigdy jeszcze w historii człowiek nie wymyślił tak wiele, w tak krótkim czasie. Dziś każdy dzień nadaje tym słowom szczególny sens. Nie można sobie wyobrazić obecnie żadnej dziedziny ludzkiej aktywności bez najnowocześniejszych osiągnięć współczesnej techniki czy elektroniki. Dlatego trzeba, na co dzień, być z nią za pan brat.

Ba, ale czy w naszych warunkach jest to obecnie możliwe? Zwłaszcza, jeśli na te nowości elektroniki patrzy się przez pryzmat cen komputerów i oprzyrządowania do nich. To prawda. W cywilu wciąż jeszcze nie dla wszystkich jest ten sprzęt dostępny. A jak jest w wojsku?

Utechnicznienie, komputeryzacja, postęp, racjonalizacja. W armii ich znaczenie jest trudne do przecenienia. Nie jest dziś dla nikogo tajemnicą, że każda nowinka techniczna, zanim zostanie wprzęgnięta na usługi ludności, trafia najpierw pod wojskowy kapelus. Poza tym współczesne uzbrojenie, którym posługują się codziennie żołnierze, to skomplikowany, nafaszerowany elektroniką, jak dobry placek rodzinami, sprzęt.

Przykład pierwszy z brzegu. We współczesnym czołgu, albo armato-haubicy, najtrudniejsze, najbardziej czasochłonne prace, wykonuje elektronika. Komputer, na spółkę z laserem, w kilka sekund ustala odległości do celów i podadzą właściwe nastawy do prowadzenia celnego ognia. Specjalne urządzenia samoczynnie załadują działo i wyrzucą łuskę po wystrzale, potrafią

ostrzegać o zagrożeniu bronią masowego rażenia, same — jeśli zajdzie taka potrzeba — uruchomią system gaszenia ognia w wozie bojowym.

Dlatego właśnie technika nie może mieć dla żadnego żołnierza tajemnic. Podczas służby wojskowej młodzi ludzie poznają jej tajniki i uczą innych posługiwania się nią. Bo nawet najnowocześniejszy sprzęt pozostanie jedynie kupą żelastwa, jeśli nie będą go obsługiwać specjaliści najwyższej klasy. Rozumieć i znać technikę, to w wojsku wymaganie dotyczące wszystkich — żołnierzy i dowódców.

Właśnie tu, za obitymi żelazem drzwiami, w ośrodku przetwarzania danych Pomorskiego Okręgu Wojskowego, widać to najlepiej. Wszystko, co się dzieje w środku, ma ściśle tajny charakter. Dlatego tak bardzo dba się tutaj o ochronę tajemnicy wojskowej. Wszędzie elektroniczne zamki oraz przeróżne zabezpieczenia, które uniemożliwiają praktycznie dostanie się do środka komuś niepowołanemu.

I czystość. Niewiarygodna czystość. Pióciennie ochraniacze na buty, klimatyzacja zapewniająca stałą temperaturę w pomieszczeniach. Kurz i pył są bowiem wrogami komputerów. Zwłaszcza że pracują bardzo długo. W ośrodku komputerowym mają zatrudnienie przez cały, dopuszczalny instrukcją, czas. Wystarczy zobaczyć, jak ogromna ilość dysków magnetycznych składa się na tutejszy bank informacji, aby mieć wyobrażenie o rozmachu działalności wojskowych elektroników, głównie absolwentów Wydziału cybernetyki Wojskowej Akademii Technicznej.

Na tych dyskach zapisane zostały dane do wszystkich systemów informatycznych. Chor. Andrzej Rohuka twierdzi, że mają tutaj nieograniczoną możliwość przesyłania da-

nych za pomocą urządzeń transmisji danych do bardzo wielu abonentów. I to z każdej dziedziny życia i pracy żołnierzy okręgu. Od racjonalnego gospodarowania poczynając, a na kontroli podzespołów i części zamiennych kończąc.

Bo komputery mogą dosłownie wszystko. W ich pamięci zapisano dane umożliwiające efektywne kierowanie i dowodzenie wojskami w koszarach i w polu. Wystarczy „naciśnąć guzik”, a komputer równie łatwo wydrukuje plakat zespołu młodzieżowego, jak udzieli aktualnych informacji o wyposażeniu i uzbrojeniu pododdziału z najdalej położonego garnizonu. Ba, jeśli jest taka potrzeba, może być ona wydrukowana z szybkością np. tysiąca wierszy na minutę i dostarczona w tej formie adresatowi.

W ośrodku obsługujący komputery specjaliści i specjalistki twierdzą, że jedynej rzeczy, której jeszcze nie potrafią robić, to... gotować. Ale za książkę kucharską mogą służyć znakomicie. Ale poważnie.

— Komputery u nas ciężko pracują — mówi operator systemu informatycznego. — Oprócz dostarczania danych, dokonują także analizy różnych zagadnień taktycznych, wskazują najbardziej efektywne sposoby przemieszczania wojsk w terenie, obliczają siły i środki niezbędne do wykonania określonego zadania, wskazują ilość i jakość zapasów.

Dlatego też oprócz stacjonarnego, jest także połowy zestaw komputerowy, niezbędny, jak się okazuje, podczas działania wojsk w polu. Bez komputerów praca dowódców oraz oficerów sztabu byłaby żmudna i uciążliwa. A tak? Życzy sobie szef służby jakieś dane — proszę bardzo! Komputer się nie męczy, jest na każde polecenie.

— Nie ma praktycznie w naszych jednostkach dziedziny służby i pracy — tłumaczył oficer kierujący ośrodkiem — w której komputer nie miałby zastosowania. Nawet jadłospisy układa. Posiłek żołnierza zawierać przecież musi określoną ilość kalorii, odpowiednio muszą być użyte — i we właściwych ilościach — najróżniejsze składniki. Dla człowieka to kawał roboty, a dla komputera — fraszka.

One programują trasy przejazdów samochodów, aby nie przewoziły powietrza, liczą pieniądze i kontrolują konta, pomagają gospodarować mieszkaniami oraz kierować zaopatrzeniem, a nawet... kontrolować obecność i efektywność pracy kadry i żołnierzy w jednostkach i instytucjach wojskowych.

Specjalny system informacyjny pozwala służbie dyżurnej na wyjątkowo szybkie otrzymanie informacji na ten temat. Wystarczy tylko podnieść słuchawkę. Komputerowy system kontroli gotowości bojowej i nadzoru pracy potrafi także rozliczyć każdego z wykonanych przez niego zadań dziennych, a także tygodniowych i miesięcznych.

Nowoczesna technika mocno zakorzeniła się w wojsku. Dziś, bez pomocy elektroniki, trudno byłoby skutecznie dowodzić i efektywnie się szkolić. I będzie ona zmieniać obraz naszego życia w dalszym ciągu. W zespole informatyków POW zainstalowano znowu mikrokomputery nowszej generacji. Opracowuje się oprogramowanie dla kolejnych systemów informacyjnych. Elektronika w żołnierskiej rogatywce czuje się dobrze i służy armii skuteczną pomocą.

Włodzimierz KALETA

SPECTRUM

Semigrafika

Jan RUBINOWICZ

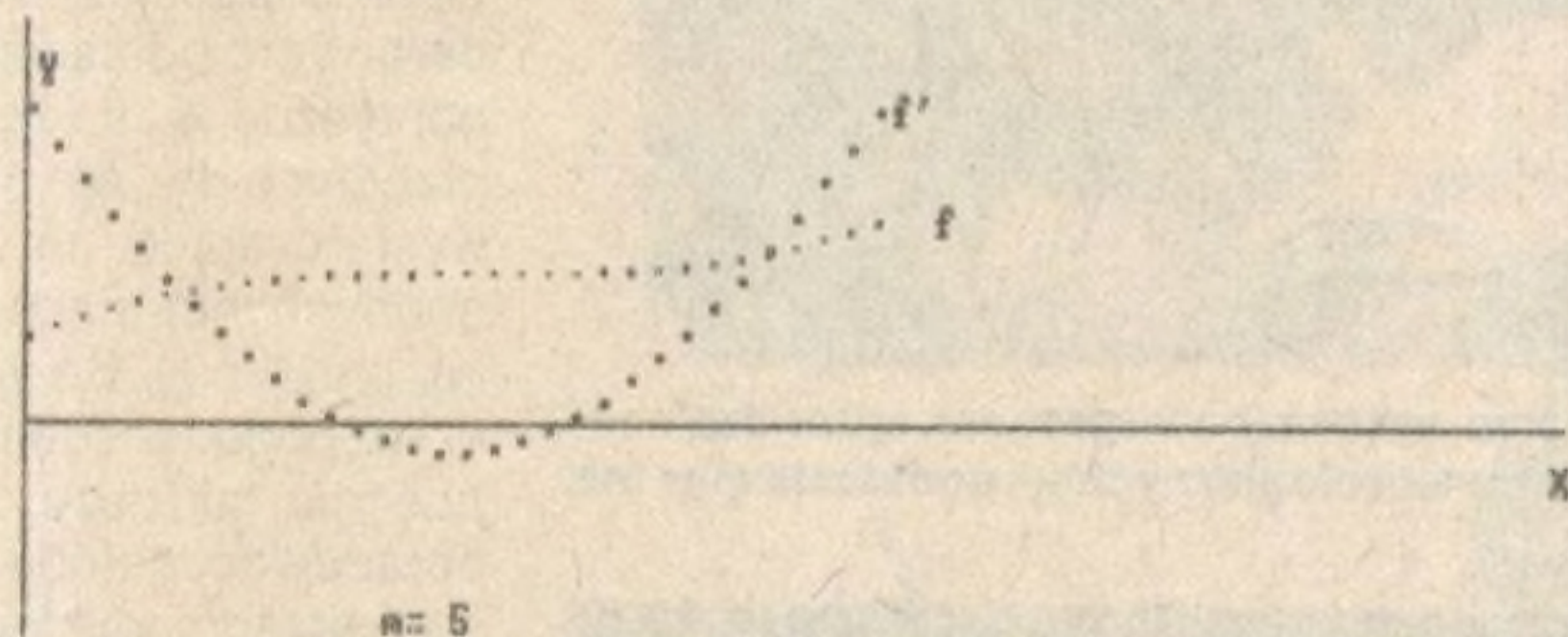
```
1 REM SEMIGRAFIKA
2 BORDER 6: CLS: FOR n=97 TO
3 112: PRINT #0; BRIGHT 1; AT 0,2*
4 (n-97);CHR$(31+n);: NEXT n
5 FOR n=97 TO 112: PRINT #0;A
6 T 1,2*(n-97);CHR$(n);: NEXT n
7 10 PAUSE 0: LET a=CODE INKEY$:
8 IF a>95 AND a<113 THEN BEEP .01
9 ,26: PRINT BRIGHT 1;CHR$(31+a);
10 15 IF a=12 OR a=48 THEN BEEP .
11 02,-10: PRINT AT 24-PEEK 23689,3
12 -PEEK 23688;" ";AT 24-PEEK 2368
13 9,32-PEEK 23688;
14 17 IF PEEK 23688=33 AND PEEK 2
15 3689<24 THEN POKE 23689,PEEK (23
16 689)+1: POKE 23688,1
17 20 IF a=115 THEN PRINT #0;AT 0
18 ,0;"
19 ": SAVE "semigrafik"SCREEN$
20 25 GO TO 10
21 30 SAVE "SEMIGRAFIK" LINE 1
```

Funkcja i jej pochodna

Zdzisław ŁUKASZEWICZ

Niejednokrotnie każdy z Czytelników „IKS-a” rozwiązywał bądź też będzie rozwiązywał w przyszłości zadania z matematyki, które zaczynają się od słów: zbadaj przebieg funkcji. Prezentowany poniżej program, napisany w języku BASIC na mikrokomputer AMSTRAD pomoże w rozwiązaniu tego typu zadań. Program ten umożliwia wykreślenie funkcji pierwotnej i jej pierwszej pochodnej. Aby otrzymać te wykresy należy przed uruchomieniem programu w linii o numerze 80 wpisać postać funkcji pierwotnej. Daną wejściową jest liczba m , która ze względu na przyjętą metodę wyznaczania wartości pochodnej funkcji jest liczbą naturalną. Należy w tym miejscu zaznaczyć, że dokładność wyznaczenia wartości pochodnej zadanej funkcji rośnie wraz ze wzrostem liczby m . Przykładowo dla $m = 2$ uzyskamy $n = 21$ $m = 4$, a dla $m = 5$ $n = 32$ wartości funkcji i jej pochodnej, trzeba o tym pamiętać przy skalowaniu wykresu. W programie wartości funkcji pierwotnej zawiera tablica $a(i)$, zaś wartości jej pochodnej tablica $c(i)$. Do programu dołączony jest podprogram WYKRES. Daje on możliwość wykonania na ekranie monitora wykresu funkcji pierwotnej i jej pierwszej pochodnej. Jest on mało wygodny w użyciu, gdyż nie zawiera automatycznego skalowania, bardziej wybrednym polecam programy wykonywania wykresów prezentowane na łamach ogólnodostępnych pism informatycznych. W przytoczonym podprogramie zmienne do skalowania mają następujące znaczenie — sk — skala współrzędnej Y, sx — skala współrzędnej X, sp i sy określają rozmiar punktu, którym wykonywany jest wykres, co ma znaczenie ze względu na nakładanie się dwóch wykresów.

sk=,sx=,sp=,sy=? 50,10,3,3
sk=,sx=,sp=,sy=? 50,10,3,3



W celu zilustrowania efektu, jaki można uzyskać posługując się prezentowanym programem, na rysunku pokazano wykres funkcji $F(x) = \text{PI} \cdot x + \text{COS}(\text{PI} \cdot x)$ dla $m = 5$.

Metoda wyznaczania wartości pochodnej funkcji, którą posłużyłem się przy opracowywaniu przedstawionego programu, oparta jest na teorii szeregów ortogonalnych.

Życzę przyjemnej zabawy.

```
10 MODE 2
20 INPUT "m=";m
30 n=2^m
40 DIM a(n+1),c(n+1),b(n+1)
50 FOR i=1 TO n+1
60 x=(i-1)/n
70 '***** W linii 80
```

```
wpisz postac funkcji *****
90 a(i)=PI*x+COS(PI*x)
90 NEXT i
100 FOR i=1 TO n:PRINT a(i);:NEXT i
110 FOR i=1 TO n
120 b(i)=((a(i)+a(i+1))/2)-a(i)
130 NEXT i
140 z=2^(m+1)
150 g=2^(m+2)
160 FOR i=1 TO n
170 FOR j=1 TO n
180 IF j>i THEN GOTO 240
190 IF i=j THEN 220
200 c(i)=c(i)+((-1)^(i+j))*g*b(j)
210 GOTO 230
220 c(i)=c(i)+((-1)^(i+j))*z*b(j)
230 NEXT j
240 NEXT i
250 FOR i=1 TO n
260 PRINT c(i);
270 NEXT i
280 FOR k=1 TO 5000
290 NEXT k
300 GOSUB 500
310 GOSUB 700
320 GOSUB 570
330 GOTO 330
490 '***** PODPROGRAM WYKRES *****
500 CLS
501 d=1
510 FOR i=20 TO 580
520 PLOT i,175
530 NEXT i
540 FOR i=20 TO 350
550 PLOT 20,i
560 NEXT i
570 INPUT "sk=,sx=,sp=,sy=";sk,sx,sp,sy
580 FOR i=0 TO n-1
590 FOR j=1 TO sp
600 FOR k=1 TO sy
610 PLOT sx*i+20+j,sk*a(i+1)+160+k
620 NEXT k
630 NEXT j
640 NEXT i
650 RETURN
700 FOR i=1 TO n
710 a(i)=c(i)
720 NEXT i
730 RETURN
```

Jak jest pogoda? Przeważnie widzimy sami. Jednak chyba najbardziej interesuje nas, jaka będzie pogoda? A ponieważ jest ona bardzo zmienna i nie mamy na jej zmianę żadnego wpływu, przeto staramy się przewidywać — prognozować. Dawniej zajmowali się tym wróżbici. Gdy udało się im przewidzieć stan pogody — potwierdzali tym samym swoją moc magiczną. Jeśli nie, wtedy zrzucali winę na moce nadprzyrodzone.

A dzisiaj? Niekiedy odwołujemy się jeszcze do wróżb, np. górali, albo „czujemy” pogodę w kościach. Najczęściej jednak słuchamy radia, patrzymy w telewizor, ufając, że nazajutrz nie musimy brać parasola do pracy, bądź wyjmować akumulatora z samochodu. Coraz bardziej ufamy prognozom naukowym i jest już niewielu ludzi, którzy na podstawie zjawisk zachodzących w atmosferze starają się na własny użytek przewidzieć pogodę. Ale też obrywa się synoptykom za niesprawdzone prognozy, bo takie się zdarzają — niestety. Mimo bardzo dokładnych i skomplikowanych metod pomiaru i określania pogody. Przyjrzyjmy się zatem, jak się to robi w świecie i u nas w kraju. Komu i czemu służą systemy meteorologiczne?

METEOROLOGIA — część geofizyki. Nauka o atmosferze, jej budowie, właściwościach i przebiegających w niej procesach fizycznych. Dużą część meteorologii, poświęconą klimatowi, wydziela się w samodzielny dyscyplinę nauki — klimatologię.

Z prognoz pogody najczęściej korzystają: lotnictwo, żegluga morska, rolnictwo oraz wiele przedsiębiorstw takich jak: stocznie, porty, bazy przeładunkowe, oczyszczania miast. Przykładów, gdy zlekceważono ostrzeżenia synoptyków o zbliżających się burzach, opadach, mrozach, wiatrach, mgłach lub innych zjawiskach atmosferycznych można podać bardzo dużo. Z naszych, polskich doświadczeń, najbardziej pamiętnym będzie tzw. zima stulecia (choć wcale nią nie była, ale z uwagi na straty, tak ją określono) w roku 1978/79. Ściślej biorąc noc sylwestrowa z 31 na 1 stycznia, gdy nagły powiew chłodu sparaliżował większość kraju. Albo ostatnie zaskoczenie huraganem w Wielkiej Brytanii. Mimo ostrzeżeń francuskich służb meteorologicznych i wskutek zlekceważenia własnej informacji z wybrzeża, kataklizm pokonał wszystkie systemy zabezpieczeń. A właśnie Wielka Brytania należy do państw szczytujących się posiadaniem bardzo nowoczesnych systemów meteorologicznych. Jeśli jesteśmy już przy kraju wyspiarzy, przypomnijmy przykład właściwego wykorzystania pogody podczas inwazji w Normandii. Pewien major ze służby meteo przewidział niewielką poprawę pogody, co spowodowało zupełne zaskoczenie u Niemców. Prognoza ta bowiem, choć bardzo ryzykowna, sprawdziła się wtedy.

To tylko znane i głośne przykłady, kiedy człowiekowi udało się lub też nie, zapobiec skutkom żywiołów spowodowanych warunkami atmosferycznymi. Niezaprzeczalne jest jednak to, że bardzo dobra znajomość stanu pogody i coraz doskonalsze opraco-

wywanie prognoz po prostu się opłaca. Już dziś informacje meteorologiczne są towarem. W Polsce jeszcze daleko do stanu, jaki jest na Zachodzie, ale wszystko wskazuje na nieuchronność tego zjawiska. Przyjrzyjmy się zatem technicznemu aspektowi opracowywania informacji o pogodzie.

Instytut Meteorologii i Gospodarki Wodnej w Warszawie. Placówka naukowa, a zarazem instytucja zajmująca się sprawami pogody, o której codziennie słyszymy w komunikatach prasowych, radiowych i telewizyjnych. W pomieszczeniu synoptyków bezustannie trwa opracowywanie danych. Pochodzą one z trzech źródeł: pomiarów naziemnych, satelitarnych, radarów meteorologicznych. A więc informacje punktowe, globalne i przestrzenne w zasięgu radaru.

— *Podstawą do prognozy jest zebranie materiału obserwacyjnego* — wyjaśnia mgr inż. **Ryszard Klejnowski**, zastępca dyrektora IMGW do spraw Prognoz. — *I to co najmniej z całej półkuli północnej w czasie rzeczywistym. Jest właśnie godzina dziewiąta.*

O tej porze na całym świecie dokonuje obserwacji w klatkach pomiarowych około 8 tysięcy ludzi jednocześnie.

SŁUŻBA POGODY. Organizacja, której zadaniem jest zapewnienie informacji o bieżących i prognozowanych warunkach atmosferycznych całej gospodarce narodowej i obronie kraju. Służba pogody IMGW liczy w Polsce 62 stacje meteorologiczne oraz 8 biur: Białystok, Gdynia, Kraków, Poznań, Szczecin, Wrocław, Warszawa.

W ciągu 10 minut wszystkie dane z Polski przekazuje się natychmiast drogą teleksową do GTS (Globalny System Telekomunikacyjny) przez regionalne centrum meteorologiczne w Pradze. Przesyła się je także do sąsiadów w Moskwie i Berlinie. Praktycznie

POGODA Z KON

Marek



Uzyskaliśmy stosunkowo wysoką, bo 86 procent, sprawdzalność codziennych prognoz meteorologicznych — podkreśla mgr inż. Ryszard Klejnowski.



Na razie mapy pogody kreślą u nas jeszcze ludzie. Ale niedługo będą to robiły komputery.

(Foto: Jan Zelman)

już w 15 minut po pomiarze znane są wszystkie dane z całego świata. Co godzinę dokonuje się tych pomiarów. Informacje z Europy otrzymujemy co godzinę, a z USA co 3 godziny regularnie przez całą

Na stole głównego synoptyka, w tym przypadku był to **Jerzy Bednarek**, leżą wszystkie potrzebne dane, w tym dane satelitarne. Z satelity geostacyjnego METEOSAT-2 oraz orbitującego nad północną Europą. Oprócz tego liczne komunikaty ze stacji meteorologicznych, od sąsiadów, poprzez komunikaty satelitarne, tj. sprzed kilku godzin i kilku dni. Ale okazuje się to jest wystarczające. Im więcej bowiem danych, tym lepiej. Bo tym lepiej można przewidzieć pogodę. Jednak natłok informacji jest obecnie tak duży, że trudno je przetworzyć metodą ręczną. Niezbędne jest więc włączenie techniki komputerowej. Prawdę znają wszyscy meteorolodzy, ale nie wszyscy chcą. Jednak względy czysto finansowe powodują, że nie stać nas na najnowszy sprzęt.

Cały świat przechodzi już na automatyczne systemy gromadzenia i opracowania danych, ich selekcję, analizę, prognozowanie, a następnie dystrybucję informacji. Dziś mówi się już o fabrykach informacji dotyczących stanu atmosfery. Kto potrafi je wszystkie wykorzystać, ten będzie miał trafniejsze prognozy.

Zostanę więc jeszcze przez jakiś czas. Główny synoptyk kraju w IMGW opracowuje mapy synoptyczne po godzinie, co trwa około dwóch godzin. Wykreśla je na podstawie danych nanoszonych przez innych ludzi z nim współpracujących. Obszar obejmuje całą Europę, część Afryki, Azję i Atlantyk północny.



...k się robi mapę pogody dla prasy? Pokazał nam to Michał Jaworski, który tego dnia pełnił dyżur.

MPUTERA

SIENIAWSKI, Tadeusz BERLIŃSKI

...są wyniki
...dzinę do-
...macje z
...ze świata
...obę.

Obecnie już niewiele państw na świecie zajmuje się ręcznym wykreśleniem map synoptycznych. Właściwie wszystko robią za człowieka komputery. Potrafią one bardzo szybko wykreślać izobary, nanosić potrzeb-

...w tym
...znajdują
...zdjęcia
...onarnego
...NOAA.
...stacji ra-
...mapy sy-
...oraz po-
...szcze nie
...n wiary-
...synopty-
...przewi-
...nacji jest
...zerobić i
...dne staje
...owej. Te
...zy. Jed-
...dują, że

...automaty-
...pracowy-
...lizę oraz
...bucję in-
...ce infor-
...ry. I ten
...ać, może
...chwile w
...MGW. Na
...potrzebuje
...izobary,
...h na ma-
...pracują-
...opę, pół-
...po Gren-

Dane te koreluje na podstawie zdjęć satelitarnych. I natychmiast przesyła odbiorcom.

Rola synoptyka ogranicza się do znajomości sytuacji i zgodnie z tym ostrzegania, podawania prognoz natychmiastowych oraz średnio- i długoterminowych. Zapotrzebowanie na informacje meteorologiczne wzrasta. Głównie zaś na informacje dla lotnictwa i sił zbrojnych. To właśnie względy militarne, a na drugim miejscu potrzeby meteorologiczne tworzą siłę napędową do budowy superszybkich komputerów. Obecnie dochodzi jeszcze fizyka jądrowa.

Najszybszy komputer CRAY-1 pracuje w Światowym Centrum Meteorologicznych Prognoz w Waszyngtonie oraz Regionalnym Europejskim Centrum Prognoz Średnioterminowych w BRACKNELL (Wielka Brytania). Jest to wspólna inwestycja EWG. CRAY-1 musi wykonać 500 miliardów operacji arytmetycznych, aby postawić prognozę na 10 dni. Przy szybkości 80 mln operacji na sekundę trwa to 1 godzinę i 45 sekund. ODR-1305 robiłaby to 2 miesiące.

Światowa Organizacja Meteorologiczna (WMO), której członkiem jest także Polska, składa się z 3 podsystemów:

— Obserwacja. 8 tys. stacji synoptycznych, 1500 stacji aerologicznych, 350 radioliniiowych, 4 satelity geostacjonarne, 3—5 satelitów orbitujących, 6500 stacji morskich. Przekazuje na dobę 40 mln bajtów informacji.

— Telekomunikacja. Przesyła informacje przetworzone do centrów narodowych, regionalnych i światowych.

— Przetwarzanie. Analiza, diagnozowanie i prognozowanie pogody na podstawie napływających informacji.



Wyże, niżej, zjawiska i elementy meteorologiczne, to wszystko widać na mapie pogody. Kpt. Andrzej Waloryszak, synoptyk wojskowy z CBH wyjaśnia, jaka będzie nazajutrz pogoda.

ne dane bez pomocy człowieka. Synoptyk otrzymuje już gotowe przetworzone informacje. I na tej podstawie oraz opierając się na swoim doświadczeniu, znanych zjawiskach, mając dzięki temu przestrzenne wyobrażenie o stanie atmosfery — prognozuje. Najczęściej piórem świetlnym na ekranie monitora. Określa kierunki, przemieszczania mas powietrza i rozwoju sytuacji.

W WMO główną rolę odgrywają centra światowe połączone telekomunikacyjną magistralą światową. Zasadniczą prędkością jest obecnie 2400 bodów, a na niektórych kierunkach stosuje się już 9600 dzięki światłowodom i łączności satelitarnej. Do Warszawy na dobę docierać może 14 mln bajtów. Cóż z tego, skoro nie jesteśmy w stanie przetworzyć tej ilości informacji.

ŚWIATOWA SŁUŻBA POGODY. Sieć stacji meteorologicznych i aerologicznych oraz innych środków obserwacji (satelity, sondy) działających według globalnego programu oraz centra meteorologiczne. Sieć łączności do wymiany i przetwarzania danych. Program badań naukowych.

Światowe centra meteorologiczne: Moskwa, Waszyngton, Melbourne. IMGW jest krajowym centrum meteorologicznym. Połączone z centrami w: Moskwie, Pradze, Poczdamie. Przez Regionalną Zbiornicę Telekomunikacyjną w Pradze połączone z Regionalnym Centrum Meteorologicznym w Offenbach.

W IMGW wkracza, powoli co prawda, ale jednak, najnowsza technika komputerowa do wspomaganie pracy synoptyka. W gabinecie dyrektora Klejnowskiego możemy zobaczyć, na razie program próbny, komputerowego zobrazowania danych, przetworzonych na podstawie napływających informacji ze świata i kraju. Natomiast w sali głównego synoptyka już pracuje rejestrator obrazów satelitarnych na barwnym monitorze telewizyjnym. Dzięki współpracy z Politechniką Wrocławską zbudowano i zainstalowano w 1985 roku w Krakowie pierwszy uniwersalny zestaw do odbioru satelitarnych danych meteorologicznych o podwyższonej zdolności rozdzielczej z satelitów TIROS, NOAA wraz z procesorem obrazów pogody. W ten sposób uzyskano możliwości odbioru obrazu satelitarnego, pozwalającego na precyzyjniejsze określenie temperatury powierzchni lądów, mórz i chmur.

Jarosław Mojzych na monitorze komputera IBM AT prezentuje możliwości barwnego przedstawiania sytuacji meteorologicznej pochodzącej z satelity geostacjonarnego METEOSAT. Są to obrazy temperatury, wykonane techniką podczerwieni, a więc bez względu na porę dnia. Obrazy docierają co 1 godzinę systemem fotografii, a na komputerze co 30 minut. Komputer animuje te zmiany, co pozwala zobrazować ruch mas powietrza.

Obok Michał Jaworski pokazuje, jak się robi na komputerze mapki pogody dla prasy codziennej. Mapa taka powstaje w niespełną godzinę. Jedną z nich, aktualną w dniu oddawania „IKS-a” prezentujemy na pierwszej stronie pisma.

PROGNOZA POGODY. Sporządzanie naukowo uzasadnionych przewidywań o przyszłym stanie pogody. Sporządza się metodą analizy i prognozy makroprocesorów w atmosferze oraz warunków pogody na dużych obszarach za pomocą map synoptycznych i dodatkowych środków (diagramy aerologiczne, przekroje pionowe itd.).

Mgr inż. Ryszard Klejnow
IMGW: Mimo nietypowego w ubiegłym ro

FRONT ATMOSFERYCZNY. Strefa przejściowa lub teoretyczna powierzchnia rozdziału między dwiema masami powietrza w atmosferze.

NIŻ (cyklon) zaburzenie w atmosferze o obniżonym ciśnieniu atmosferycznym (minimalne ciśnienie w centrum) i cyrkulacji powietrza dokoła centrum. W lewo na półkuli północnej i w prawo na półkuli południowej.

WYŻ (antycyklon) wyż baryczny — obszar podwyższonego ciśnienia atmosferycznego, który na przyziemnych (dolnych) mapach synoptycznych ma kształt układów zamkniętych, w przybliżeniu koncentrycznych izobar o kształtach nieregularnych, lecz na ogół kolistych lub owalnych.

na początku bieżącego — przebiegu pogody, uzyskaliśmy stosunkowo wysoką sprawdzalność codziennych prognoz meteorologicznych — około 86 proc. W pełni wywiązałyśmy się z funkcji ostrzegania gospodarki narodowej. Nie zanotowano żadnej katastrofy komunikacyjnej, czy też strat gospodarczych związanych z brakiem ostrzeżenia prognostycznego.

Rok 1987 był przełomowym w zakresie modernizacji służby prognoz IMGW. Wprowadzono technikę mikrokomputerową. Pozwoliła na to samodzielność finansowa instytutu. Uzyskaliśmy 5 mld zł ze sprzedaży naszych usług. Pracujemy nad systemami przetwarzania i odbioru danych satelitarnych oraz metodyki ich wykorzystania w służbie prognoz. Przełomem stało się zastosowanie komputerów IBM PC. Dysponujemy unikatową aparaturą, wykonaną w kraju, pozwalającą na zastosowanie danych satelitarnych nie tylko w meteorologii, ale również w hydro- i oceanografii, ochronie środowiska, rolnictwie i leśnictwie.

— Zespół w składzie: Marek i Ludmiła Kozłowski (małżeństwo), Przemysław Ignatowicz oraz Stanisław Kozłowski od kwietnia 1987 roku zajmuje się opracowaniem systemu odbioru i opracowania informacji meteorologicznych zawartych w depeszach SYNOP, TEMP, GRID przy użyciu mikrokomputerów kompatybilnych z IBM PC/AT/XT — mówi dr Anna Olszowicz.

kierownik Centralnego Biura Prognoz w Warszawie.

Efekty już widać na monitorach. Komputer sam kreśli izobary, nanosi dane odbierane ze stacji pomiarowych. Barwy pozwalają na rozróżnianie istotnych elementów pogody.

— Najślabszym elementem systemu są sieci do transmisji danych między komputerami — dodaje dr Olszowicz. — Próbowa- liśmy sieć TRANS-NET i się nie sprawdziła. Obecnie robimy eksperymenty z siecią D-LINK, także amerykańską.

Zajrzyjmy jeszcze do meteorologów z Wojsk Obrony Powietrznej Kraju.

— Bardzo istotne jest przekazywanie informacji meteorologicznych odbiorcom, czyli na lotniska wojskowe — mówi mjr dr inż. Leonard Kaczanowski z dowództwa WOPK. — Sytuacje pogodowe zmieniają się z godziny na godzinę. Piloci wojskowi muszą mieć bardzo aktualne i wiarygodne informacje. Dlatego komputeryzujemy służbę. Warunkują to także prace międzynarodowe, skąd również czerpiemy informacje i dane.

Na przykład tuż po wojnie obowiązywał system przekazywania informacji alfabetem Morse'a i operatorzy wprost z radia nanosili dane na mapy. O błąd nietrudno, a dane nie były już powtarzane. Potem przeszło się na łączność dalekopisową. I w 1981 roku Europa zrezygnowała z łączności dalekopisowej na korzyść teletransmisji. Mówi się już poważnie o przejściu na łączność w systemie cyfrowym.

W wojsku przeważnie sporządza się prognozy krótkoterminowe. Gdy zbliża się termin lotów lub ćwiczeń, wówczas rozpoczyna się gorączka w służbie meteo. Zresztą tu pracuje się bez przerwy. Prognozy dla lotnictwa wojskowego muszą być aktualne w każdej chwili.

Elementy meteorologiczne. Charakterystyki stanu powietrza i niektórych procesów atmosferycznych: ciśnienie, temperatura, wilgotność, wiatr, zachmurzenie (ilość, rodzaj), widzialność, rodzaj i ilość opadów.

Zjawiska meteorologiczne. Są to: opady i mgły różnych typów, zamiecie, zjawiska natury elektrycznej takie jak: burze, błyskawice, zorze polarne, szkwały, burze pyłowe, wiry, trąby powietrzne, igły lodowe, gołoledź, pokrywa śnieżna.

Bywa i tak niekiedy, że za złą pogodę wini się meteorologów. Sypią się też gromy, gdy nad poligonem się przejaśni i mogłoby działać lotnictwo, ale służba meteo nie zezwała. Tylko że nikt nie rozumie, iż na odległych lotniskach pogoda jest zła zarówno do startu, jak i przede wszystkim do lądowania. No cóż, bywa i tak, że meteorologów dowódcy oceniają nie tyle za prognozę, ile za pogodę. Najczęściej też widzi się tych 10 proc. niesprawdzonych prognoz — stwierdza jeden z meteorologów wojskowych. — Na zasadzie czarnej owcy w stadzie.

A praca ta nie jest łatwa. Zajmuje się nią Służba Meteorologiczna WOPK. Informacje

odbierane są różnymi systemami i przekazywane do komputera MERA-60. Również z IMGW, Pragi, Moskwy. Komputer ten przyjmuje i rozsyła informacje do wszystkich lotnisk wojskowych w Polsce i sojuszniczych. Kiedyś wszystkie prace wykonywało 5 osób. Teraz komputer pracuje sam. Ppłk mgr inż. Maciej Ostrowski blokuje na chwilę komputer i ten po minucie emituje sygnał alarmowy. Natychmiast zgłasza się technik chor. Zbigniew Paćkowski, który usuwa przyczynę awarii. Możemy spytać o pogodę na dowolnym lotnisku w Europie. Wybieram z mapy kod lotniska w Neapolu — IY LIRN. Podpułkownik wypisuje daną na klawiaturze mikrokomputera sprzężonego z MERA-60. W niespełną minutę mamy dane za pośrednictwem biura regionalnego w Pradze: wiatr zachodni 7 węzłów, widzialność 8 km, zamglenie, zachmurzenie 3/8, stratocumulus 750 metrów. I dalej dane o przewidywanych zmianach.

— Ten komputer załatwia sprawy związane z telekomunikacją — mówi kierownik biura. — Obecnie pracujemy nad wdrożeniem do pracy plotera przy użyciu komputera IBM. Będzie on wpisywać na mapę te wszystkie dane, które dotąd zapisywali ludzie.

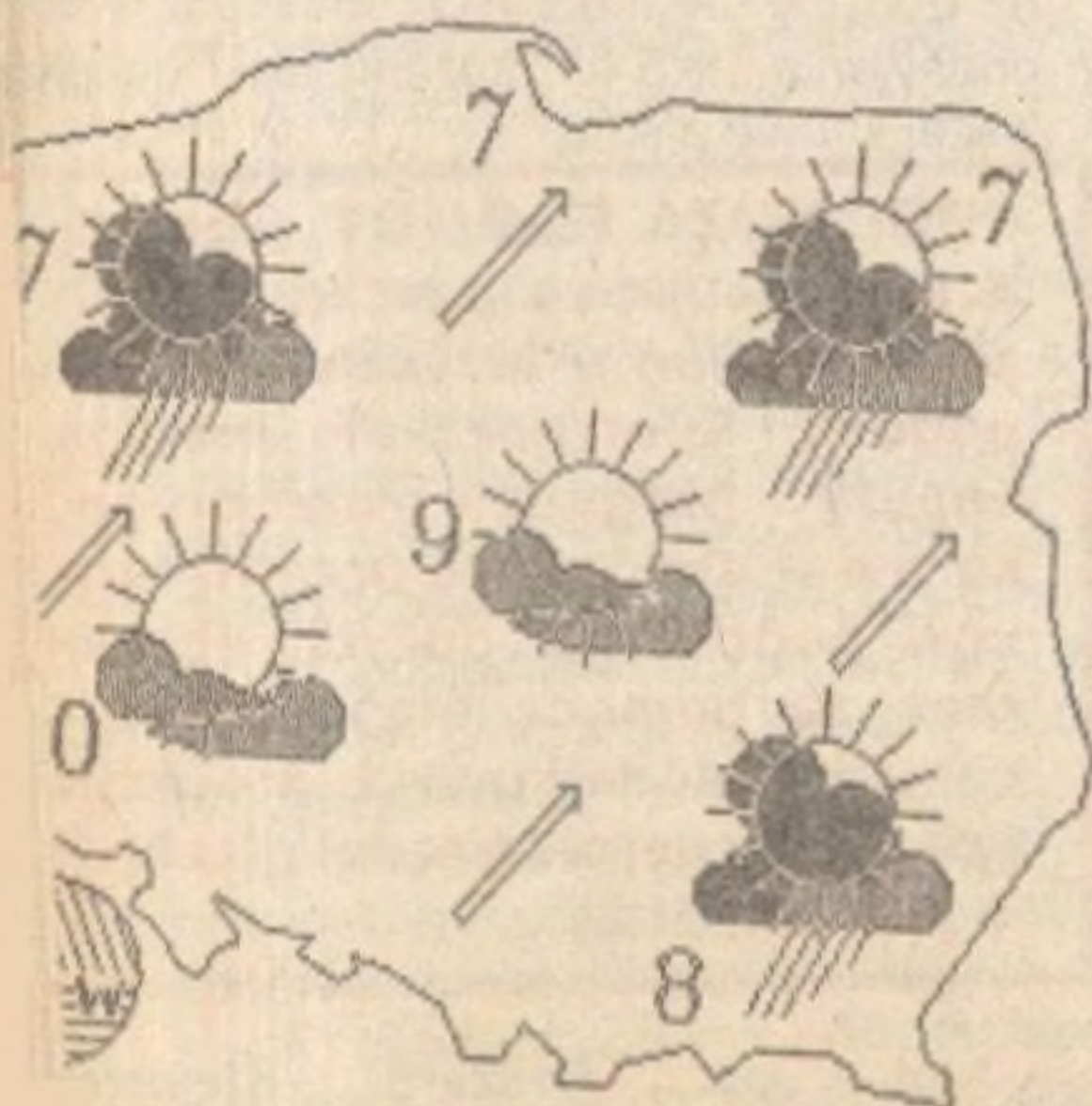
Także tutaj wykorzystuje się faksymilowe zdjęcia satelitarne i wiele innych informacji ze źródeł zagranicznych. Dyżurny synoptyk, ppłk Klemens Bartysiewicz, zaznacza na wykreślonej mapie synoptycznej. Wiemy więc o wiele więcej o stanie pogody i prognozie na najbliższych 10 godzin, aniżeli z telewizji. Możemy potwierdzić, że prognoza na następną dzień się sprawdziła.

Służba meteo opracowała i uruchomiła system NIMBUS, obsługujący wejścia i wyjścia: dalekopisowe, transmisji danych i urządzeń zewnętrznych. Zapewniło to rozpoznanie typu informacji (wg światowego standardu służby meteo), czasu nadawania, numeru stacji i następnie przesyłanie informacji do urządzeń zewnętrznych oraz mikrokomputerów. Zastosowano przy tym 15 wniosków racjonalizatorskich i nowatorskich. Szybko też wdrożono obsługi biur lotniskowych, nadające depesze do komputera. Ten bowiem bezlitośnie traktuje wszystkie niedokładności, a błędne informacje zwraca nadawcy, do czasu ich poprawienia. NIMBUS zastąpił pracę 10 ludzi, zmniejszył opóźnienie zobrazowania stanu pogody na lotniskach do zera. Dotąd opóźnienie wynosiło 15—20 minut. Pracuje się jednak nad tym, aby wspomagać proces opracowywania prognoz pogody bez udziału ludzi, formułować komunikaty na przeloty, ostrzeżenia o niebezpiecznych zjawiskach pogodowych.

Biuro Prognoz Lotniczych w Kansas City (USA) zastąpiło 9 biur regionalnych, daje prognozy na obszar 30 razy większy od Polski i pracuje z częstotliwością 3—4 razy większą. Zatrudnia połowę jego stanu osobowego. Jest więc 200 razy wydajniejszy. Zastosowano w nim 4 systemy komputerowe. 2 zewnętrzne (łączność i prognozy tornad) oraz do wspomaganie synoptyka i redagowania prognozy pogody.

A jaką pogodę będziemy mieli jeszcze tej kapryśnej zimy, jaka będzie wiosna? Wierzcie, to naprawdę nie zależy od meteorologów.

**Marek SIENIAWSKI
Tadeusz BERLIŃSKI**



Rozstawienie żołnierzy na warcie

Mieczysław SKONIECZNY

W drugim odcinku cyklu poświęconego anegdotom matematycznym, chciałbym przedstawić rozstawianie żołnierzy na warcie („LILAVATI” — anegdota nr 16). Treść historyjki jest następująca:

„Na kwadratowe podwórze zbrojowni wszedł porucznik z oddziałem żołnierzy w celu rozstawienia warty. Postawił wzdłuż każdej ściany 4 żołnierzy i się oddalił. Po chwili przybył kapitan i uważając, że warta nie jest dostateczna, umieścił wzdłuż każdej ściany 5 wartowników. Wreszcie na podwórzu zbrojowni zjawił się major i wzdłuż ściany umieścił 6 żołnierzy. Jakie było rozstawienie żołnierzy w 1, 2 i 3 przypadku, jeśli wszyscy trzej oficerowie rozporządzali tym samym oddziałem”.

Na ekranie mikrokomputera podwórze to zielony prostokąt, a budynek zbrojowni, to czarna ramka. Na podwórzu rozstawionych jest 8 ponumerowanych posterunków (budki wartownicze). Zasady gry są następujące:

1. Gra składa się z 3, kolejno po sobie następujących, etapów:
I — rozprawdzający: porucznik;
II — rozprawdzający: kapitan;
III — rozprawdzający: major;
2. Posterunki są ponumerowane od 1 do 8, a wartownicy mają nadane kody literowe od A do L (12 żołnierzy).
3. Maksymalnie na jednym posterunku można postawić 5 wartowników, a minimalnie dopuszcza się, że posterunek jest pusty.
4. Umieszczenie wartownika na wybranym posterunku polega na:

- podaniu numeru posterunku;
- podaniu kodu literowego żołnierza, którego chcemy rozstawić.
- Po podaniu numeru posterunku na pierwszym wolnym miejscu w tej budce wartowniczej pojawia się czerwony kwadrat.

5. Rozstawiać można wartownika z dwuszeregu, dopuszczalne jest przestawienie go z posterunku na inny posterunek.

6. Kolejny etap gry kończy się sukcesem, gdy grający rozmieści na posterunkach wszystkich wartowników z dwuszeregu, a liczba żołnierzy chroniących każdą ścianę będzie zgodna z wymaganiami tego etapu.

7. Użytkownik wygrywa grę, gdy zakończy sukcesem wszystkie 3 etapy (koniec pracy programu).

Wersja źródłowa programu jest następująca:

```
1 ranga$(1)=" Porucznik":ranga$(2)=" Kapitan":ranga$(3)=" Major"
5 DIM szereg(12):DIM budka(8,5):DIM suma(8)
10 BORDER 18:INK 0,18:INK 1,0:INK 2,26:INK 3,6:PAPER 0:CLS:GOSUB 9000
50 DATA 3,3,20,3,36,3,3,11,36,11,3,19,20,19,36,19
70 PAPER 0: CLS:BORDER 18:PEN 1
80 PRINT"Na podworze zbrojowni wszedl porucznik z oddzialem 12 zolnierzy. Rozstawil wzdłuż kazdej sciany
4 zolnierzy i oddalil sie po wykonaniu zadania."
90 PRINT:PRINT"Po chwili przybyl kapitan i uwazajac, ze warta nie jest dostateczna, rozstawil oddzial w ten
posob, ze wzdłuż kazdej sciany stalo 5 zolnierzy."
100 PRINT:PRINT"Wreszcie na podworzu zjawił sie major i wzdłuż kazdej sciany umiescil 6 zolnierzy."
110 PRINT:PRINT"Jakie bylo rozstawienie zolnierzy w 1,2 i 3 przypadku ?
130 PEN 2:LOCATE 1,25:PRINT"NAKISNIJ DOWOLNY KLAWISZ":c$="":IF INKEY=c$ THEN GOTO 130
300 etap=1
310 BORDER 0:RESTORE 50:PAPER 0: CLS: FOR c=1 TO 8:READ a,b:GOSUB 1000:NEXT c
315 PAPER 1:LOCATE 1,1:FOR l=1 TO 80:PRINT " ";NEXT l
320 LOCATE 1,1:PEN 2:PRINT"Etap:";USING "E";etap:LOCATE 8,1:PRINT"Rozprawdzajacy:";ranga$(etap)
330 GOSUB 2000
360 FOR t=1 TO 2000:NEXT t
370 PAPER 1:PEN 2:LOCATE 1,1:PRINT"Kazda sciana ma chronic ";USING "E";etap+3:PRINT" wartownikow"
380 ruch=1:FOR i=1 TO 12:szereg(i)=64+i:NEXT i
390 FOR i=1 TO 8:FOR j=1 TO 5:budka(i,j)=0:NEXT j:NEXT i
400 PAPER 0:PEN 2:LOCATE 9,4:PRINT"Ruch:";ruch
410 LOCATE 21,2:PAPER 1:PRINT"Podaj nr posterunku: ",nrb
420 IF nrb<1 OR nrb>8 GOTO 470
430 blad=0:FOR i=1 TO 5
440 IF budka(nrb,i)=0 THEN blad=1
450 NEXT i
460 IF blad=1 GOTO 490
470 LOCATE 1,2:PRINT"Blad. Wprowadz poprawny parametr. ":FOR T=1 TO 1500:NEXT T
480 GOTO 410
490 GOSUB 3000:LOCATE x1,y1:PEN 3:PRINT CHR$(143)
500 PEN 2:LOCATE 21,2:PRINT"Podaj kod wartownika: ",kod$
510 kodw=ASC(kod$):IF kodw<65 OR kodw>77 GOTO 560
520 blad=0:FOR i=1 TO 5
530 IF budka(nrb,i)=kodw THEN blad=1
540 NEXT i
550 IF blad=0 GOTO 580
560 LOCATE 1,2:PRINT"Blad. Wprowadz poprawny parametr. ":FOR T=1 TO 1500:NEXT T
570 GOTO 500
580 IF SZEREG(kodw-64)=0 GOTO 700
590 y2=11:FOR i=66 TO 76 STEP 2
600 IF kodw=i THEN y2=15
610 NEXT i
620 FOR i=1 TO 6
630 IF kodw=63+2*i OR kodw=64+2*i GOTO 650
640 NEXT i
650 PAPER 0:x2=12+3*i:FOR i=0 TO 2:LOCATE x2,y2+i:PRINT " ":NEXT i
660 szereg(kodw-64)=0
670 PEN 1:LOCATE x1,y1:PRINT kod$:budka(nrb,h)=kodw:GOTO 800
```

```

700 FOR i=1 TO 8:FOR j=1 TO 5
710 IF budka(i,j)=kodw GOTO 730
720 NEXT j:NEXT i
730 BUDKA(I,J)=0:XX1=4:IF i=2 OR i=7 THEN XX1=21
732 IF i=3 OR i=5 OR i=8 THEN XX1=37
734 YY1=4+j:IF i=4 OR i=5 THEN YY1=YY1+8
736 IF i=6 OR i=7 OR i=8 THEN YY1=YY1+16
740 LOCATE X1,Y1:PAPER 0:PRINT " "
750 PEN 1:LOCATE x1,y1:PRINT kod%:budka(nrb,h)=kodw
800 RUCH=RUCH+1
805 SUK=1:FOR I=1 TO 12
810 IF SZEREG(I)<>0 THEN SUK=0
815 NEXT I
820 FOR P=1 TO 8
830 suma(p)=0
840 FOR p1=1 TO 5
850 IF budka(p,p1)<>0 THEN suma(p)=suma(p)+1
860 NEXT p1:NEXT p
870 s1=suma(1)+suma(2)+suma(3):s2=suma(3)+suma(5)+suma(8):s3=suma(8)+suma(7)+suma(6):s4=suma(6)+suma(4)+suma(1)
890 IF etap<>1 GOTO 900
892 IF s1=s2 AND s2=s3 AND s3=s4 AND s4=4 AND SUK=1 THEN GOSUB 4000:etap=etap+1:GOTO 310
894 GOTO 400
900 IF etap<>2 GOTO 920
910 IF SUK=1 AND s1=s2 AND s2=s3 AND s3=s4 AND s4=5 THEN GOSUB 4000:etap=etap+1:GOTO 310
915 GOTO 400
920 IF SUK=1 AND s1=s2 AND s2=s3 AND s3=s4 AND s4=6 THEN GOSUB 4000:CLS:STOP
930 GOTO 400

```

Do programu dołączono kilka podprogramów, które spełniają następujące funkcje:

1. Rysowanie jednego posterunku w obszarze zadanym parametrami a, b (współrzędne znakowe szczytu daszku).

```

1000 PAPER 0:PEN 2:LOCATE a,b:PRINT CHR$(189);CHR$(143);CHR$(190)
1010 LOCATE a-1,b+1:PRINT CHR$(189);CHR$(193);CHR$(194);CHR$(195);CHR$(190)
1020 LOCATE a-2,b+2:PRINT CHR$(191);CHR$(143);CHR$(192);" ";CHR$(196);CHR$(143);CHR$(197)
1030 PAPER 3:LOCATE a-1,b+3:PRINT CHR$(189):LOCATE a+3,b+3:PRINT CHR$(190):LOCATE a-1,b+5:
PRINT CHR$(189):LOCATE a+3,b+5:PRINT CHR$(
190)
1040 PAPER 2:PEN 3:LOCATE a-1,b+2:PRINT CHR$(189):LOCATE a+3,b+2:PRINT CHR$(190):LOCATE a-
1,b+4:PRINT CHR$(189):LOCATE a+3,b+4:PRINT
CHR$(190)
1050 LOCATE a-1,b+6:PRINT CHR$(189):LOCATE a+3,b+6:PRINT CHR$(190)
1060 LOCATE a+1,b:PRINT USING "E";c:PAPER 0
1100 RETURN

```

2. Wyrysowanie na ekranie oddziału 12 żołnierzy z kodami literowymi oraz rozprowadzającego (ranga rozprowadzającego jest uzależniona od etapu gry).

```

2000 PAPER 0:PEN 1:n=65
2010 RESTORE 2000:FOR a=15 TO 30 STEP 3:FOR b=11 TO 15 STEP 4
2020 GOSUB 2100:LOCATE a+1,b+1:PAPER 1:PEN 2:PRINT CHR$(n):n=n+1:PAPER 0:PEN 1:NEXT b:NEXT
a
2030 a=10:b=13:GOSUB 2100:GOSUB 2200
2040 RETURN
2100 LOCATE a,b:PRINT CHR$(198);CHR$(199);CHR$(200)
2110 LOCATE a,b+1:PRINT CHR$(201);CHR$(143);CHR$(202)
2120 LOCATE a,b+2:PRINT CHR$(203);CHR$(143);CHR$(204)
2130 RETURN
2200 GRAPHICS PEN 2:ON etap GOTO 2210,2300,2400
2210 FOR wy=198 TO 170 STEP -14
2220 PLOT 166,wy:DRAW 2,0:DRAW 2,-2:DRAW -6,0:DRAW 0,-2:DRAW 6,0:DRAW -2,-2:DRAW -2
,0:NEXT wy
2230 GOTO 2500
2300 FOR wy=198 TO 168 STEP -10
2310 PLOT 166,wy:DRAW 2,0:DRAW 2,-2:DRAW -6,0:DRAW 0,-2:DRAW 6,0:DRAW -2,-2:DRAW -2
,0:NEXT wy
2320 GOTO 2500
2400 PLOT 166,188:DRAW 2,0:DRAW 2,-2:DRAW -6,0:DRAW 0,-2:DRAW 6,0:DRAW -2,-2:DRAW -
,0
2410 PLOT 154,174:DRAW 26,0:DRAW 0,-2:DRAW -26,0:PLOT 152,166:DRAW 30,0:DRAW 0,-2:ORA
R -30,0
2500 RETURN

```



3. Ustalenie współrzędnych znakowych pierwszego wolnego miejsca w budce wartowniczej (x1, y1).

```

3000 X1=4:IF NRB=2 OR NRB=7 THEN X1=21
3010 IF NRB=3 OR NRB=5 OR NRB=8 THEN X1=37
3020 FOR H=1 TO 5
3030 IF BUDKA(NRB,H)=0 GOTO 3050
3040 NEXT H
3050 Y1=4+H:IF NRB=4 OR NRB=5 THEN Y1=Y1+8
3060 IF NRB=6 OR NRB=7 OR NRB=8 THEN Y1=Y1+16
3070 RETURN

```

4. Zakończenie sukcesem dowolnego etapu gry.

```

4000 PAPER 0:PEN 1:LOCATE 15,12:PRINT"OSIAGNALES SUKCES":LOCATE 18,14:PRINT" * "; USING "E"
E";RUCH-1:PRINT" RUCHACH
4010 PAPER 1:PEN 2:LOCATE 1,1:PRINT"
B R A W O
4030 FOR I=0 TO 26:ORDER I
4040 FOR T=1 TO 100:NEXT T:NEXT I
4050 RETURN

```

5. Definiowanie własnych symboli graficznych wykorzystanych do rysowania budki i oddziału żołnierzy.

```

9000 SYMBOL AFTER 199
9400 SYMBOL 193,255,255,255,255,255,254,252,248
9410 SYMBOL 194,255,255,255,255
9420 SYMBOL 195,255,255,255,255,255,127,63,31
9430 SYMBOL 196,15,7,3,1
9440 SYMBOL 197,0,128,192,224,240
9450 SYMBOL 189,0,1,3,7,15,31,63,127
9460 SYMBOL 190,0,128,192,224,240,248,252,254
9470 SYMBOL 191,0,1,3,7,15
9480 SYMBOL 192,240,224,192,128
9490 SYMBOL 198,0,0,0,0,1,1,1,1
9500 SYMBOL 199,24,60,126,255,255,255,255
9510 SYMBOL 200,0,0,0,0,128,128,128,128
9520 SYMBOL 201,1,3,3,3,3,3,7
9530 SYMBOL 202,128,192,192,192,192,192,224,224
9540 SYMBOL 203,7,7,7,15,15,15,15
9550 SYMBOL 204,224,224,224,240,240,240,240,240
9600 RETURN

```

Życzę Czytelnikom sukcesów w kolejnych etapach gry.

Bardziej dociekliwym proponuję rozwinięcie gry o etap 4. Łatwo bowiem zauważyć, że istnieje jeszcze jedno możliwe rozstawienie wartowników, w którym każdej ścianie chroni 3 żołnierzy. Jakże? Poszukajcie sami. Rozprowadzającym w tym etapie mógłby być porucznik.

Jeżeli modyfikacja programu zakończy się powodzeniem, będzie to dodatkowa satysfakcja.

Mieczysław SKONIECZNY

BASIC

Powróćmy jeszcze do „pocziwego” Basica.

W wykładzie tym omówimy operacje na tekstach oraz związane z tym odpowiednie instrukcje i funkcje. Zajmować będziemy się także tablicami alfanumerycznymi.

Wykład ósmy:

stałe, i zmienne proste alfanumeryczne, nadawanie „wartości” zmiennej alfanumerycznej, łączenie tekstów (konkatenacja), wybieranie tekstów (za pomocą TO), funkcje standardowe:

LEN, STR\$, VAL, VAL\$, CHR\$, CODE, porównywanie znaków (operatory relacji, instr. IF) tablice alfanumeryczne.

Istnieje grupa problemów, których rozwiązanie opiera się na przetwarzaniu tekstów. Dla tych problemów tworzone są specjalne języki programowania zorientowane tylko na przekształcenie tekstów, natomiast pozostałe języki są wzbogacane o różne operacje tekstowe. I tak jak w innych reprezentacjach języka BASIC, w BASIC-u na ZX-Spectrum wprowadzono zmienne alfanumeryczne proste i indeksowane, rozszerzono działania funkcjonalne instrukcji podstawienia i warunkowej, wprowadzono funkcje standardowe wykonujące operacje na tekstach.

Tekstem w języku BASIC jest ciąg znaków. W wykładzie 2 i 3 omawialiśmy stałe i zmienne proste alfanumeryczne, powtórzmy wybrane definicje.

Stała alfanumeryczna (tekstowa) ma postać:

„ciąg znaków”

- ciąg znaków: dowolny ciąg znaków.
- znaki cudzysłowu ograniczają stałą tekstową, nie należą do stałej, lecz jedynie wyznaczają jej początek i koniec.
- tekst " " bez żadnego znaku zwany jest tekstem pustym.
- tekst " " nie jest tekstem pustym, oznacza jedną spację; spacje wewnątrz cudzysłowu są znakami tekstu.

Długością tekstu jest liczba znaków (ze znakami spacji włącznie) znajdujących się w ciągu. Maksymalna długość tekstu wynosi 255 znaków.

Poniższe teksty mają następujące liczby znaków:

„ZX SPECTRUM” — 11 znaków
„WYNIK = a+b*c-d” — 15 znaków

Analogicznie do pojęcia zmiennej prostej numerycznej istnieje pojęcie zmiennej prostej alfanumerycznej, której „wartością” jest tekst. Nazwa takiej zmiennej składa się z litery (tylko jednej), po której musi nastąpić znak dolara (\$), np. a\$, p\$, r\$ są nazwami zmiennych prostych alfanumerycznych.

Z dotychczasowych wykładów wynika, że:

— tekst może być zapamiętany pod nazwą zmiennej alfanumerycznej.

Nazwom zmiennych numerycznych można nadawać „wartości”, używając instrukcji LET lub INPUT lub ze zbioru DATA instrukcją READ np.

20 LET m \$ — „mały”
lub
10 INPUT m \$
(z klawiatury wprowadzamy: „mały”)
lub
30 READ m\$

100 DATA „mały”
— tekst może być wydrukowany instrukcją PRINT: „bezpośrednio” np.:

40 PRINT „mały”
można też wydrukować tekst zapamiętany pod zmienną alfanumeryczną np.:

20 LET m \$ = „mały”
40 PRINT m \$

Do tekstów można stosować wszystkie operatory relacji, a tym samym instrukcję IF.

Kolejne operacje na tekstach wraz z dotychczas poznanymi zostaną omówione na przykładach. Zakładamy, że instrukcje

wprowadzania (INPUT, READ, DATA) i wyprowadzania (PRINT) tekstów są już opanowane przez czytelnika. A więc włączamy komputer i zaczynamy.

Przykład 8.1.

a) Dany jest program:

```
5 REM przykład 8.1
10 INPUT "a$ = ";a$: PRINT "a$
=";a$
20 INPUT "b$ = ";b$: PRINT "b$
=";b$
30 INPUT "c$ = ";c$: PRINT "c$
=";c$
40 INPUT "d$ = ";d$: PRINT "d$
=";d$
50 PRINT "a$; b$; c$; d$
60 LET e$=a$+" "+b$+" "+c$+" "
+d$
70 PRINT "e$ = ";e$
80 PRINT "Wybieranie fragment
ow tekstu przy uzyciu TO"
90 PRINT "e$(1 TO 5), ":";e$(T
O 12)
100 PRINT "e$(3 TO 10), ":";e$(3
TO )
110 PRINT "e$( TO ), ":";e$(1 TO
15)
120 PRINT "e$(), ":";e$(9)
130 STOP
```

b) wyniki dla przykładowych danych

```
a$ = is
b$ = to
c$ = jest
d$ = pies
```

```
Asto jest pies
```

```
e$ = As to jest pies
```

```
Wybieranie fragmentow tekstu
przy uzyciu TO
```

```
As to :As to jest p
```

```
to jest : to jest pies
```

```
As to jest pies :As to jest pies
```

```
As to jest pies :s
```

Do przenoszenia tekstów między zmiennymi alfanumerycznymi oraz do przypisywania zmiennym „wartości” stałych alfanumerycznych służy instrukcja podstawienia postaci:

n LET V = e

n — nr wiersza,

V — nazwa zmiennej alfanumerycznej,

e — wyrażenie tekstowe

Wyrażenie tekstowe może być stałą lub zmienną alfanumeryczną — prostą lub indeksowaną, standardową lub definiowaną funkcją alfanumeryczną. Elementy wyrażenia tekstowego można łączyć operatorem konkatenacji, oznaczanym symbolicznie znakiem plus (+) tworząc jeden tekst. Postać i wykonanie tej instrukcji jest analogiczne do arytmetycznej instrukcji podstawienia (por. wykład 2)

I tak wykonanie instrukcji:

```
60 LET e$ = a$ + b$ + c$ + d$
```

spowoduje przypisanie (nadanie) zmiennej alfanumerycznej e\$ wartości będącej tekstem utworzonym z 4 tekstów (przez sumowanie).

Mając dany tekst można wyróżnić jako jego fragment dowolny tekst składający się z części jego kolejnych znaków ustawionych w tym samym porządku.

Instrukcja zwana wybieraniem służąca do określenia fragmentu dowolnego tekstu (w BASIC-u na ZX Spectrum) ma postać: wyrażenie tekstowe (p TO k)

— wyrażenie tekstowe: jak wyżej.

— p, k — stałe lub zmienne numeryczne całkowite, określające odpowiednio: początek (p), koniec (k), wybranego tekstu i muszą odnosić się do istniejących części tekstu.

np.:

"abcdef" (2 TO 5) = "bcde"

Obowiązuje tu kilka reguł:

— jeżeli pominiemy początek (TO k), to zakłada się, że jest on równy 1 i wybrane zostaną znaki od 1 do k — tego — danego wyrażenia tekstowego,

— jeżeli pominiemy koniec (p TO), to zakłada się, że jest on równy ostatniemu znakowi i wybrane zostaną znaki od p — tego do ostatniego danego wyrażenia tekstowego,

— napisanie (TO) lub () powoduje wybranie wszystkich znaków tekstu,

— napisanie (m) (m — stała lub zmienna określająca liczbę całkowitą), powoduje wybranie m — tego znaku tekstu ($m \leq$ od liczby znaków tekstu),

— jeżeli $p > k$ (początek $>$ koniec), to efektem jest tekst pusty,

— jeżeli $k > 1$ (koniec $>$ liczby znaków wyrażenia tekstowego) to będzie sygnalizowany błąd:

3 subscript wrong
(zły indeks)

— jeżeli początek lub koniec są liczbami ujemnymi ($p < 0$ lub $k < 0$) to będzie sygnalizowany błąd:

B integer out of range
(liczba całkowita poza zakresem)

Prostą ilustracją obowiązujących reguł są instrukcje o nr. 80 ÷ 120 w programie z przykładu 8.1.

Analizę wykonania tych instrukcji pozostawiamy czytelnikowi, zalecając wykonanie tego programu dla różnych tekstów (zwracając uwagę na różną długość tych tekstów).

W celu ugruntowania nowych wiadomości wprowadźmy i wykonajmy następujące programy:

Przykład 8.2.

a) Dany jest program:

```
10 REM Przykład 8.2
20 LET W$="Katarynka"
30 FOR n=1 TO 9
40 PRINT W$(n TO 9)
50 NEXT n
60 STOP
```

b) wyniki:

```
Katarynka
atarynka
tarynka
arynka
rynka
ynka
nka
ka
a
```

Przykład 8.3.

a) dany jest program:

```
10 REM Przykład 8.3
20 LET P$="warszawianka"
30 FOR n=1 TO 12
40 PRINT P$(n TO 12),P$((12-n+
TO 12)
50 NEXT n
60 STOP
```

b) wyniki:

```
warszawianka a
arszawianka ka
rszawianka nka
szawianka anka
zawianka ianka
awianka wianka
wianka awianka
ianka zawianka
anka szawianka
nka rszawianka
ka arszawianka
a warszawianka
```

Programy z przykładów 8.2 i 8.3 radzimy wykonać kilka razy, zmieniając stałą tekstową w instrukcji o nr. 20.

Przykład 8.4.

a) Dany jest program:

```
10 REM Przykład 8.4
20 INPUT "Twoje imie? ";i$
30 INPUT "Twoje nazwisko? ";n$
40 LET X$=i$+n$
50 LET Y$=i$+" "+n$
60 PRINT "Nazywam sie: "
70 PRINT 'X$
80 PRINT 'Y$
90 LET K=LEN (i$)
100 LET M=LEN (n$)
110 LET P=LEN (i$+n$)
120 LET T=LEN (i$+" "+n$)
125 PRINT "Twoje imie: ";i$;"
zawiera ";k;" znaki(ow)"
130 PRINT "Twoje nazwisko: ";n$;"
zawiera ";m;" znaki(ow)"
140 PRINT "tekst w zmiennej x$
";X$;" zawiera ";P;" znaki(ow)
150 PRINT "tekst w zmiennej y$
";Y$;" zawiera ";t;" znaki(ow)
160 STOP
```

b) wyniki dla przykładowych danych:

Nazywam sie:

KazimierzAbacki

Kazimierz Abacki

Twoje imie: Kazimierz zawiera 9 znaki(ow)

Twoje nazwisko: Abacki zawiera 6 znaki(ow)

tekst w zmiennej x\$: KazimierzAbacki zawiera 15 znaki(ow)

tekst w zmiennej y\$: Kazimierz Abacki zawiera 17 znaki(ow)

W wykładzie 3 omawialiśmy funkcje standardowe, służące do obliczania wartości podstawowych funkcji matematycznych. Teraz omówimy funkcje standardowe służące do „wykonywania obliczeń” na tekstach.

W programie z przykładu 8.4 w instrukcjach o nr. 90, 100, 110, 120 występuje funkcja o nazwie LEN.

Funkcja LEN oblicza długość tekstu, jej argumentem jest tekst, którego długość chcemy obliczyć, a wartością (wynikiem) — jego długość.

Weźmy pod uwagę instrukcję:

```
110 LET p = LEN (X$)
```

```
115 LET t = LEN (Y$)
```

i zapiszemy je w następujący sposób

```
110 LET p = LEN (I$) + LEN (N$)
```

```
(i) 115 LET t = LEN (I$) + LEN (" ") + LEN (N$)
```


prawej. Jeżeli porównane znaki są różne, to jeden z nich ma kod mniejszy od drugiego i tekst rozpoczynający się tym znakiem jest wcześniejszy, (mówimy, że jest mniejszy). Jeżeli są takie same, to porównujemy następne znaki. Tak więc określenia relacji między tekstami następuje po znalezieniu pierwszej pary nieidentycznych znaków. Jeżeli w porównywanych znakach nie pojawi się różnica w kodach kolejnych znaków, ale jeden z tekstów jest krótszy, to jest on wcześniejszy. Przy takiej samej długości są one równe.

Operatory relacji =, <, >, <=, >= oraz <> mogą być użyte do tekstów tak samo jak do liczb (np. < oznacza „jest przed”, a > — „jest po”). Wynikiem obliczenia jest jedna z dwóch wartości — 1 oznaczającej wartość prawda lub 0 oznaczającą fałsz.

Dla porównania testów rozszerzoną warunkową instrukcją sterującą o testowanie wyniku relacji. Postać i wykonanie tej instrukcji jest analogiczne do instrukcji IF... THEN... Tak więc powyższa instrukcja umożliwia wyszukiwanie i analizę tekstów.

Poniższe programy obrazują wykonanie funkcji CODE oraz instrukcji IF... THEN...

Przykład 8.7.

a) Dany jest program:

```

10 REM Przykład 8.7
15 PRINT "tekst"; " " "ko
d pierwszego znaku": PRINT
20 FOR i=1 TO 12
30 READ a$
40 PRINT a$,CODE (a$)
50 NEXT i
55 STOP
70 DATA "ALFA","BETA","C","D",
"EWA"," "," ","alf","beta","c",
"d","ewa"

```

b) wyniki:

tekst	kod pierwszego znaku
ALFA	65
BETA	66
C	67
D	68
EWA	69
	70
	71
	72
	73
	74
	75
	76
	77
	78
	79
	80
	81
	82
	83
	84
	85
	86
	87
	88
	89
	90
	91
	92
	93
	94
	95
	96
	97
	98
	99
	100
	101
	102
	103
	104
	105
	106
	107
	108
	109
	110
	111
	112
	113
	114
	115
	116
	117
	118
	119
	120
	121
	122
	123
	124
	125
	126
	127
	128
	129
	130
	131
	132
	133
	134
	135
	136
	137
	138
	139
	140
	141
	142
	143
	144
	145
	146
	147
	148
	149
	150
	151
	152
	153
	154
	155
	156
	157
	158
	159
	160
	161
	162
	163
	164
	165
	166
	167
	168
	169
	170
	171
	172
	173
	174
	175
	176
	177
	178
	179
	180
	181
	182
	183
	184
	185
	186
	187
	188
	189
	190
	191
	192
	193
	194
	195
	196
	197
	198
	199
	200

Przykład 8.8.

a) Dany jest przykład:

```

10 REM Przykład 8.8
20 PRINT "Ustawianie dwóch tek
stów w kolejności alfab
etycznej"
30 PRINT "teksty dane : "
40 INPUT "wprowadz dwa teksty
: ",x$,y$
50 PRINT "x$;" "y$
60 IF x$>y$ THEN LET z$=x$:
LET x$=y$: LET y$=z$
70 PRINT "teksty po porównani
u : "
80 PRINT "x$;" "y$
90 IF x$<y$ THEN PRINT "<";: G
O TO 110
100 PRINT "=";
110 PRINT " ";y$
120 STOP

```

b) Wyniki dla przykładowych danych:

ustawianie dwóch tekstów
w kolejności alfabetycznej

teksty dane :

katarunka katar

teksty po porównaniu :

katar < katarunka

Przykład 8.9.

Napisać program znajdowania w zestawie DATA nazwisk o za-danym z klawiatury imieniu.

a) program:

```

10 REM Przykład 8.9
20 PRINT " *** wyszukiwanie
nazwisk ***"
30 INPUT "podaj imie : ";a$
40 PRINT "podane imie : ";a$
50 READ n
60 LET k=0
70 FOR i=1 TO n
80 READ b$,c$
90 IF b$<>a$ THEN GO TO 120
100 PRINT "znalezione nazwisko
: ";c$
110 LET k=k+1
120 NEXT i
130 IF k=0 THEN PRINT "w zestaw
ie danych nie bylo nazwisk
o podanym imieniu": STOP
140 PRINT "w zestawie danych b
ylo ";k"nazwisk(a)";
150 PRINT " o imieniu ";a$
160 STOP
170 REM n-liczba par (imie,naz-
wisko)
180 DATA 5
190 DATA "Agata","Kowalska","Ew
a","Wisniewska","Ewa","Klobukows
ka","Irena","Polkowska","Anna","
Xksinska"

```

b) Wyniki dla przykładowych danych:

```

*** wyszukiwanie nazwisk ***
podane imie : Ewa
znalezione nazwisko :
Wisniewska
znalezione nazwisko :
Klobukowska
w zestawie danych bylo 2
nazwisk(a) o imieniu Ewa

```

Zalecamy czytelnikowi poćwiczenie z programami z przykła-dów 8.7, 8.8, 8.9 dla różnych danych wejściowych.

W podanych przykładach używaliśmy zmiennych prostych alfa-numerycznych. Takiej zmiennej mogliśmy nadać „wartość” będą-cą tekstem o długości od 0 do 255 znaków. Analogicznie do zmiennych indeksowanych i tablic numerycznych wprowadzamy zmiennie indeksowane i tablice alfanumeryczne.

Teksty w tablicy różnią się od zmiennych prostych alfanumery-cznych (tekstowych) tym, że mają określoną długość 1 i są dopa-sowane zgodnie z zasadą Prokrutesa — obcinane z prawej strony, albo dopełniane spacjami do wymaganej długości. Nazwa tablicy tekstowej jest pojedynczą literą ze znakiem dolara (\$) i przeciwie do tablic numerycznych zmienna prosta i tablica tekstowa nie mogą mieć tej samej nazwy.

Deklaracja tablicy postaci np.:

```
DIM X$ (3,15)
```

rezerwuje w pamięci komputera tablicę 3 X 10 znaków, może za-wierać ona np. trzy wiersze tekstu, a każdy ma 15 znaków:

X\$ (1) = X\$ (1, 1) X\$ (1, 2) X\$(1, 15)
 X\$ (2) = X\$ (2, 1) X\$ (2, 2) X\$(2, 15)
 X\$ (3) = X\$ (3, 1) X\$ (3, 2) X\$(3, 15)

Jeżeli napiszemy zmienną indeksowaną postaci:
 X\$ (1) lub X\$ (2) lub X\$(3), to rozumiemy, że odnosi się ona do całego wiersza tablicy X\$ (3, 15), (otrzymamy tekst o określonej długości (15 znaków), jeżeli natomiast napiszemy X\$ (2, 8), to otrzymamy jeden znak, tzn. ósmy znak wiersza drugiego (X\$ (2)).
 Tak więc w tablicy alfanumerycznej (tekstowej) wszystkie wiersze mają stałą określoną długość, a deklaracja tablicy DIM zawiera dodatkową liczbę (ostatnią) określającą tę długość.
 Zamiast ostatniego indeksu (który z zasady opuszczamy) możemy użyć instrukcji do wybierania fragmentów tekstu np.:

```
10 DIM X$(3,15)
20 LET X$(2)="abcdefghijklmnop"
30 LET Y=X$(2,9 TO 14)
40 LET Z=X$(2) (9 TO 14)
50 PRINT Y$,Z$
60 REM Y$=Z$="ijklmn"
```

Deklaracja tablicy postaci np.:

DIM a\$ (6)

deklaruje nam tablicę tekstową bezwymiarową, tzn. a\$ zachowuje się jak zmienna prosta alfanumeryczna, przy czym zawsze ma długość równą 6 znaków i znaki dopasowane są wg zasady Prokrutesa.

Uwaga: w innych reprezentacjach BASICU tablice alfanumeryczne mogą mieć inną organizację, np. taką samą, jak tablice numeryczne, tzn. że w każdy element tablicy mogą wpisać tekst o dowolnej długości (długość — określona max. liczbą znaków tekstu dla danej reprezentacji).

Omówione własności zmiennych numerycznych oraz ich funkcje są często stosowane w programach graficznych, np. w rysowaniu wykresu funkcji. Wykorzystanie ich przy przetwarzaniu tekstów obrazują poniższe dwa przykłady.

Przykład 8.10.

Zadanie do wykonania:

1. Założyć książkę telefoniczną (przykładową)
2. Na żądanie:
 - wyświetlić nr telefonu dla nazwiska abonenta podanego z klawiatury,
 - wyświetlić nazwisko abonenta dla numeru telefonu podanego z klawiatury

a) program:

```
10 REM Przykład 8.10
20 REM n$ - nazwiska abonentow
   o dlugosci (maksymalnej)
   nazwiska 20 znakow
25 REM t - numery telefonow
30 INPUT "podaj liczbe wprowad
zanych nazwisk : ";n
35 PRINT "Książka telefoniczn
a zawiera : ";n;" na
zwisk"
40 DIM n$(n,20): DIM t(n): DIM
k$(20)
45 FOR i=1 TO n
50 INPUT "nazwisko : ";n$(i)
55 INPUT "nr. telefonu : ";t(i)
60 NEXT i
65 PRINT "   wybieramy wedlug
:
:   - nazwiska      n
:   - nr. telefonu  n
:
:   nacisnij - 1
:   nacisnij - 2"
70 INPUT z
75 IF z<>1 AND z<>2 THEN GO TO
65
80 IF z=2 THEN GO TO 135
85 INPUT "podaj nazwisko : ";k
$
90 FOR i=1 TO n
95 IF k$=n$(i) THEN GO TO 110
100 NEXT i
105 PRINT "brak nazwiska w ksi
azce": GO TO 115
```

```
110 PRINT k$ "nr.tel. : ";t(i)
115 INPUT "szukamy dalej ?(t/n)
: ";x$
120 IF x$="T" OR x$="t" THEN GO
TO 65
125 PRINT "koniec pracy"
130 STOP
135 INPUT "nr. telefonu = ";y
140 FOR i=1 TO n
145 IF y=t(i) THEN GO TO 160
150 NEXT i
155 PRINT "brak nr. telefonu ";
y;" w ksiazce": GO TO 115
160 PRINT n$(i),"nr.tel. : ";t
(i)
165 GO TO 115
```

b) wyniki dla przykładowych danych

Książka telefoniczna
 zawiera : 5 nazwisk

wyberamy wedlug :
 - nazwiska nacisnij - 1
 - nr. telefonu nacisnij - 2

XYZ
 nr.tel. : 393939

wyberamy wedlug :
 - nazwiska nacisnij - 1
 - nr. telefonu nacisnij - 2

Abacki
 nr.tel. : 102200

koniec pracy

Przykład 8.11.

Napisać program, który wprowadza do tablicy n — tekstów o max. długości 20 znaków każdy, następnie w każdym z wprowadzonych tekstów zlicza częstotliwość występowania podanych liter (umieszczonych w DATA).

a) program

```
10 REM Przykład 8.11
15 REM a$ tablica tekstow o
   dlugosci 20 znakow
20 INPUT "podaj liczbe wprowad
zanych tekstow : ";n
25 PRINT "wyszukiwanie i zlicz
anie odpo-
wiednich liter w dan
ych tekstach"
30 DIM a$(n,20)
35 REM litery ktorych szukamy
umieszczone sa w zestawie DATA
40 FOR i=1 TO n
45 INPUT (i);" -tekst : ";a$(i)
)
50 NEXT i
60 FOR i=1 TO n
65 READ m
70 PRINT "tekst : ";i;" : "
75 PRINT a$(i)
80 FOR k=1 TO m
85 READ z$
90 LET p=0
95 FOR j=1 TO 20
100 IF a$(i,j)=z$ THEN LET p=p+
1
105 NEXT j
110 PRINT "w tekście : ";i;"
litera : ";z$" wystąpiła : ";p;"
razy"
115 NEXT k
120 RESTORE : COPY : CLS
125 PRINT
130 NEXT i
135 STOP
140 DATA 6
145 DATA "a","c","o","p","x","z"
```

Test czytania

Po uruchomieniu programu na ekranie pojawia się na krótko wyraz, który należy następnie napisać. Jeżeli wyrazy pisane są poprawnie, to oczywiście przybywa nam punktów w tej rozgrywce z komputerem. Czas trwania wyrazu na ekranie można zmniejszyć lub zwiększyć, zmieniając długość pętli w linii 170. Może być to zatem pożyteczna zabawa dla przedszkolaków i dla bardziej zaawansowanych. Oczywiście, w programie mogą być użyte inne wyrazy.

```

10 REM TEST CZYTANIA
27 POKE 53281,13:POKE 53280,12
30 AT$=CHR$(17):FOR T=1TO6:AT$=AT$+AT$:N
EXT T:AT$=CHR$(19)+AT$
35 GOSUB 10000
40 SC=0
60 HS=0
70 DIMW$(50)
80 FOR J=1TO50
90 READW$(J)
100 NEXT J
110 FOR N=1TO20
120 R=INT(RND(1)*50)+1
130 PRINT CHR$(147)
140 PRINTLEFT$(AT$,2)SPC(4)'WYNIK: ' ;SC
150 PRINTLEFT$(AT$,2)SPC(18)'NAJWYŻSZY W
YNIK: ' ;HS
160 PRINTLEFT$(AT$,8)SPC(18-(LEN(W$(R))/
2))W$(R):WW$=W$(R)
165 IF LEN(W$(R))<10 THEN W$(R)=W$(R)+'
':GOTO 165
170 FOR DL=1TO500 :NEXT DL
180 PRINTLEFT$(AT$,8)SPC(9)'
190 PRINTLEFT$(AT$,16)SPC(9)'JAKIE TO BY
LO SLOWO?'
195 PRINTLEFT$(AT$,18)SPC(9);
200 INPUT A$
202 IF LEN(A$)<10 THEN A$=A$+' ':GOTO 20
2
210 IF A$=W$(R) THEN 270
220 POKE 50+1,20:POKE 50+4,17:FOR QQ=1TO
30:NEXT QQ:POKE 50+4,16
230 PRINTLEFT$(AT$,8)SPC(11)'TYM SLOWEM
BYLO: '
235 PRINTLEFT$(AT$,10)SPC(17-(LEN(WW$)/2
))WW$
244 PRINTLEFT$(AT$,16)SPC(3)'
245 PRINTLEFT$(AT$,18)SPC(3)'
250 FOR DL=1TO 400:NEXT DL
260 GOTO 300
270 POKE 50+1,25:POKE 50+4,17:FOR QQ=1TO
30:NEXT QQ:POKE 50+4,16
280 SC=SC+10
300 NEXTN
310 IF SC>HS THEN HS=SC
312 PRINTCHR$(147)
314 PRINTLEFT$(AT$,8)SPC(3)'TWOJ KONCOWY
WYNIK.....'SC
316 IF SC >HS THEN PRINTLEFT$(AT$,10)SPC
(3)'TO JEST NAJLEPSZY WYNIK DZISIAJ.'
320 INPUT 'JESZCZ RAZ (T/N)';Q$
330 IF LEFT$(Q$,1)='N' THEN 380
340 SC=0
360 GOTO110
380 END
400 DATA 'OGROD','CHLOPIEC','DZIEWCZYNA'
410 DATA 'STRAWA','OGAREK','KOBIETA'
420 DATA 'LAMPA','PIES','KOT'
430 DATA 'OKNO','DRZWI','SCIANA'
440 DATA 'DZIS','JUTRO','WCZORAJ'
450 DATA 'KWIAT','ROLA','WAZA'
460 DATA 'LAS','TRAWA','SZOSA'
470 DATA 'KASIA','OBRAZ','SZARY'
480 DATA 'SZYMPANS','KROWA','ZEBRA'
490 DATA 'KOZA','KOGUT','KACZKA'
500 DATA 'MONIKA','MARCIN','DANUTA'
510 DATA 'ADAM','TOMEK','ZOSIA'
520 DATA 'PIERWSZY','DRUGI','TRZECI'
530 DATA 'OSTATNI','NAJLEPSZY','DOBRY'
540 DATA 'PAPIER','ALA','AS'
550 DATA 'WINDO','SZKLANKA','WODA'
560 DATA 'LITERA','SYLABA'
10000 REM DZWIEK
10010 SO=54272
10020 FORT=0TO28:POKESO+T,0:NEXT
10030 POKESO+1,20
10040 POKE SO+5,9
10050 POKE SO+6,240
10060 POKE SO+24,15
10070 POKE SO+4,16
10080 RETURN

```

T. CISEK

PROGRAM

40

41

Histogram

Program służy do otrzymywania histogramów. Po jego uruchomieniu otrzymamy pytanie o liczbę słupków (maksymalnie 9). Gdy komputer uzyska odpowiedź, zostanie uruchomione przyjmowanie danych o poszczególnych słupkach. Kolejnym etapem jest kreślenie histogramu. Pod każdym słupkiem pojawia się kolejny numer; wartość danej w zaokrągleniu (liczby większe od stu są zredukowane do dwóch pierwszych cyfr); oraz procentową

zawartość danej w stosunku do sumy wszystkich składników histogramu.

Ostatni słupek odwzorowuje sumę składników.

Największa podana wartość słupka uzyskuje pełną wysokość ekranu, pozostałe są jej procentowym odwzorowaniem.

Naciśnięcie klawisza START powoduje ponowne uruchomienie programu.

Artur NOWAKOWSKI

```

AP 10 REM ***** HISTOGRAM *****
MR 20 REM *** wersja 1.0 POPR.***
ZM 30 REM *** (c) 1987-08-11***
KC 40 REM ** Artur Nowakowski **
AO 50 REM ul. W. Wasilewskiej 5/8
GF 60 REM 08-110 Siedlce
LF 70 REM POLSKA
VG 80 REM /tel. 209-02/
AK 90 DIM A$(1),B$(9),ILOSC(9)
GN 100 REM STRONA TYTULOWA
SW 110 GRAPHICS 1
BX 120 B$="histogram"
VK 130 FOR X=1 TO 9
VJ 140 A$=B$(X,X):WYS=INT(RND(0)*13
+1)
AA 150 FOR Y=1 TO WYS
HL 160 POSITION 2*X,14-Y: ? #6;A$
MJ 170 NEXT Y
MB 180 NEXT X
CK 190 POSITION 0,14: ? #6;" WER
SJA 1.0"
SW 200 ? #6;" (C) 1987-08-11"
TQ 210 ? #6;" artur nowakowski"
CJ 220 ? #6;" W.WASILEWSKIEJ 5/8"
UD 230 ? #6;" 08-110 SIEDLCE"
JA 240 ? #6;" POLSKA"
RT 250 IF SPR=1 THEN GOTO 300
MV 260 ? " ";CHR$(145);CHR
$(146);CHR$(146);CHR$(146);CHR$(
146);CHR$(146);CHR$(133)
IB 270 ? "Nacisnij ";CHR$(252);CHR
$(211);CHR$(212);CHR$(193);CHR$(
210);CHR$(212);CHR$(252)
GG 280 ? " ";CHR$(154);CHR
$(146);CHR$(146);CHR$(146);CHR$(
146);CHR$(146);CHR$(131)
PW 290 IF PEEK(53279)<>6 THEN POKE
755,0:GOTO 290
OR 300 REM PODAWANIE DANYCH
WQ 310 POKE 755,0: ? CHR$(125);"Ile
słupkow (1-9) ";
KQ 320 TRAP 300
HL 330 INPUT SKLAD
PO 340 IF SKLAD>9 OR SKLAD<1 THEN G
OTO 300
AC 350 SUM=0:REM *****
PA 360 ? CHR$(125);"Podawaj kolejno
dane." ? : ?
PO 370 FOR X=1 TO SKLAD:TRAP 870
IU 380 ? CHR$(28);"
"
XH 390 ? CHR$(28);" ";X;" - ";
INPUT ILOSC:IF ILOSC<0 THEN GOT
O 880
TF 400 ILOSC(X)=ILOSC:SUM=SUM+ILOSC
(X)
LQ 410 NEXT X
UK 420 REM SZUKANIE NAJWIEKSZEJ DAN
EJ
EZ 430 FOR X=1 TO SKLAD
FK 440 FOR Y=1 TO SKLAD
MV 450 IF ILOSC(X)>=ILOSC(Y) THEN N
EXT Y:MAX=ILOSC(X):GOTO 470
MA 460 NEXT X
RG 470 REM RYSOWANIE SLUPKOW

```

```

TL 480 IF MAX<=0 THEN SPR=1:GOTO 10
0
BV 490 IF MAX>=100 THEN SUM=0:FOR X
=1 TO SKLAD:ILOSC(X)=ILOSC(X)/10
:SUM=SUM+ILOSC(X):NEXT X:GOTO 42
0
UA 500 GRAPHICS 3
KH 510 POKE 755,0
CI 520 RESTORE
FA 530 FOR X=1 TO SKLAD
SD 540 ? X;" ";:NEXT X
OC 550 ? " SUMA"
FG 560 FOR X=1 TO SKLAD
YF 570 A$=""
IN 580 IF ILOSC(X)<10 THEN A$=""
ND 590 ? INT(ILOSC(X));" ";A$;:NEXT
X
CU 600 ? "-ILOSC"
EX 610 FOR X=1 TO SKLAD
UD 620 WYS=INT(ILOSC(X)*20/MAX)
PT 630 A$="" :IF ILOSC(X)*100/SUM<10
THEN A$=""
ZI 640 ? INT(ILOSC(X)*100/SUM);" ";
A$;
KR 650 IF ILOSC(X)*20/MAX<0.5 AND X
<SKLAD THEN NEXT X
HI 660 IF ILOSC(X)*20/MAX<0.5 THEN
GOTO 720
IF 670 READ A$
AL 680 FOR Y=1 TO WYS
RR 690 POSITION 3*X-1,20-Y: ? #6;A$
MB 700 NEXT Y
LT 710 NEXT X
KB 720 REM OSTATNIE SLUPKI
OC 730 KOL=INT(SUM/MAX):WYS=INT(SUM
*20/MAX)-20*KOL
UL 740 READ A$: ? " 100 x"
RA 750 IF KOL<1 THEN GOTO 800
BD 760 FOR X=4 TO KOL+3
XJ 770 FOR Y=1 TO 20
JJ 780 POSITION 3*SKLAD+X-1,20-Y: ?
#6;A$
UW 790 NEXT Y:NEXT X
OJ 800 FOR X=0 TO WYS:POSITION 3*SK
LAD+KOL+3,20-X
SP 810 ? #6;A$
LW 820 NEXT X
TV 830 ? " Nacisnij ";CHR$(
160);CHR$(211);CHR$(212);CHR$(19
3);CHR$(210);CHR$(212);CHR$(160)
;
DG 840 IF PEEK(53279)<>6 THEN 840
FN 850 SPR=1:GOTO 100
CM 860 REM KONIEC PROGRAMU
JD 870 REM OBSLUGA BLEDU
HZ 880 ? CHR$(253);CHR$(29);" Zla
dana! - popraw"
OF 886 FOR Y=1 TO 1000:NEXT Y: ? CHR
$(28);"
DP 888 ? CHR$(28);CHR$(28);CHR$(28)
;" " :X=X-1
MK 890 NEXT X
BV 900 REM DANE KOLOROW
LH 910 DATA A,B,C,A,B,C,A,B,C,A

```


INFORMATYCZNY SŁOWNIK ANGIELSKO-POLSKI

COMPUTER MEMORY - pamięć komputera,
COMPUTER MICROGRAPHICS - mikrografika komputerowa,
COMPUTER NETWORK - sieć komputerowa, zespół (połączonych ze sobą) komputerów,
COMPUTER NUMERICAL CONTROL - sterowanie numeryczne za pomocą komputera,
COMPUTER ORIENTED LANGUAGE - język (programowania) zorientowany maszynowo,
COMPUTER OUTPUT MICROFILMER - rejestrator mikrofilmowy (rejestruje obraz z komputera na mikrofilmie lub mikrofiszach),
COMPUTER OUTPUT MICROFILM - dane wyjściowe z komputera w postaci mikrofilmu,
COMPUTER OUTPUT MICROFILM - dane wyjściowe na mikrofilmie,
COMPUTER PROGRAM - program wewnętrzny,
COMPUTER READABLE INFORMATION - informacja w postaci czytelnej dla komputera,
COMPUTER SCIENCE - informatyka,
COMPUTER SCIENTIST - informatyk,
COMPUTER SERVICES CENTRE - centrum komputerowych usług obliczeniowych,
COMPUTER STORE - pamięć komputera,
COMPUTER SYSTEM - zestaw komputera,
COMPUTER THEORY - informatyka,
COMPUTER WORD - słowo komputerowe,
COMPUTER-BASED INFORMATION SYSTEM - skomputeryzowany system informacyjny,
COMPUTERIZATION - komputeryzacja,
COMPUTERIZED - skomputeryzowany,
COMPUTERIZED MANAGEMENT INFORMATION SYSTEM - system informatyczny zarządzania,
COMPUTING CENTER - ośrodek obliczeniowy,
CONCATENATE OPERATOR - operator sklejanego, operator spinania, operator konkatenacji,
CONCATENATION - scalanie (łączenie ciągów znaków za pomocą operatora),
CONCATENATION - wiązanie łańcuchowe,
CONCENTRATOR - koncentrator,
CONCEPTUAL - pojęciowy,
CONCEPTUAL DICTIONARY - słownik pojęć,
CONCLUDE - wnioskować, także: zakończyć, sfinalizować,
CONCLUSION - wniosek,
CONCURRENCY - zbieżność,
CONCURRENT - równoczesny, współdziałający, zbieżny,
CONCURRENT PROCESSES - procesy współbieżne,
CONCURRENT PROCESSING - wieloprogramowość,
CONCURRENT USER STACKS - stosy użytkowników (w systemie współbieżnym),
CONDENSATION - skrócenie (np. programu), skondensowanie,
CONDENSING ROUTINE - procedura zagęszczania,
CONDITION - warunek, także: stan, okoliczność,
CONDITION BOX - figura na schemacie blokowym symbolizująca warunek,
CONDITION CODE REGISTER - rejestr wskaźników,
CONDITION NAME - nazwa warunku,
CONDITIONAL - warunkowy,
CONDITIONAL BRANCH - rozgałęzienie warunkowe,
CONDITIONAL FIELD - pole warunków,
CONDITIONAL INSTRUCTION - rozkaz warunkowy,
CONDITIONAL JUMP - skok warunkowy,
CONDITIONAL JUMP INSTRUCTION - rozkaz skoku warunkowego,
CONDITIONALLY - warunkowo,
CONDUCT - przewodzić (np. prąd elektryczny), także: wykonywać (np. działania arytmetyczne),
CONDUCTOR TRACK - ścieżka przewodząca,
CONFERENCE ON DATA LANGUAGE - język programowania dostosowany do potrzeb zarządzania,
CONFIGURATION - dobór, zestaw, komplet, konfiguracja, także: układ, ukształtowanie, kombinacja,
CONFORMITY CERTIFICATE - atest,
CONFUSION - zamieszanie, chaos,
CONJUGATE - sprzężony,
CONJUNCTION CIRCUIT - układ koniunkcji,
CONNECT TOGETHER - łączyć ze sobą razem,
CONNECTING LINK - element łączący,
CONNECTION - połączenie, także: związek (pomiędzy),
CONNECTION LAYOUT - schemat montażu,
CONNECTION SET-UP - nawiązywanie połączenia,
CONNECTIVE - łącznik (logiczny), element łączący,
CONNECTIVE LINE - linia łącząca,
CONNECTOR - łącznik, wtyk, także: symbol na schemacie blokowym, odsyłacz na schemacie blokowym,
CONNOTATION OF A CONCEPT - treść pojęcia,
CONSECUTIVE - kolejny, następny,
CONSECUTIVE COMPUTER - komputer sekwencyjny,
CONSECUTIVELY - kolejno,
CONSEQUENT DRIVEN - wnioskowanie wsteczne (w systemach ekspertowych),

CONSIDERABLE DIFFERENCE - znaczna różnica, istotna różnica,
CONSISTENCE CHECK - sprawdzian logicznej zgodności,
CONSOLE - konsola,
CONSOLE COMMAND PROCESSOR - moduł, który interpretuje i wykonuje polecenia wprowadzane przez użytkownika za pomocą klawiatury,
CONSOLE MODE - bezpośredni tryb pracy "z konsoli",
CONSOLIDATION - scalanie, łączenie, konsolidacja,
CONSOLIDATOR - program scalania (łączenia), konsolidator,
CONSTANT - stała,
CONSTANT AREA - część pamięci zawierająca dane stałe,
CONSTITUENT - element, składnik,
CONSTRAINT - ograniczenie, warunek ograniczający,
CONTACT - styk, zestyk,
CONTENT - zawartość, także: stan licznika, rejestru,
CONTENT ADDRESSABLE MEMORY - pamięć adresowana zawartością, pamięć asocjacyjna, pamięć skojarzeniowa,
CONTENTS OF THE COUNTER - zawartość licznika, stan licznika,
CONTEXT-FREE LANGUAGE - język bezkontekstowy,
CONTIGUOUS - przyległy, sąsiedni,
CONTINUOUS - ciągły, nieprzerwany,
CONTINUOUS DRAWING - rysunek ciągły (elementy składowe są opisane współrzędnymi względnymi z wyjątkiem punktu początkowego),
CONTINUOUS FORM - papier ciągły, formularz ciągły,
CONTINUOUS FORM SET - zestaw formularzy ciągłych,
CONTINUOUS STATIONARY - papier rolkowy do drukarek,
CONTINUOUS TAPE - pętla taśmy dziurkowanej,
CONTRACTING A FILE - zmniejszanie pliku (zbioru),
CONTRAST - kontrast,
CONTRAST WITH ... - przeciwstawiać (coś czemuś),
CONTROL - sterowanie, kierowanie, także: kontrola, regulowanie,
CONTROL AND READ-ONLY MEMORY - pamięć stała ze sterowaniem,
CONTROL AND SIMULATION LANGUAGE - symulacyjny język programowania,
CONTROL AREA - programowy obszar sterowania pliku (zbioru),
CONTROL BALL - manipulator kulowy,
CONTROL BLOCK - blok sterujący,
CONTROL BOARD - pulpit sterowniczy,
CONTROL BRUSH - szczotka sterująca,
CONTROL CHANNEL - kanał sterujący,
CONTROL CHARACTER - znak sterujący,
CONTROL COMMAND - rozkaz sterujący,
CONTROL CONSOLE - pulpit sterowania,
CONTROL DEVICE - urządzenie sterujące,
CONTROL INFORMATION - informacja sterująca, rozkaz sterujący,
CONTROL JET - strumień sterujący,
CONTROL KEY - klucz sterujący, klawisz sterujący,
CONTROL LOOP - pętla sterowania, pętla regulacji,
CONTROL MEMORY - pamięć sterująca,
CONTROL MODE - tryb sterowania,
CONTROL PROGRAM FOR MIKROKOMPUTERS - program sterujący dla mikrokomputerów,
CONTROL QUANTITY - wielkość sterująca,
CONTROL RECORD - rekord sterujący,
CONTROL ROUTINE - część sterująca (programu organizacyjnego),
CONTROL SIGNAL - sygnał sterujący,
CONTROL STATEMENT - instrukcja sterująca,
CONTROL SYSTEM - układ sterowania, układ regulacji,
CONTROL TOTAL - suma kontrolna,
CONTROL TRANSFER INSTRUCTION - rozkaz skoku,
CONTROL UNIT (CU) - układ sterowania,
CONTROLLABILITY - sterowność, sterowalność,
CONTROLLABLE - regulowany, sterowany,
CONTROLLED CONDITION - wielkość regulowana, parametr regulowany,
CONTROLLED PLANT - obiekt regulacji,
CONTROLLED VARIABLE - wielkość regulowana, parametr regulowany,
CONTROLLER - regulator, urządzenie sterujące, sterownik, kontroler,
CONTROLLING - sterowanie,
CONVENTIONAL - konwencjonalny, klasyczny, typowy, powszechnie stosowany,
CONVENTIONAL DATA PROCESSING - przetwarzanie danych konwencjonalne,
CONVENTIONS - symbole, umowne znaki, także: typowe procedury w programach,
CONVERGENCE - zbieżny,
CONVERSATIONAL LANGUAGE - język konwersacyjny,
CONVERSATIONAL MODE - konwersacyjny tryb współpracy,
CONVERSATIONAL OPERATION - praca konwersacyjna, praca interakcyjna,
CONVERSATIONAL STATION - stacja konwersacyjna,
CONVERSATIONAL TERMINAL - terminal konwersacyjny,
CONVERSION - konwersja, przekształcenie, także: przechodzenie z jednego systemu na drugi,
CONVERSION PROGRAM - program konwersji,
CONVERSION TIME - czas konwersji,

CONVERT - przekształcać, przeliczać, także: przetwarzać,
CONVERTER - konwerter, przemiennik, przetwornik,
CONVERTER CONTROL PROGRAM - program sterujący konwertera,
CONVERTING - przekształcanie,
CONVEY - komunikować, przemieszczać,
CONVEYER - przenośnik,
COORDINATE GRAPHIC - grafika konturowa,
COORDINATE INDEXING - indeksowanie koordynujące,
COORDINATE ORIGIN - początek układu współrzędnych,
COORDINATE PLOTTER - pisak X-Y; kreslak,
COORDINATE SYSTEM - układ współrzędnych,
COORDINATES - współrzędne,
COPY - kopia, odpis, także: kopiować, przepis (jako instrukcja),
COPY PROTECTION PROGRAM - program zabezpieczający przed kopiowaniem,
COPYABLE SOURCE DOCUMENT - oryginał przystosowany do powielania,
COPYING - kopiowanie,
COPYING ROUTINES - programy kopiowania,
CORD - sznur, także: przewód elektryczny,
CORE - rdzeń, rdzeń ferrytowy, także: rdzeniowa pamięć operacyjna komputera,
CORE ARRAY - maczyca rdzeni,
CORE IMAGE LIBRARY - biblioteka programów przystosowanych do załadunku,
CORE LAYOUT - schemat rozdysponowania pamięci,
CORE MEMORY - pamięć rdzeniowa, pamięć magnetyczna ferrytowa,
CORE STORE - pamięć rdzeniowa, pamięć magnetyczna ferrytowa,
CORNER OUT - ścięty rożek (karty),
CORRECT - prawidłowy, poprawny, także: poprawiać, wprowadzać poprawkę,
CORRECTABLE - dający się poprawiać (sprostować),
CORRECTABLE CODE - kod umożliwiający korygowanie,
CORRECTION - poprawka, wniesienie poprawki,
CORRECTION STATEMENT - instrukcja korygowania,
CORRECTIVE MAINTAINING - usuwanie awarii w maszynie,
CORRECTOR - korektor, urządzenie prostujące,
CORRELATION - korelacja, współzależność, wzajemne powiązanie,
CORRELATION NET - siatka korelacji,
CORRESPONDENCE - odpowiedniość, także: stosowność,
CORRESPONDING - odpowiedni,
CORRUPTION - zepsucie,
COSTS PER BYTE - koszty zapamiętywania bajtu,
COUNT - liczenie, obliczanie, zliczanie, także: stan licznika, liczyć, obliczać,
COUNT-DOWN COUNTER - licznik poprzednikowy (zliczający "do tyłu"),
COUNT-UP COUNTER - licznik następnikowy (zliczający "w przód"),
COUNTABLE - policzalny, przeliczalny,
COUNTABLY INFINITE SET - zbiór przeliczalny,
COUNTDOWN - praca licznika powodująca odejmowanie,
COUNTER - licznik, mechanizm zliczający,
COUNTER CAPACITY - pojemność licznika,
COUNTER REGISTER - rejestr liczący, rejestr zliczający,
COUNTER TIMER CIRCUIT - programowany zegar,
COUNTERCLOCKWISE - odwrotnie do ruchu wskazówek zegara,
COUNTERPART - odpowiednik,
COUNTING - liczenie, obliczanie, zliczanie,
COUNTING BOARD - liczydło,
COUNTING CIRCUIT - układ liczący,
COUPLING - sprzężenie, połączenie,
COUPLING ELEMENT - element sprzęgający,
CPM - patrz: CRITICAL PATH METHOD,
CPU - patrz: CENTRAL PROCESSING UNIT,
CRAM - patrz: CARD RANDOM ACCESS MEMORY,
CRC - patrz: CYCLIC REDUNDANCY CHECK CHARACTER,
CREATING ADDITIONAL SYSTEM CONTROL AREA ROUTINES - programy poszerzania systemowego obszaru sterowania,
CREATION DATE - data zapisu,
CRISS-CROSS - krzyżujące się (przecinające się) linie,
CRITERION - kryterium,
CRITICAL PATCH METHOD - metoda krytycznej ścieżki,
CROM - patrz: CONTROL AND READ-ONLY MEMORY,
CROSS FEETS - działanie arytmetyczne polegające na dodawaniu i odejmowaniu,
CROSS PROTECTION - zabezpieczanie krzyżowe,
CROSS SOFTWARE - oprogramowanie skrośne,
CROSS-REFERENCE - odsyłacz,
CROSSBAR SWITCH - przełącznik krzyżowy,
CROSSCHECK - sprawdzanie poziome, sprawdzanie dwoma różnymi sposobami,
CROSSWISE - krzyżowo,
CRT - patrz: CATHODE-RAY TUBE,
CRT DISPLAY - monitor ekranowy, ekranopis,

CRT RASTER DISPLAY - monitor rastrowy,
CRYOGENIC MEMORY - pamięć kriogeniczna, pamięć nadprzewodnikowa,
CS (CODE SEGMENT) - segment programu,
CSL - patrz: CONTROL AND SIMULATION LANGUAGE,
CSMA-CD - patrz: CARRIER-SENSE MULTIPLE-ACCESS WITH COLLISION-DETECTION,
CTI - patrz: COAXIAL TRANSCEIVER INTERFACE,
CU - patrz: CONTROL UNIT,
CUBE - sześcian, także: trzecia potęga,
CUMULATE - kumulować, nagromadzać,
CUMULATIVE DATA - dane zbiorowe,
CURRENCY SIGN - znak waluty,
CURRENT - prąd elektryczny, także: bieżący,
CURRENT AWARENESS SERVICE - bieżąca służba informacyjna,
CURRENT DIRECTORY - katalog bieżący (aktualnie otwarty),
CURRENT INFORMATION - informacja bieżąca, informacja ciągła,
CURRENT LINE - linia bieżąca (aktualnie zapisywana lub błędna),
CURRENT POINT - bieżący punkt odniesienia,
CURRENT POSITION - pozycja bieżąca,
CURRENT PROGRAM - program bieżący (aktualnie w pamięci),
CURRENT SEGMENT - segment bieżący (zdefiniowany ostatnią instrukcją DEF)
CURRENT VIEWPORT - wziernik bieżący (aktualnie używany),
CURSOR - kursor, wskaźnik pozycji,
CURSOR CONTROL KEYS - klawisze sterujące kursorem,
CURTATE LOWER - dolna część karty 80-kolumnowej z wierszami 1-9,
CURTATE UPPER - górna część karty 80-kolumnowej z wierszami 10,11,12,
CURVE - krzywa, krzywa charakterystyczna,
CURVE GENERATOR - generator krzywych,
CURVE PLOTTER - pisak krzywych,
CUSTOMER - nabywca,
CUSTOMIZED - także: przyjęty
CUT DOWN - redukować, zmniejszać,
CUT OFF - zatykać, odcinać,
CX (COUNT REGISTER) - rejestr zliczający (pętli),
CYAN - niebiesko-zielony,
CYBERNATION - komputeryzacja,
CYBERNETIC ENGINEERING - cybernetyka techniczna,
CYBERNETIC MODEL - model cybernetyczny,
CYBERNETICS - cybernetyka,
CYCLE - cykl, także: okres,
CYCLE CARRY - przeniesienie cykliczne (z najwyższej pozycji liczby do najniższej),
CYCLE COUNT - liczenie cykli,
CYCLE COUNTER - licznik cykli,
CYCLE STEALING - zabieranie cykli,
CYCLE TIME - czas cyklu,
CYCLES PER SECOND - liczba cykli na sekundę,
CYCLIC - cykliczny, okresowy,
CYCLIC CODE - kod cykliczny,
CYCLIC PERMUTED CODE - kod cykliczny permutowany,
CYCLIC PROGRAMMING - programowanie cykliczne,
CYCLIC REDUNDANCY CHECK CHARACTER (CRC) - znak CRC, znak kontroli cyklicznej,
CYCLIC SHIFT - przesunięcie cykliczne,
CYCLICAL REDUNDANCY CHECK - cykliczna kontrola redundancji,
CYLINDER - cylinder,
CYLINDER ADDRESS - adres cylindra,
CYLINDER INDEX - indeks cylindra,
CZ-13 FONT - pismo CZ-13,

D

DAILY DATA - dane dzienne,
DAISY-CHAIN - układ łańcuchowy, rozproszony układ przerwań skierowanych,
DAISY-WHEEL PRINTER - drukarka z wirującą głowicą, drukarka rozetkowa,
DAMAGE - uszkodzenie, awaria, także: uszkodzić,
DAMAGEABLE - podatny na uszkodzenia,
DARK TRACE TUBE - lampa z ciemnym śladem (strumień elektronów daje ciemny ślad na ekranie),
DAS - patrz: DATA ACQUISITION SYSTEM,
DASH - kreska, znak łamania,
DASHED LINE - linia kreskowa,
DATA - dane, także: wartość,
DATA ACCEPTED - potwierdzenie odczytu danych,
DATA ACQUISITION - gromadzenie danych, zbieranie danych,
DATA ACQUISITION SYSTEM - układ gromadzenia danych,
DATA AVAILABLE - sygnał gotowości danych,

OŚRODEK BADAŃ I EKSPERTYZ INŻYNIERII SYSTEMÓW

Polskiego Towarzystwa Cybernetycznego w Poznaniu
organizuje kursy z zakresu użytkowania i oprogramowania
mikrokomputerów SPECTRUM+ i IBM PC:

1. Kurs podstawowy dla dyrektorów i kierowników przedsiębiorstw — 20 godz. 10 000 zł
2. Użytkownik-operator mikrokomputera — 60 godz. 15 500 zł
3. Programowanie w języku BASIC — 100 godz. 26 900 zł
4. Projektowanie i oprogramowanie systemów informatycznych w technologii dBASE III — 140 godz. 36 000 zł

Zajęcia odbywać się będą w Poznaniu. Terminy rozpoczęcia kursów podamy przed uruchomieniem kursu.

Zgłoszenia prosimy kierować pod adresem: Ośrodek Badań i Ekspertyz Inżynierii Systemów Polskiego Towarzystwa Cybernetycznego ul. 28 Czerwca 231/239, 60-915 Poznań, budynek TASKO tel. 32-12-41, telex 0413353 w godz. 8.30—11.30

OGŁOSZENIA

Comodore 16/116/+4 — programy — wymienię. Jacek Rubinkowski, 09-400 Płock, ul. Orlińskiego 3 m 20

Wideo niesprawne zamienię na mikrokomputer lub sprzedam. Andrzej Peta, Okumino, 77-203 Dretyn, woj. Słupsk

Programy na C-64 sprzedam. Marek Rudnicki, 51-376 Wrocław, ul. Kłokoczycka 154

Comodore 64 — chętnie wymienię się oprogramowaniem. Posiadam około 400 programów i gier. Alexander Stempel, 40-160 Katowice 2, ul. Armii Czerwonej 104/1

Kupię TIMEX 2020. Michał Praceł, 81-116 Gdynia, ul. Bosmańska 28/12

Sprzedam Commodore 64 ze stacją 1541 i magnetofonem 1530 lub samą stacją. Piotr Węclawik, 71-015 Szczecin, ul. Osikowa 26, tel. 82-41-82

Wymienię programy na Commodore 64. Radosław Dubiella, 75-347 Koszalin, ul. Władysława IV 58a/37, tel. 529-36

Pocztowa giełda

Rozwiązanie krzyżówki z nr. 7/8 „IKS-a”.
Hasło brzmi: „Kluby komputerowe młodych mistrzów techniki”.

Bony pieniężne (1000 zł) wylosowali: Irena Pietruszewicz — Poznań, Dariusz Sidel —

Kielce, Dorota Noworyta — Zielonki, Konrad Kwiatkowski — Jedlicze, Sławomir Załęcki — Łask.

Nagrody książkowe otrzymują: Wojciech Stefaniak — Włocławek, Klementyna Dembek — Toruń, Maria Ramian — Tarnów, Małgorzata Kaliszan — Murowana Goślina, Wiesław Kołder — Ustroń, Maciej Trąd — Zielona Góra, Marian Borawski — Lubaczów, Wojciech Abramczuk — Sokołów Podlaski, Andrzej Piela — Zabrze, Artur Rapacz — Legnica.

Rozwiązanie krzyżówki z nr. 9/87 „IKS-a”.
Hasło brzmi: „Pamiętajmy o komputerze nie tylko na co dzień ale i w święta”.

Bony PKO (1000 zł) wylosowali: Jerzy Trzaska — Zawadzkie, Jarosław Kędrak — Częstochowa, Andrzej Tadewicz — Lublin, Anna Rynkiewicz — Dolistowo, Grażyna Stefanek — Lublin.

Nagrody książkowe otrzymują: Danuta Bamber — Margonin, Mariusz Zacharkiewicz — Buraków k. W-wy, Weronika Kowalska — Sosnowiec, Sewer Nyckowski — Gostynin, Tadeusz Dziugiel — Ruda Śl., Jacek Szczeniński — Zielona Góra, Alicja Kokot — Częstochowa, Wojciech Pilski — Gdańsk, Jarosław Murenia — Gorzów, Piotr Bakun — Janów Lub.

Rozwiązanie krzyżówki z nr. 10/87 „IKS-a”.
Bony pieniężne (1000 zł) wylosowali: Elżbieta Mazurowska — Szczecin, Dariusz Lubawy — Gniezno, Edward Chmielak — Kłodzko, Ag-

nieszka Jastrzab — Dąbrowa Górnicza, Aleksander Wierzbicki — Białystok.

Nagrody książkowe otrzymują: Jacek Wychowaniec — Sosnowiec, Waclaw Skrzypczak — Warszawa, Krzysztof Chudy — Szczecin, Janusz Midura — Pszczyna, Henryka Kaczmarek — Radom, Tomasz Żak — Wrocław, Maciej Ziętek — Poznań, Jolanta Pogorzelska — Gdańsk, Stefan Kubowicz — Kraków, Jan Strzałka — Stargard Szczeciński.

Rozwiązanie krzyżówki z nr. 11/87 „IKS-a”.
Bony pieniężne (1000 zł) wylosowali: Michał Zimnicki — Malbork, Włodzimierz Plewa — Gdańsk, Grzegorz Romanowski — Łomża, Jerzy Wieloch — Poznań, Krzysztof Michalski — Elk.

Nagrody książkowe otrzymują: Leszek Pniak — Siemianowice Śl., Leszek Bakalarczyk —

Bydgoszcz, Tomasz Woźny — Poznań, Marek Wachel — Kraków, Radosław Łepk — Gdynia, Marek Lasecki — Warszawa, Tadeusz Mamak — Kraków, Piotr Kalewicz — Gdynia, Andrzej Bałec — Olkusz, Aleksander Stochalski — Kraków.

Rozwiązanie krzyżówki z nr. 12/87 „IKS-a”.
Hasło brzmi: „Poznaj komputer — zyskasz przyjaciela”.

Bony pieniężne (1000 zł) wylosowali: Sławomir Bukowski — Pruszków, Grażyna Nowak — Kraków, Andrzej Pieszalski — Słupsk, Andrzej Herla — Głogówek, Dorota Radziwon — Torzyn.

Nagrody książkowe otrzymują: Janina Stefanowska — Bielsko-Biała, Danuta Stryczek — Kluczbork, Ewa Dobrucka — Jastrzębie-Zdrój, Andrzej Ćwikliński — Wrocław, Wiesław Luks — Poznań, Maciej Trólka — Czeładź, Barbara Raczkowska — Szczecin, Maciej Dusza — Warszawa, Robert Perłowski — Ziębice, Tomasz Salachna — Częstochowa.

OLIVETTI PC M380

Ten komputer pracuje na 32-bitowym mikroprocesorze 80386. Podobnie jak inne modele tej firmy jest programowo zgodny ze swoim odpowiednikiem z IBM. M 380 produkowany jest w trzech wersjach: **M 380 C** (Compact), **M 380 S** (Standard) i **M 380 T** (Tower). W ostatniej z nich pamięć RAM może być rozbudowana do 64 Mb!



OLIVETTI PC M380

KRZYŻÓWKA NR 2

WSPÓŁPRACUJE Z KOLA BORAVENTE	OPIEKA PRAWNA	KOBIETA Z KAREMĄ FILMOWĄ	IMIĘ AUTORA POW. „GRYPA SZALEJE W NAPRAWIE”	WENECKI GEOGRAF, PIONIER HISTORII GEOGRAFII /1485-1553/	ATROFIA	GRECKA BOGINIA WYSTĘPKÓW	MALINO -WY	
			LOG LUB LG	KONPAN LOLKA	MUCHA	NA MIN-DANAD ANDRYCZ		
UDANY EFEKT POLDHAWA	K	O	L	A	B	O	R	A ¹⁴
ZBIÓR PRZE-PIŚÓW	U	P	O	L	O	W	A	N
ZJAWA	R ²	E	G	U ⁵	L	A	M	I
	M	A	R	A	E	D	U ⁹	K
POŚCIEL	T	A	R					
	B	E	I	Y				
PIASIE PIÓRO	L	O	T	K	A			
DZIAŁO	A	R	M ⁶	A	T	A		
NA GŁOWIE MOTOCYKLISTY	K			T	E ³	L	E	W
MOŻE BYĆ Z DZIAŁ	K	A	N	O	N	A	D	A
STRÓŻ KOBIECY Z INDI	A			W	A			
KŁODA	S	A	R ⁸	I				
SZTUKA UKŁADANIA KWIATÓW	I ¹¹	K	E	B	A	N	A	

Litery z kratek ponumerowanych od 1 do 14 dadzą hasło, które wystarczy nadesłać (jako rozwiązanie zadania) pod adresem redakcji na kartach pocztowych w terminie do końca kwietnia, naklejając kupon „IKS”-a. Wśród autorów prawidłowych odpowiedzi rozlosujemy bony pieniężne i nagrody książkowe.

KUPON ►



COMMODORE PC 10 i PC 20



PC 10 i PC 20 zbudowane zostały na Intelu 8088. Pierwszy z nich ma RAM 512 Kb drugi 640 Kb. Oba pracują pod systemem operacyjnym MS DOS 3.2. Zasadniczą różnicą jest dysk twardy (20 Mb) zainstalowany w modelu PC 20.

OLIVETTI PC M240



Zbudowano go na mikroprocesorze 8086. M240 dysponuje pamięcią RAM 640 Kb. Standardowo wyposażony jest w jedną stację dysków miękkich (5 1/4 cala). Istnieją oczywiście możliwości rozbudowy konfiguracji podstawowej. Można ją wzbogacić między innymi o dyski miękkie i 20 Mb dysk twardy. Zaletą tego modelu Olivetti jest jego programowa zgodność ze standardem IBM.

Systemem operacyjnym jest MSDOS 3.2. M240 pobiera w czasie pracy moc 183 W (napięcie sieci zasilającej 220 V ± 10%).

„IKS” — dodatek „Żołnierza Wolności”. Redaguje Wiesław Cetera (kierownik zespołu); Rada programowa: Krzysztof Chmarra, Romuald Głęb, Włodzimierz Gogolek, Janusz Janiec, Henryk Krasuski, Ireneusz Miernik, Ludwik Piela, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77, Fotoskład i druk offsetowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 9564. Nr. ind. 361682.