

*и фран.*

# КОМПЮТЪР

Издание на ЦК на ДКМС

# ЗА ВАС

4'86 Година Втора Цена 0,60 лв.

ISSN-0205-1893



СПИСВА РЕДАКЦИЯ  
ОРБУИ

*Скениране и обработка:*

*Антон Оруш*

*[www.sandacite.net](http://www.sandacite.net)*

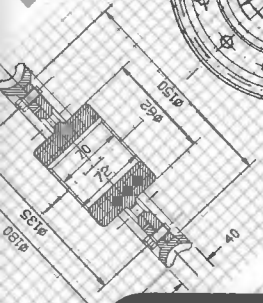
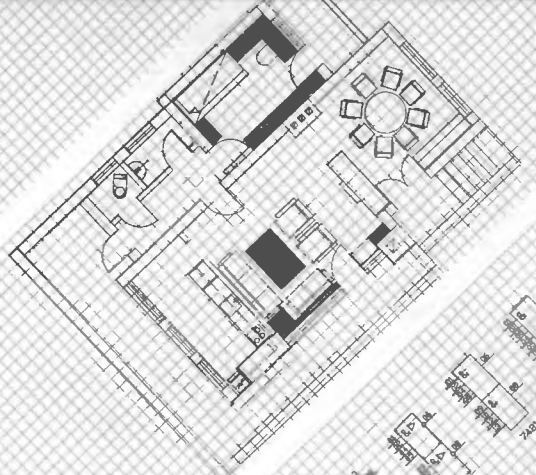
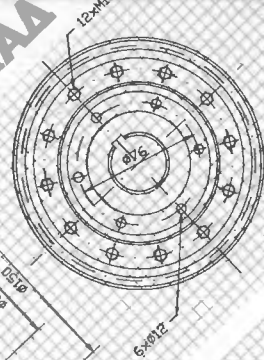
*[deltichko@abv.bg](mailto:deltichko@abv.bg)*

*0896 625 803*



**ФОРУМ  
САНДЪЦИТЕ**

ТЕХНОКАД



### Програмен продукт ТЕХНОКАД

ТЕХНОКАД е специализирана графична програма, предназначена за автоматизация на инженерно-конструкторския труд. Със своите многобройни функции, с възможността за надстройки и допълнения ТЕХНОКАД е приложим в следните области на народното стопанство:

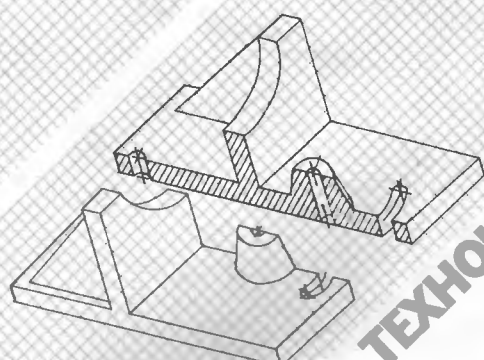
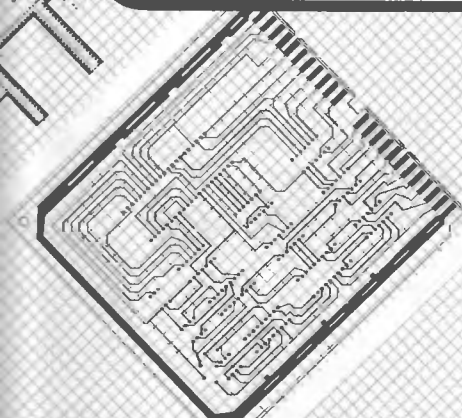
- машиностроене;
- електроника и електротехника;
- архитектура;
- картография.

ТЕХНОКАД дава възможност на потребителя да изгражда свои библиотеки от примитиви, видове шрифтове и прекъснати линии. Автоматичното оразмеряване и шрихване, възможността за връзка с плотер и дигитайзер, двупосочната връзка с изчислителни програми (написани на BASIC и FORTRAN), връзката с програмни продукти за проектиране на печатни платки и NC програми съкращава многократно срока на проектиране и изработване на конструкторска и технологична документация.

Програмният продукт ТЕХНОКАД е разработен в СО "Програмни продукти и системи" от СППС - Габрово и се разпространява от комбинат "Комплектация и програмно-технически услуги".

Адресът е:

комбинат "К П Т У"  
1618 СОФИЯ  
ул. Й.Видоверадски 46  
тел. 56-28-21



ТЕХНОКАД



7411

Всеки партиен конгрес е своеобразен връх в новата история на България, отправна точка за нейното социалистическо изграждане. Тринадесетият ще остане в историята като конгреса, разтворил широко вратите към бъдния двадесет и първи век, очертал пътищата и подходите, осигуряващи място на страната ни в челната група, която на крилете на научно-техническата революция първа ще пресече финалната лента, разделяща столетията.

Българските комунисти, целият наш иарод разполагат с ясна и конкретна програма за действие, изложена във Възпитателното и Заключение-но слово на генералния секретар на партията др. Тодор Живков, в Отчетния доклад на ЦК на БКП, в решенията на конгреса. С партинна мъдрост, проникновение и откровеност, от висотата на тридесетгодишнината на априлската генерална линия на партията в тях е казано всичко, за да могат всяка организация, всеки член на нашето общество да видят своето място, да оценят реалния си принос за обществено развитие и направят необходимите изводи.

Така че сега са нужни не красиви думи и обещаващи намерения, а конкретни дела. Такъв е примерът, който ни дава партията.

Отговорни задачи стоят пред нашата младеж, като творец на научно-техническия прогрес и особено на неговите приоритети, стратегически направления.

Голяма е отговорността и на нашето списание, което работи на челния фронт на научно-техническата революция, компютеризацията на народното стопанство. Защото освен компютри са нужни и кадри, способни да работят с новата техника, кадри, които да разработват нови програмни продукти, да създават новите поколения компютри.

Предпоставка за ускорената подготовка на младежта за работа с компютрите е единствено то у нас специализирано издание по компютърна техника и информатика да достига до своите читатели, до всички, на които то е нужно. За съжаление обаче при сегашния му тираж това не става!

Иска ни се да отбележим още едно събитие, което в предконгресните дни остана като че ли малко в сянка. Най-младият член на фамилията Пraveц пое своя път към българския дом. Едва ли е нужно да казваме колко очакван и желан бе този ден. Защото, макар и домашен, той полага пъртината, която ще отведе децата и юношите към големите, професионалните компютри. Пraveц 8Д е липсващото досега звено в

системата за подготовка на младежта за работа с електронноизчислителната техника. Той няма да замести организираното обучение в клубовете „Компютър“ и училищата, но ще го допълни, защото една от най-ефективните форми на обучението, която не може да бъде заместена от друга, е самообучението, продължителната творческа работа насаме с компютъра!

Предконгресен рекорд в научно-техническото ни книгоиздаване поставиха авторският колектив на книгата „Домашен компютър Пraveц 8Д“ и издателство „Техника“. Тя бе подготвена и отпечатана за близо два месеца срок.

Появяването на домашния компютър поставя и редица проблеми. Преди всичко за него практически няма никакви програмни продукти, което поне на първо време го прави, общо взето, малко използваем. Изводът е повече от ясен и вярваме, че в най-скоро време нашата програмна индустрия ще запълни тази празнина. „Компютър за вас“ от следващия брой също започва редовно да публикува програми за Пraveц 8Д. Дано не се наложи да се червим пред читателите си за грешки в тях, защото засега ще разчитаме единствено на прецизността на авторите и консултантите си — редакцията ни не разполага с домашен компютър. Пак по същата причина засега не можем да открием на страниците на списанието темата за 16-битовите компютри и когато те се появят в масово производство, а това ще стане скоро, вероятно ще се оправдаваме с обективни причини!

Но да се върнем към домашния Пraveц. В този брой публикуваме статия, в която са разгледани различията му от Пraveц-82, за който има вече и литература, и предостатъчно програми. Така че човек, който познава Пraveц-82, лесно ще може да адаптира негови програми.

Нас обаче ни блазни идеята да публикуваме програма, която автоматично да „транслира“ бейсови програми от Пraveц-82 за Пraveц 8Д. Идеята ни е осъществима, след като разполагаме с програма (която скоро ще публикуваме) за превод на програми от Пraveц-82 за Комодор 64 и обратно.

Както и друг път, сега отново се обръщаме към нашия колективен автор, към нашите читатели, за разработването на такава програма. Ние пак ще се погрижим вложеният труд да бъде подходящо възнаграден.

Очакваме с нетърпение вашите разработки, драги читатели и сътрудници на списанието!

Ижж. ГЕОРГИ БАЛАНСКИ

# НЕВИДИМАТА ДИПЛОМА

НИКОЛАИ ДЮЛГЕРОВ

## В ЦЕНТЪРА НА ОБУЧЕНИЕТО

Район „Христо Ботев“ на столицата обхваща жилищните комплекси „Младост“, кварталите „Горубляне“ и „Полигона“, научно-промишлената зона „Изток“ и студентския град „Христо Ботев“. В него са концентрирани много аншеи учебни заведения и училища. Вузовете са вече шест, и то от най-големите, а училищата дванадесет, и се строят още три.

Тук всяка година завършват всъщност си образование над 20 хиляди студенти от страната и чужбина. Това е базата за прозорливото решение на градския и районния комитет на Комсомола да се създаде районен младежки клуб „Компютър“ на територията на студентския град.

Клубът започва дейността си през януари миналата година и бързо стана популярен сред младежите. Ръководството е възприело стратегията за широк достъп до компютърната техника. Всеки, който пожелае, може да започне обучение по един от програмните езици от високо ниво — Бейсик, Паскал, Асемблер, Лого или друг. По думите на ръководителя на клуба инж. Пантелей Кочорапов, „колкото по-малко бездействава техниката, толкова по-добре за всички“.

Второто главно съображение при избора на мястото за този клуб е концентрацията на много научноизследователски и производствени звена. ВМЕИ „В. И. Ле-

- Столичният район „Христо Ботев“ — огнище на I и II грамотност
- Компютризация без компютри не става — или как бе събрана техниката
- Комсомолски завод за софтуер
- Формулата на успеха: динамични програмни колективи

нини“, БИИ „Карл Маркс“, Институтът по микроелектроника, Централният институт по изчислителна техника, Институтът по приборостроене, ДСО „Изот“, ДО „Изотсервна“, Заводът за електронни преобразователни елементи и други помагат с каквото могат — едни с валута, други с техника, трети с кадри.

Първите 15 персонални компютри Правец-82 подарява Министърството на машиностроенето, шест броя Изот-1031 — ДСО „Изот“, а трите броя Интелекст клубът купува със средства, отпуснати от Съвета за висше образование към МНП. Те са напълно съвместими с IBM PC/XT и новите български Правец-16 и Изот-1036, комплектувани са със запаметяващи устройства тип „Уичестър“, печатащи устройства, цветни монитори и т. и. Освен тях в момента се използват и седем Правец-8Е, получени за временно ползване от ИВСД „Авангард“.

Колкото повече и по-разнообразна е техниката, толкова по-трудно се поддържа. И ако този проблем владява ръководствата

на много други клубове „Компютър“, тук той е решен. Обслужването е прието от сервисната база на ВМЕИ „В. И. Ленин“, а там са все специалисти от голяма класа, които разполагат с необходимата диагностична апаратура. И което е по-важно, проявяват пълна отзивчивост към желанието на младите хора да овладяват новите знания.

Ще отбележим още подчертаното желание на научните звена от района за сътрудничество с клуба, проява не на благотворителност, а на далновидност. Няма по-добра инвестиция от вложението средства за подготовка на висококвалифицирани млади кадри, притежаващи едновременно

## ИНЖЕНЕРНА И КОМПЮТЪРНА ГРАМОТНОСТ

Заедно с основното свидетелство за висша зрелост, получено във ВУЗ, членовете на РМК „Компютър“ ще отнесат със себе си и една невидима диплома за знанията си по информатика — га-

ранция, че ще се справят с новата компютърна техника по своите работни места.

Обикновено клубовете подготвят членовете си как да работят с компютри и програмни продукти, а тук освен това (не случайно ни го посочиха от ИВСД „Авангард“ за еталон) възпитаат а младите и уменията да прилагат новия подход за динамично решаване на динамичните задачи на нашата бързо развиваща се икономика.

Това означава, че тук навреме са доловили клубовете на новото, което все повече ще се утвърждава. За всички е ясно, че бързото навлизане на компютрите в нашия живот изпреварва реалните възможности на традиционните институти за подготовка на знаещи и можещи кадри. И имен-

ложни програми за Завода за магнитни дискове в Пазарджик, разработен и вече внедрен от съвместен колектив — специалисти от завода и студенти от клуба. Завърши и разработката по договор с АЕЦ „Козлодуй“ за създаване и внедряване на автоматизирана система за управление на кабелното стопанство на централата.

В началото на годината бе подписан договор и създаден програмен колектив по задача на Инженерната стопанска организация „Вюинвест“. Поръчката изисква да бъдат разработени 20 информационно-рекламни продукти — асъщност цяла информационна система на равнището на най-доброто в света. На различни езици тя ще представи ИСО „Вюинвест“ при участието ѝ в 25

добри познания в програмни работи, студента и учейца.

## ЗАБЕЛЯЗВАТЕ ЛИ НОВИЯ МОМЕНТ?

В клуба за всяка конкретна задача се подбира най-подходящите специалисти, а не както често се случва, за колектив с постоянен състав се търси подходяща задача. Това е гъвкавият подход — динамични програмни колективи. А те са истински жив и жизнеспособен организъм — един специалисти например изготвят технико-икономическото задание и напускат колектива, за да решават следващи задачи, а на тяхно място се включват други, които разработват програмите. След приключване на даден договор, когато задачата е изпълнена, колективът се разформирава и специалистите се пренасочват към нови задачи.

Ако сте останали с впечатлението, че всичко върви по мед и масло, това не отговаря на истината. И тук, както навсякъде, където се проправя път, възникват проблеми, но не там поставяме ударението.

Ние търсиме чертите на новото — и ги намерихме.

Една година и три месеца, колкото е възрастта на клуба, е малък срок, но в тази динамична област — информатиката, той е съвсем достатъчен за оценка. Действително РМК „Компютър“



но това е, според мен, главно постижение на РМК „Компютър“ — Ботевски район. В него се подготвят не само добри специалисти, а и бъдещи ръководители на бъдещи клубове.

## НАУКА И ПРОИЗВОДСТВО

Макар че не са най-важното от дейността на клуба, договорите се появяват закономерно, породени от реална обществена необходимост. Тяжното съществуване е обществената оценка за равнището, достигнато от младите творци. От една страна, младите затвърдяват своите знания и умения в практиката, а от друга — към клуба се насочват държавни организации и ведомства, които поръчват изпълнението на един или друг програмен или хардуерен продукт и подкрепят доверието си с финансови средства. Тези средства стимулират колективите, подпомагат развитието на материалната база на клуба и базата за младежки дейности в района.

Досега са сключени седем договора на обща стойност над 80 000 лева. Между тях е пакетът при-



международни изложби и панаир през тази година.

Може би най-значимият програмен продукт се разработва по поръчка на ИВСД „Авангард“. Това е пакетът приложни програми „Автоматизирано управленско място“, в който са включени графичен редактор, статистика, „Електронна картотека“, „Контрол на решенията“, текстообработваща програма, текстова програма и т. н. Колективът, който се ръководи от преподаватели във ВИИ „Карл Маркс“, включва специалисти икономисти с много

в студентския град заслужава най-висока оценка! Защото ръководителите и участниците в него заедно с ръководството на районния център за ТНТМ и РК на ДКМС правилно са уловили модерните тенденции в съвременната организация на науката в тази бурно развиваща се област — компютризацията, правилно са очертали линиите на развитие, тясно свързани с поставянето на нови основи на стопанството и икономиката ни, диктувани от развитието на научно-техническия прогрес в нашата страна.

# КОМПЮТЪРЪТ

Пред образователната система стои задачата за широко използване на електронноизчислителната техника в учебния процес. През изминалите две-три години беше натрупан известен опит в областта на програмното осигуряване за учебни цели. Много е направено за анализа на чуждия опит. Всичко това ни дава основание да твърдим, че е напълно моментът за преосмисляне на постигнатото.

Преди всичко трябва ясно да различаваме ролята на компютъра като предмет и като средство за обучение.

В използването на компютъра като предмет на обучение има натрупан значителен опит, особено във ВУЗ, и може да се твърди, че проблемът като цяло е решен. Като особено ефективно средство се очертава т. нар. *автоматизирана среда за обучение (АСО)*. Тя представлява комплекс от специализирани за обучението трансформатори, операционни системи, редактори, бази данни, верификационни програми, системи за документирани и други.

Значително по-трудно е компютърът да се използва като средство за обучение. Трудността произлиза от качествено новия характер на това средство. Чрез компютъра ученикът може да бъде научен самостоятелно да анализира конкретната обстановка, да формулира задачата, да подбере средства за нейното решаване, да определи оптималната стратегия на решението и да докаже верността му.

Ала компютризицията има и обратна страна. При неправилен подход използването на компютъра може да доведе до нарушаване на връзките на човека с обществото, стесняване на културата, технократизъм, шаблонност на мисленето и дори до нарушения на психическото и физическото здраве на обучаемия.

За да се постигнат добри резултати и се избегнат аредните последици, трябва да се спазват следните основни принципи (без претенции за изчерпателност):

● Въвеждането на компютри в обучението не е самоцел. Компютърът

**В**

# ОБРАЗОВАНИЕТО

трябва да се приобщи към системата учител — ученик — общество, без да подменя нито един от нейните компоненти.

● Като нов активен елемент на учебния процес компютърът изисква преосмисляне и пренастройка на всички страни на този процес: учебно съдържание, методика, документация, организация, разпределяне на времето, норми на поведение, подготовка на преподавателите. В идеалния случай програмните продукти трябва да се разработват едновременно и да бъдат неделима част от учебника и останалата учебна документация.

● Компютризицията трябва да се съпроводи от засилване на водещата роля на учителя, от укрепване на неговия авторитет и значение като възпитател, съветник и приятел. Това изисква повишаване на професионалната подготовка на учителя.

● Използването на компютъра има смисъл само при осигуряване на активна позиция на ученика. Внимателно следва да бъдат отчетени изменените отношения в учебническия колектив.

● Програмните продукти трябва да имат положително естетическо въздействие и да повишават езиковата култура.

● Използването на компютъра трябва да бъде съобразено с психическите и физическите особености на дадената възрастова група и дори на дадения учебнически колектив.

● Компютърът не бива да измести учителя, а напротив — да стимулира самостоятелната работа с книгата.

● Компютърът трябва да се ползва преди всичко за постигане на конкретни цели, които са недостижими с други средства.

● Новите знания, придобивани по време на работа с компютъра, трябва да бъдат открити, „настрадани“ от ученика, а не поднесени наготово.

● С компютъра трябва да се решават задачи, изискващи знания и умения по различни учебни дисциплини.

● При оценка на резултатите от използването на компютъра трябва да

К. Т. Н. инж. НЕДЯЛКО  
ТОДОРОВ  
директор на НИПЛ  
„Програмно осигуряване“ при  
СУ „Климент Охридски“

се отчетва неговата първоначална отговорност и да се планира преобладаващ спад на интереса.

● Програмните продукти за учебни цели трябва да бъдат добре защитени от случайни грешки или умислени опити за прекратяване на работата им, налучкване на отговори, „надзъртане“ в програмата в в информацията, предизначена за учителя.

● Програмните продукти за учебни цели трябва да бъдат гъвкави, т. е. да генерират автоматично различни варианти на отделните задачи, да могат да бъдат настройвани на различни нива на трудност, обхват и т. н. Всеки продукт трябва да предоставя на учителя информация, достатъчна за оценка на индивидуалното протичане на учебния процес.

● Поддържащият диалог „ученик — програма“ трябва да бъде приятелски, максимално ясен и еднозначен. Ученикът трябва да може да използва компютъра в хода на учебния експеримент, при решаване на контролна задача и т. н.

● Програмният продукт трябва да представлява комплекс от програми, обединени от една идея и покриващи относително голям дял от учебния материал. Учителят трябва да има възможност да решава кога, как и доколко да използва продукта. Межличното пренасяне и слепото спазване на урочната система рязко намалява ефекта от използването на компютъра.

С други думи, като използваме новата техника, трябва да се ръководим от осъзнатата необходимост, а не от изискванията на модата.

Ще се опитаме да очертаем някои основни методи (съответно типове учебни програмни продукти), които биха позволили да спазим формулираните принципи.

Учебни игри с общо предизначение. Освоената цел е да се тренира уменията за откриване на закономерности в учебническа стратегия, да се достигне до идеи за алгоритъм, сте-

пен на влияние на отделни фактори, да се развива логическо и комбинативно мислене и устойчиво аниманне.

Учебни игри — тренажори. Основната цел е затвърждаването на конкретни знания и умения в игрова обстановка.

Специализирани учебни игри, в които е нужно да се формулират и решават комплексни задачи, изискващи знания от различни дисциплини.

Контролно-обучаващи програми — съчетават учебния експеримент с решаване на контролни задачи.

Учебно моделиране. Основна цел е осмислянето на сложни процеси и явления, достигането до абстрактни понятия, самостоятелното осъзнаване на влиянието на отделни компоненти върху поведението на сложни системи.

Учебни експерименти. Основна цел е създаването на умение за подбор на средствата на експеримента и за анализ на получените резултати.

Компютърни системи за решаване на конструктивни и технологични задачи. Основна цел е осмислянето чрез експеримент на понятията за технология и жизнен (производствен) цикъл, трениране на уменията за композиция на комплексни задачи.

Системи за поддържане на бази знания. Основна цел е компютърът да поеме рутинните действия при решаването на един проблем, като на обучаемия се предоставя творческо подбиране и прилагане на усвоените знания и намърането на оптимална стратегия на научното търсене.

Накрая трябва да се спрем и на следния въпрос. Много хора мислят, че всеки, който има малко свободно време и добро желание, може да разработва програмни продукти за учебни цели. Това е груба грешка. Всеки програмен продукт за учебни цели трябва да бъде плод на усилията на най-добрите педагози, методисти, информатици, психолози, лекари и т. н. В същото време производството на програмни продукти не трябва да се превръща в монопол. И най-вече, трябва да се помни, че ефективността на един програмен продукт се определя преди всичко от неговата методическа и педагогическа основа. И най-брилянтната програмистка техника не може да замени знанията и опита на педагога.

За повишаване на ефективността от приложението на микрокомпютърната техника във всички сфери на народното стопанство, за създаване на по-съвършени български програмни продукти

ЦК НА ДКМС  
ДСО "ПРОГРАМНИ ПРОДУКТИ И СИСТЕМИ"  
СП. "КОМПЮТЪР ЗА ВАС"  
В. "НАПРАВИ САМ"

ОБЯВЯВАТ

КОНКУРС

ЗА РАЗРАБОТВАНЕ НА СОБСТВЕНИ ПРОГРАМНИ ПРОДУКТИ В ОБЛАСТТА НА:

I - БАЗОВ СОФТУЕР ЗА 8- И 16-БИТОВИ МИКРОКОМПЮТРИ - ДОПЪЛНЕНИЯ И РАЗШИРЕНИЯ НА СЪЩЕСТВУВАЩИТЕ ОПЕРАЦИОННИ СИСТЕМИ, ТЕХНОЛОГИЧНИ СРЕДСТВА, ПОДПОМАГАЩИ ПРОГРАМИСТА В РАЗРАБОТВАНЕТО НА ПРОГРАМНИ ПРОДУКТИ, ГЕНЕРАТОРИ /ПРОГРАМИ, КОИТО ПРИ ЗАДАВАНЕ НА ПАРАМЕТРИТЕ, ГЕНЕРИРАТ КОНКРЕТНИ ПРИЛОЖНИ ПРОГРАМИ/, ПОМОЩНИ И ДРУГИ ПРОГРАМИ.

II - ПРИЛОЖЕН СОФТУЕР, ЗА РЕШАВАНЕ НА ЗАДАЧИ ОТ ВСИЧКИ ОБЛАСТИ НА НАРОДНОТО СТОПАНСТВО ЗА 8- И 16-БИТОВИ МИКРОКОМПЮТРИ.

III - ИГРИ ЗА 8- БИТОВИ МИКРОКОМПЮТРИ /ВКЛЮЧИТЕЛНО И ЗА ПРАПЕЦ 8Д/ С НАЦИОНАЛЕН, УЪМАНЕН И ВЪЗПИТАТЕЛЕН ХАРАКТЕР.

## НАГРАДИ

### РАЗДЕЛ I

I НАГРАДА	- 5 000 ЛЕВА
II НАГРАДА	- 3 000 ЛЕВА
III НАГРАДА	- 2 000 ЛЕВА

### РАЗДЕЛ II

I НАГРАДА	- 3 000 ЛЕВА
II НАГРАДА	- 2 000 ЛЕВА
III НАГРАДА	- 1 000 ЛЕВА

### РАЗДЕЛ III

I НАГРАДА	- 1 000 ЛЕВА
II НАГРАДА	- 700 ЛЕВА
III НАГРАДА	- 500 ЛЕВА

КРАЕН СРОК НА КОНКУРСА - 30.IX.1986  
РАЗРАБОТКИТЕ ШЕ СЕ ПРИЕМАТ В РЕДАКЦИЯТА НА ДВЕТЕ ИЗДАНИЯ

СОФИЯ  
БУЛ. ТОЛЕУХИН 51  
ТЕЛ. В7-50-45 И 87-09-14



## ЗИМНИ ПРАЗНИЦИ

Ст. н. с. ПАВЕЛ АЗЪЛОВ

И тази януарска ваканция зимните празници на младите математици и информатици „Знаме на мира“ събраха стотици деца. Празниците са организирани от Съюза на математиките в България и са добре познати на учителите и учениците от нашата страна. Те вече имат свой облик и история. За пети пореден път с разменено домакинство в Русе и Варна деца от цялата страна се събират, за да изаявят способностите си по математика, информатика и математическа лингвистика. Още от първата година състезанието по информатика бе посрещнато с огромен интерес. По това време центровете, където в извънкласна форма се изучаваха елементи на информатиката, бяха все още твърде малко. Един поглед към състезанието от 1982 г. ни напомня, че тогава са участвували само 18 единадесетокласници от общо 9 окръга. През следващите години, особено след решението на Политбюро на ЦК и ВКП за създаване на комплексни условия за обучение и работа на младешката с електронноизчислителна техника, броят на участниците нарасна многократно. Труд-

## НА ИНФОРМАТИКАТА

## „ЗНАМЕ НА МИРА“ - РУСЕ '86

ни бяха първите състезания. Не само защото това бе началото, а и ай-вече поради неудобството да се провеждат състезания с използване на големи ЕИМ. Преломна в това отношение бе миналата година, когато домакните от Варна направиха първите стъпки в използването на микрокомпютърна техника в състезанието по информатика. Тази година всеки от 96-те участници в състезанието по информатика в Русе имаше и разположение отделен

ПК Пращец-82. Така се създадоха много добри условия за решаване на задачите от темите, но въпреки това има какво да си пожелаем за следващото състезание.

Една пълна окомплектовка на микрокомпютрите ще даде възможност за допълнителна изява и на онези ученици, които вече не се задоволяват само с използването на езика Бейсик.

Новост в сега проведеното състезание беше и това, че в него за първи път участваха и деца от VI и VII клас. За тях това бе сериозно изпитание, но и те като своите по-големи съотборници се представиха достойно. И в двете възрастови групи имаше ученици, които в конкурсното време предадоха решения, оценени от журито с максимален брой точки. А ето и резултатите:

**БЕЛЕЖКА НА РЕДАКЦИЯТА:** В рубриката „Софтуер“ предлагаме на нашите млади читатели състезателните задачи за двете възрастови групи заедно с разпечатки и анализ на програмните решения.

## КЛАСАЦИЯ

ПЪРВА ГРУПА  
(VI—VII КЛАС):

## Индивидуално

1. Десислав Ванчовски (В. Търново) — 20 т.
2. Камен Вакалов (Бургас) — 18 т.
3. Ивайло Врачански (Шлевен) — 17 т.

## Отборно (по окръзи)

1. В. Търново — 31 т.
2. Бургас — 29 т.
3. Варна — 28 т.

ВТОРА ГРУПА  
(VIII—IX КЛАС):

## Индивидуално

1. Росен Русев (Разград) — 20 т.  
Христо Христов (Ямбол) — 20 т.
2. Веселин Вараков (Бургас) — 19 т.  
Петър Стоянов (Варна) — 19 т.  
Цветан Статков (Враца) — 19 т.
3. Росен Ичев (Влагеоград) — 18 т.  
Красимир Момчев (В. Търново) — 18 т.  
Максим Зайков (Шлевен) — 18 т.

Георги Василев (Пловдив) — 18 т.  
Нико Николов (Разград) — 18 т.

## Отборно (по окръзи)

1. Разград — 38 т.
2. Бургас — 36 т.
3. Пловдив — 35 т.  
Ямбол — 35 т.

## КОМПЛЕКСНА:

1. Бургас — 65 т.
2. В. Търново — 65 т.
3. Варна — 57 т.  
3. Шлевен — 49 т.

Най-общо, една експертна система може да се разглежда като посредник между висококвалифицирани в дадена област специалисти (взаимодействащи с нея чрез подсистема за изграждане на база от знания) и потребителите (встъпващи в диалог със системата в режим на консултации за получаване на допълнителна информация и съвети). Тази дефиниция, макар и непълна, определя една съществена особеност на експертните системи: те се създават за решаването на проблеми с голяма степен на трудност, където познанията на отделния индивид са крайно недостатъчни. С други думи, експертни системи се използват най-често там, където съществува „тезиисна“ при натрупването на експертна информация за даден процес, обект или събитие. Обикновено тяхното разработване и внедряване е оправдано, ако тази информация се консумира от много потребители.

Общата структура на една експертна система включва три основни блока: интерфейсния програми за връзка с потребителя, интерпретатор на съставените от експерти правила и база от знания. От своя страна базата от знания е изградена от факти, правила и метаправила за съставяне на хипотези. Диалогът между експертната система и потребителите се осъществява под формата на въпроси, отговори и обяснения към отговорите, както и под формата на изграждане и актуализиране на базата от знания.

Общата структура е модулна: фактите и съставените правила за дадената област на приложение могат да бъдат отделени от процедурата за изводи или хипотези или от интерфейсната структура, докато другата част от системата — общата база знания, е модел на „част от света“, свързан със специфичен проблем, неговото състояние и история.

Изследванията са показали, че при разработването на експертни системи налицето на големи банки от събрани факти е много по-важно за мощността на системата, отколкото използваната техника на програмията. Основните изисквания при проектирането на дадена експертна система са следните:

**ИЗКУСТВЕН  
ИНТЕЛЕКТ**

## ЕКСПЕРТНИ СИСТЕМИ

К. Т. Н. ИЖ.  
СТАНИСЛАВ ДИМОВ

- Експертът трябва да състави базата от знания и да може да обясни методите, използвани за решаване на отделните задачи.

- Проектираната експертна система трябва да има точно определени граници на приложение.

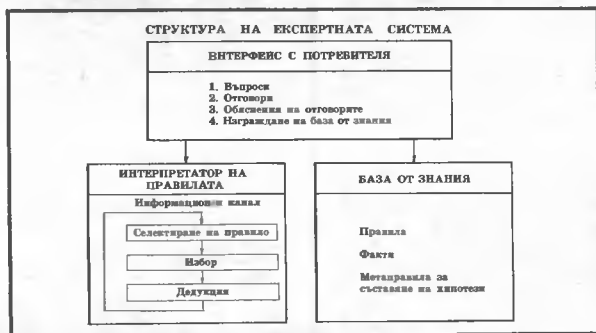
Един от най-важните проблеми при конструирането на експертните системи е организацията на потребителския интерфейс и на базата от знания.

Обикновено при повечето експертни системи чрез потребителския интерфейс се поддържат три режима на работа:

- консултация във вид на отговор на въпросите на потребителя;

- въвеждане на нови знания от страна на потребител-експерт;

- обучение на потребители с правилата на базата от знания. Диалогът се води в свободна форма и в реално време, често на език, близък до свободната човешка реч. Системата задава насочващи и уточняващи въпроси на потребителя и дава препоръки за вземане на решения.



- За определена област от човешкото знание е необходим най-малко един експерт, който да състави набора от правила на системата. Трябва да се отбележи, че изборът на признаци, по които се извършва класификацията, е най-трудоемката и отговорна работа.

- Първичните източници за събиране на експертната информация трябва да бъдат практиката, разсъжденията и знанията.

При организацията на базата от знания те в значителна степен са независими от обработващите модули. Определени еталони от данни избирателно задействуват отделни фрагменти от програмата. От гледна точка на конкретната задача голяма част от натрупаните знания могат да се окажат излишни. Данните могат



## МАТЕМАТИЧЕСКО СЪСТЕЗАНИЕ „ЗНАМЕ НА МИРА“ РУСЕ '86

# РЕШЕНИЯ НА ЗАДАЧИТЕ

### ТЕМА ЗА VI—VII КЛАС

Дадени са 2п цели положителни числа  $n$  ( $n \geq 5$ ). Измежду тях половината са четни, а останалите са нечетни. Да се състави програма, чрез която:

- а) да се въвеждат стойностите на  $n$  и на дадените 2п числа;
- б) да се разместят въведените 2п числа, така че нечетните позиции да бъдат заети от четните числа;
- в) да се променя наредбата на числата, получени от условие б), по такъв начин, че четните числа да остават на нечетни позиции, но вече да са разположени в нарастваща последователност, докато разположението на останалите числа да остане непроменено.

Резултатите да бъдат извеждани върху екрана на видеомонитора.

### АНАЛИЗ НА ЗАДАЧАТА

Формулировката на задачата е дадена в три отдели, но зависими условия. Първото условие изглежда твърде елементарно, но то изисква няколко проверки за коректност на входните данни:

- $n \geq 5$  (редове 10, 20);
- проверка за четност на едно число (ред 80);
- броят на въведените четни числа да е равен на броя на нечетните числа (редове 80—180).

Естествено разполагането на четните числа върху нечетните позиции може да се извършва непосредстве-

но след самото въвеждане на всяко число (редове 90—150), така както е представеня по-долу пример: фиг. 1.

Решаването на третото условие се свежда до нареджане на четните числа във възходящ ред. Особено в случая е, че числата, подлежащи на сортиране, не са едно по друго в масива В, а са разположени само в нечетните му позиции, т. е. те са в елементите В(1), В(3), В(5) и т. н.

Един от най-популярните (но не и от най-бързите) и най-лесен за програмане реализация метод за сортиране е т. нар. метод на „мехурчето“. При него всеки два съседни елемента В(1) и В(1+2) за  $i=1, 3, 5, \dots, 2N-1$  се сравняват и ако  $B(i) > B(i+2)$ , разменят местата си. След първото преглеждане на масива най-голямото четно число ще „застане“ в позицията  $(2N-1)$  на масива В. При второ подобно преглеждане следващото по големина четно число ще „застане“ в позицията  $(2N-3)$  и т. н.

Преглеждането на четните числа ще продължи, докато при поредното преглеждане се констатира, че не е извършено нито едно разменване на елементи, или пък след като бъдат извършени всичките  $(N-1)$  преглеждания, т. е. след като масивът е сортиран (и в двата случая). Сортирането на четните числа е описано в програмата с редове 200—280, а резултатите от решаването на отделните условия се извеждат в една и съща подпрограма (редове 320—380).

да се степенуват по своята достоверност и важност. Такиото тълкуване може да се променя в широки граници в зависимост от логическата им връзка с другите данни.

Друг важен проблем са програмните езици, на които се реализират експертните системи. Досега използваните езици от високо ниво са от трето или четвърто поколение. Те бяха използвани главно за решаване на изчислителни задачи. Решаването на проблемите на изкуствения интелект изложени създаването на нови непроцедурни езици — Лисп, Пролог и др.

Езикът Лисп е създаден в Масачузетския технически институт от Мвкарти през 1959 г. и се отличава със следните предимства:

— при него могат да се конструират дървовидни структури от данни с произволна дълбочина (брой йерархични нива);

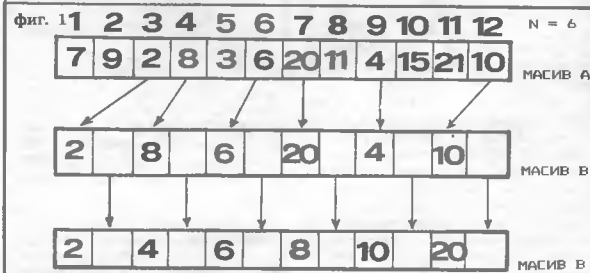
— определени функции могат да бъдат дефинирани рекурсивно;

— в Лисп няма формална разлика между данни и програми.

Всичко това прави Лисп незаменимо програмно средство при реализацията на различните експертни системи и решаването на проблемите на изкуствения интелект.

Много важна перспективна при разработването на експертните системи е възможността те да бъдат реализирани върху персонални компютри. Това се обуславя от нарастващата изчислителна мощност на ПК, техният обем памет и др. Досега подобни системи са реализирани на IBM-PC, IBM-PC-XP. Това са системите М1, Експерт-изи, R1 и др. Системата Експерт-изи например работи на персонален компютър IBM-PC със 128 Кбайта памет и най-малко едно минифлопидисково устройство.

В заключение може да се каже, че експертните системи като важна част на научното направление „Изкуствени интелект“ се намират едва в началото на своето създаване и развитие, но и тях се възлагат големи надежди.



## ТЕКСТ НА ПРОГРАМАТА:

```

10 INPUT "ВЪВЕДЕТЕ
N "; N
20 IF N < 5 THEN PRINT
"ГРЕШКА: N<5 ": GOTO
10
30 N2 = N + N
40 DIM A(N2), B(N2)
50 L1 = - 1: L2 = 0: IR
= 1
60 FOR I = 1 TO N2
70 PRINT "ВЪВЕДЕТЕ
A("; I; ")": INPUT A(I)
80 IF A(I) / 2 < >
INT ( A(I) / 2) THEN
GOTO 130
90 L1 = L1 + 2
100 IF L1 > N2 THEN
GOTO 180
110 B(L1) = A(I)
120 GOTO 160
130 L2 = L2 + 2
140 IF L2 > N2 THEN
GOTO 180
150 B(L2) = A(I)
160 NEXT I
170 IR = 0
180 IF IR = 1 THEN
PRINT "ГРЕШКА: ВЪВЕДЕТЕ
ЧИСЛАТА ОТНОВО": GOTO
50
190 GOSUB 320
200 M = 2 - 1
210 K = 0
220 I = 1
230 IF B(I) > B(I +
2) THEN X = B(I): B(I)
= B(I + 2): B(I + 2)
= X: K = 1
240 I = I + 2
250 IF I < M THEN
GOTO 230
260 IF K = 0 THEN
GOTO 290
270 M = M - 2
280 GOTO 210
290 GOSUB 320
300 END
310 REM ПОДПРОГРАМА
ЗА ИЗВЕЖДАНЕ НА ПОЛУЧЕНИТЕ
РЕЗУЛТАТИ
320 PRINT SPC( 10); "МАСИВ
A": PRINT SPC( 10); "МАСИВ
B"
330 FOR I = 1 TO N2
340 PRINT SPC( B ); I;
TAB( 13); A(I); TAB(
30); B(I)
350 NEXT I
360 RETURN

```

### ТЕМА ЗА VIII—IX КЛАС

Нека A и B са изпъкнали многоъгълници в една равнина съответно с върхове  $A_1, A_2, \dots, A_m$

и  $B_1, B_2, \dots, B_n$ , чиито координати са цели неотрицателни числа ( $m \geq 5, n \geq 5$ ). Разположението на многоъгълниците е такова, че върховете  $A_2, A_3, \dots, A_k$  са вътрешни точки за многоъгълника B, а върховете  $B_2, B_3, \dots, B_l$  са вътрешни точки за многоъгълника A ( $1 < k < m, 1 < l < n$ ), като точка  $A_l$  съвпада с точка  $B_{l+1}$ , а точка  $B_l$  съвпада с точка  $A_{k+1}$ . Последователността на върховете и за двата многоъгълника е дадена в посока обратна на часовниковата стрелка. На представената по-долу фигура е посочен случай за  $m=9, n=8, k=4, l=3$ .



Съставете програмата, чрез която:

а) да се въвеждат координатите на върховете на многоъгълниците A и B, след което многоъгълниците да се изобразяват на екрана;

б) да се пресмятат лицата на многоъгълниците A и B и на тяхното обединение, т. е. на многоъгълника с върхове  $B_{l+1}, B_{l+2}, \dots, B_n, A_{k+1}, A_{k+2}, \dots, A_m$ ;

в) да се изобразяват всички диагонали в многоъгълника  $A_1 A_2 \dots A_k B_1 B_2 \dots B_l$ .

Обосновайте в писмена форма използвания от вас алгоритъм.

### АНАЛИЗ НА ЗАДАЧАТА

След стойностите на променливите M и N ( $M \geq 5, N \geq 5$ ) се въвеждат координатите на върховете на многоъгълниците. Удобно е тези координати да се съхраняват в по два масива AX(M), AY(M) и BX(N), BY(N). Върху входните данни е изложително да се извърши контрол за коректност, включващ следните проверки:

- $M \geq 5, N \geq 5, K < M, L < N$  (редове 20, 30);
- $0 \leq AX(I) \leq 279, 0 \leq AY(I) \leq 159, I=1, 2, \dots, M$  (редове 70—110);
- $0 \leq BX(I) \leq 279, 0 \leq BY(I) \leq 159, I=1, 2, \dots, N$  (редове 120—160).

За изобразяването на самите многоъгълници се използват графичните възможности на Бейсик, като за целта е дефинирана една подпрограма (редове 560—

590). Забележете, че тя използва работните масиви X и Y, съдържащи координатите на върховете на многоъгълника, който в дадения момент се начертава, като  $X(NR+1) = X(1)$  и  $Y(NR+1) = Y(1)$ . Същите тези масиви се използват и от подпрограмата за пресмятане на лице на многоъгълник. Използувана е формулата:

$$S = \frac{1}{2} \left| \sum_{I=1}^{NR} (Y(I+1) + Y(I)) \cdot (X(I+1) - X(I)) \right|,$$

който представлява сума от ориентирани лица на NR трапеца, чиито основи са успоредни на оста OY. Трябва да се отбележи, че този метод за пресмятане на лице на многоъгълник не е единствен, но той има няколко съществени предимства:

- важи както за изпъкнали, така и за неизпъкнали многоъгълници;
- лесен е за програмна реализация;
- по-точен е спрямо методите, използващи и действие коренуване.

Ако означим с T(X, Y, P) подпрограмата (редове), пресмятаща лицето на r-ъгълника, чиито координати се съдържат в масивите X и Y, тогава решаването на второто условие на задачата се свежда до извършване на следните операции:

$$\begin{aligned}
 S_1 &= T(AX, AY, M) \\
 S_2 &= T(BX, BY, N) \\
 S_3 &= T(X, Y, K+L),
 \end{aligned}$$

където X и Y съдържат координатите на точките  $A_1, A_2, \dots, A_k, B_1, B_2, \dots, B_l$ .

$$S_4 = S_1 + S_2 - S_3,$$

където  $B_4$  ще бъде лицето на обединението на многоъгълниците A и B.

Съществено при решаването на третото условие е, че от всеки връх излизат по (p-3) диагонала, но броят на неповтарящите се диагонали за отделните върхове на r-ъгълника ще бъде:

- p-3 за 1-и връх;
- p-3 за 2-и връх;
- p-4 за 3-и връх;
- .....

- 2 за (p-3)-и връх;
- 1 за (p-2)-и връх.

Координатите на върховете на многоъгълника, който е сечение на двата многоъгълника A и B, отново се съхраняват в работните масиви X и Y (редове 260—300 и 340—380), а диагоналите се начертават чрез операторите с измера 420—470.



```

10 HOME
20 INPUT "ВВЕДЕТЕ: M, N, K, L
  +M, N, K, L
30 IF M < 5 OR K >
  M OR L > N THEN PRINT
  "ГРЕШКА!": GOTO 20
40 MN = N + 1
50 IF N < M THEN MN
  = M + 1
60 DIM AX(M), AY(M), BX(N), BY(N), X
  (MN), Y(MN)
70 FOR I = 1 TO M
80 PRINT "ВВЕДЕТЕ
  AX( ; I ); AY( ; I );":
  INPUT AX(I), AY(I)
90 IF AX(I) < 0 OR
  AY(I) > 159 THEN PRINT
  "ГРЕШКА": GOTO 80
100 IF AX(I) > 279
  OR AY(I) > 159 THEN
  PRINT "ГРЕШКА": GOTO
  80
110 NEXT I
120 FOR I = 1 TO N
130 PRINT "ВВЕДЕТЕ
  BX( ; I ); BY( ; I )":
  INPUT BX(I), BY(I)
140 IF BX(I) < 0 OR
  BY(I) < 0 THEN PRINT
  "ГРЕШКА": GOTO 130
150 IF BX(I) > 279
  OR BY(I) > 159 THEN
  PRINT "ГРЕШКА": GOTO
  130
160 NEXT I
170 HGR : HCOLOR = 3
180 FOR I = 1 TO M
190 X(I) = AX(I):Y(I)
  = AY(I)
200 NEXT I
210 X(M + 1) = AX(I):Y(M
  + 1) = AY(I)
220 NR = M
230 GOSUB 500
240 GOSUB 560
250 S1 = S
260 FOR I = 1 TO N
270 X(I) = BX(I):Y(I)
  = BY(I)
280 NEXT I
290 X(N + 1) = BX(I):Y(N
  + 1) = BY(I)
300 NR = N
310 GOSUB 500
320 GOSUB 560
330 S2 = S
340 FOR I = 1 TO K
350 X(L + I) = AX(I):Y(L
  + I) = AY(I)
360 NEXT I
370 NR = K + L
380 X(NR + 1) = BX(I):Y(NR
  + 1) = BY(I)
390 GOSUB 500
400 S3 = S
410 S = S1 + S2 - S3
420 FOR I = 1 TO K
  + L - 2
430 NR = K + L - 3
440 IF I > 1 THEN NR
  = K + L - 1 - I
450 FOR J = 1 TO NR
460 HPLOT X(I), Y(I)
  TO X(I + J + 1), Y(I
  + J + 1)
470 NEXT J, I
480 PRINT "SA=";S1;
  SPC( 5); "SB=";S2; SPC(
  5); "SB=";S
490 END
500 S = 0
510 FOR I = 1 TO NR
520 S = S + (Y(I + 1)
  - Y(I)) * (X(I + 1)
  - X(I))
530 NEXT
  540 S = 0.5 * ABS (S)
550 RETURN
560 FOR I = 1 TO NR
570 HPLOT X(I), Y(I)
  TO X(I + 1), Y(I + 1)
580 NEXT I
590 RETURN

```

## РЕЧНИК

Допреди няколко години у нас работеха 3—4 хиляди професионални информатици, които заедно с повечето си колеги — специалисти по изчислителна техника, и известни брой непрофесионални потребители на информатиката и любители затваряха кръга на словотворците в тази област. Напоследък с нахлуването на компютрите положението се промени. Към днешния ден десетки, а скоро — и стотици хиляди хора ще бъдат по един или друг начин свързани с компютризацията на обществото. Тогава проблемите на терминологията, правописа и правоговора в информатиката и българския ще станат и техни проблеми.

Ето защо българските информатици трябва да бъдат готови да дадат своя принос за уеднаквяването и стандартизирането на езика на информатиката в страната, който сега се отличава със завидна неустойчивост и голяма доза непрофесионализъм.

С настоящите бележки се цел да бъдат напомнени на всички автори и читатели на списанието някои положения от най-новия „Правописен речник на съвре-

# ЕЗИКОВИ БЕЛЕЖКИ

Доч. ДИМИТЪР П. ШИШКОВ

## ТЕРМИНОЛОГИЯ

## ПРАВОПИС

## ПРАВОГОВОР

## В

## ИНФОРМАТИКАТА

жения български книжовен език", издание на Института за български език (ИБЕ) при БАН от 1983 г., които пряко засягат текстовете по информатиката. Той ще бъде цитиран по-нататък като Речника. Основните езикови положения и решения, залегнали в него, според мен трябва да бъдат приемани от всички. Други бележки ще бъдат посветени на някои предложения, които могат да бъдат подложени на пробна експлоатация (и не само от Компютър за вас), след което, ако целесъобразността им бъде пот-

върдена в практиката, могат да бъдат предложени за разглеждане от ИБЕ при БАН. Авторът на тези редове е професионален математик и информатик, но не и лингвист. Ето защо всички забележки и предложения ще приемем с радост.

Предлагам основните принципи в тези бележки да бъдат:

**Принцип 1.** В текстовете по информатика на български не бива да има по възможност думи на латиница, освен служебните (ключови) думи на английски от различните езици и програмни

системи в информатиката. На български трябва да се пишат изваданията на фирмите, компютрите, езиците, програмните системи и пакети, а също и възприетите като думи *звукони съкращения* (ром, рам, епром, коп и др.). Едва ли някой си спомня, че *лазер* е пълно звуково съкращение, подобно на чисто иашенските *вуз*, *тец* и много други. *Възможни изключения*: някои съкращения на латиница, които по различни причини (например иеоднозначна обратна транслитерация) не бива да се транслитерират. Така



фирмата DEC не трябва да се пише ДЕК или Дек, професионалният компютър PC на фирмата ИБМ също не бива да се по-българява с ПС, ПК или ПЦ и др.

**Правило 2.** Ако терминът или извадението са на чужд език — да се запазва правописът на оригинала. Ако са на български — да се пишат съгласно някакво общо правило, а не в томои съответствие с оригинала. Това особено важи за различните съкращения, които на латиница обикновено се пишат с главни букви.

**1. Названията на компютрите**  
Съгласно изданието на ИВЕ: *Л. Крумова и М. Чорозеева. Съкращаването и съкращенията в българския език. С., БАН, 1982*, в което специално се разглеждат и изваденията на машините (с. 78 и с. 100), трябва да се премине към правопис на названията на компютрите, устройствата *към тях, микропроцесорите и калкулаторите без кавички* (вж. също и §80, т. 4 на Речника), само първата буква главна за думи и всички звукови съкращения, които се четат и се възприемат вече като обикновени думи (с главни букви остават само бук-

вените съкращения, които се произнасят буква по буква), а *числото отзад* (ако има такова) се присъединява с малко тире (съединително тире, дефис) например Минск-22, Бесм-6, Стрела, Зит-151 (въпреки, че заводът е ЗИТ, аналогично ФИАТ — Фиат), Правец-82, Имко, СМ-4, Изот-1002С, ЕС-2706, ИВМ-360 и др.

Да отбележим, че досега компютрите и устройствата от ЕС (Единната система от ЕИМ) са писани винаги без тире, тези от СМ (СМ-4) — винаги с тире, а изделията на ИЗОТ — кога с тире,

кога — без, почти винаги — с ИЗОТ, рядко с Изот.

**1.1.** Наклонена черта не е необходимо да се използва в извадения на машини, респ. компютри. Предлагам съгласно българските машиностроителни стандарти да се пише ИВМ-370-158 и ПЦП-11-34 в.м. ИБМ-370/158 и ПДП-11/34, защото как да постъпим, ако се иналожат две или повече наклонени черти?

**1.2.** Предлагам всичко, което следва марката (фирмата) след малкото тире и е свързано със сериата или номера в нея, акл. обяснителните буквени съкращения, които не трябва да се транслитерират, да се пише *слато*, ако числата се разедияват с букви, а буквените части — с числа, например Изот-0220М1, Касио-FX602P, K1810BM86 (тук липсва марка или фирма), Електроника-МК54 и др. Две или повече числа (например серия — номер) се разделят с малко тире (вж. 1.1). Поради това, че в БДС се допуска и *разделител точка*, само за българските компютри се допуска например Изот-0220.М1. *Изключения*: Електроника-В3-34, а не Електроника-Б334, пожеже

БЗ са букви и буквата З не се отделя от цифрата 3.

**1.3.** Съгласно §114—116 на Речника основно правило при транскрипцията на чуждите думи е да се предава по възможност най-точно техният звуков състав със средствата на приемашния език — в случая — българския. Също така са важни препоръките на ООН по стандартизацията на правописа и правописа на чуждите имена и названия (вж. статията *„Традиция или точност?“ на Кристина Попова в в.р. 6 на АВВ от 11.02.1986 година*). Въз основа на това предлагам названията на чуждите фирми да се транскрибират правилно, а не традиционно, например Хюлит-Пакард (а не Хюлет...), Къитрол дейтъ (а не Контрол дейта), Сайбър (а не Кибер) и др. Съгласно § 32, т. 1 и примерите в § 115 на Речника втората и следващата дума от названията на чуждите фирми трябва да се пишват на български с малка (редовна) първа буква например Диджитъл икуипмънт корпърейшън като обикновено думата корпърейшън се изпуска.

**2. Названията на езиците за програмиране**

Досега българските лингвисти не са се занимавали с този въпрос. Има три начина за изписване на названията на тези езици на български и руски: а) само с главни букви; б) първа буква — главна, остиялите — малки и в) само с малки букви (засега единствено от съветското издателство „Наука“, у нас — никога досега освен автокод, асемблер и макроасемблер, и то не винаги).

По пълна аналогия с изваденията на компютрите (т. 1) предлагам да се възприеме вторият начин — без кавички, само първата буква да е главна за всички названия на езици за програмиране (програми езици, вкл. и звуковни съкращения), които се четат и възприемат като думи. Нещо повече — създателите на тези езици винаги са се стремил да намерят (ако може) такива съкращения, които да се възприемат именно като *известни* (Бейсик, Сирус) или *звучни* (Алгол, Фокал) думи. Само тези названия, които са останали буквени съкращения и се възприемат и изговарят (скандират) като таки-



ва, трябва да продължават да се пишат изцяло с главни букви. Названието на езика и числото след него (ако има такова) предлагам да се обединяват с малко тире, дори цифрите да са римски. Примери: Алгол-60, Кобол, Фортран-IV, Фортран-77, АПЛ, С (да се чете си, неправилно — це; тъй като е необходимо „С“ да се печата с друг шрифт, за да се разбира, че става дума за този език освен когато изрично се пише „езика С“, предлагам да пишеш Си, както е на руски), ПЛ-1 (чете се пи-ел-едно, не бива ПЛ/1, но може PL/1), РПГ и др.

**Изключения.** Езиците за програмиране от високо ниво са значително машиннонезависими. Поради различни причини те имат *диалекти* (за една група езици се използвав терминът алгоподобни езици, но те не са диалекти на Алгол). Асемблерите и другите машиннозависими езици са *уникални* и практически нямат диалекти. Съгласно т. 2 при пълно изписване на извадието на такъв език, включващо и названието на съответния компютър, предлагам да се започва с главна буква например Асемблер за ЕС или Асемблер за СМ-4. Когато обаче компютърът се подразбира или става дума за такъв тип езици, предлагам традицията да се запази и да се пишат с малка първа буква *автомат, асемблер и макроасемблер*.

Съветското издателство „Наука“ е възприело названията на всички езици да се пишат с малки букви: руска, английска, есперанто, симула-67, алгол-68 и пр. Тук се предлага първата буква от наименованието на програмните езици да бъде главна, за да отличаваме тези езици от говоримите (есперанто е изкуствен, но говорим).

2.1. Не всички извадения на езици за програмиране образуват на практика прилагателни. Езиците, чиито названия са буквени (а не звукови) съкращения, обикновено нямат прилагателни, например РПГ, Си и др. Наложили се **изключения:** прилагателни, построени на базата на английското произношение на названието на езика, например *пиексия* или *айпиексия* програма.

Други езици с извадения думи не са образували прилагателни — Ада, Симула-67. При трети отпада и се подразбира числото в названието — алголска, фортранска програма (вж. Приложение 2).

2.2. Винаги е възможно извадението на езика да се съчлени чрез малко тире с думата *про-*

*грама* — Бейсик-програма, Фал-програма. Тире не се използва само когато названието е буквено съкращение с повече от една буква (§50, съществителни имена, т. 5 на Речника) — РПГ програма, но С-програма и, разбира се, Си-програма.

## ПРИЛОЖЕНИЕ 1

### В ПРОБНА ЕКСПЛОАТАЦИЯ

### НАЗВАНИЯ НА КОМПЮТРИ, УСТРОЙСТВА, МИКРОПРОЦЕСОРИ И КАЛКУЛАТОРИ

● С точка са отбелязани произвежданите у нас.

● С малко х се отбелязват цифрови позиции, които се запълват с номера вътре в серията.

● Следващите след първата дума от названията на чуждестранните фирми се пишат с малка буква, а думата *корпорейшън* обикновено се изпуска.

ВК-1300 (чете се Бекъ)  
Въроуз-5500 или В-5500 и др. (а не Въроуз)  
Бесм-2, -4, -6 и ВЭСМ (два бѣсѣма)  
Вакс-780, -11750 и др. (два вакса)

Джиджитъл икуипмъйт (корпорејшън) или DEC (чете се ДЕК)  
Елка-6521, -22, -25, -45, -58, -90, 99, -101, -130 и др.  
ЕС-1020, -1022, -1035, -1036, както и всички устройства на ЕС като ЕС-2706 и др. (ѣсовсѣа машинѣа)  
Зит-151 (два зѣта)  
Изот-310, -0220, -0220M1, -0220M1, -1002С, -1030С, -1036С и др. (два изотѣа)  
ИВМ-360, -370, -303х, -403х, -403хх, ИВМ-370-158 (но ИВМ/370, ИВМ 370/158) и др.  
ИВМ-РС-ХТ

Имко и Имко-2  
Интел, Интел-8008, -8080, -8086, -286 и др. (два интѣла)  
Интел-432 или iAPX 432 и др.  
Искра-526 и др.  
Каско-FX602P  
Крей (два креѣа)  
КР580ИКС80А, К1810ВМ86 (без специална марка)  
М-20, -220, -222 (две М-20)  
Мик-16 (два мѣка)  
Минск-2, -22, -23, -32 (два мѣнсѣа)  
Моторола, Моторола-6800, -6809, -68000 (но МС68000; две мотороли)

Найри-2, -3, -4  
NCR  
Нѣшънъл сѣмѣкѣндѣкѣтр-16000, НС-16000, НС-32С016 или NS16000, NS 32С016  
ПДП-11-34 (а не ПДП-11/34, но PDP-11/34) и др. (две пидѣпѣта)  
Роботрон-1720 (два роботрона)  
Правец, Правец-82, Правец-8Д, Правец-16 и др. (два правѣца)  
Сайбър-205 (но не Кибѣр)  
Кантрол дѣйтѣ (корпорејшън) или CDC 6600, CDC 7600 и др. (но не Кантрол дѣѣта)  
СМ-3, -1800  
СМ-4  
СМ-500  
СМ-600, СМ-601 и др. (нямат нищо общо със СМ-4 — Система малѣах (машини)  
Спѣри (рејд корпорејшън)  
Стрѣла (две стрѣли)  
Тексѣс ѣнструмѣнтс, ТИ-58-59 или ТИ, ТИ-59, ТИ-58/59  
Урал-12, -14, -16 (два урѣла)  
Хѣйуѣл (а не Хѣйуѣл)  
Хюлит-Пакард или HP (а не Хюлет-...)  
Електроника-ВЗ-34, Електроника-МК54, Електроника-МК56, Електроника-60, -70, Електроника-НЦ80 и др.  
(но Електроника ВЗ-34, Електроника МК-54 и т. н.)  
Юнѣвак сѣйѣнтѣфик  
Wang (или необѣчѣйното Ван — така се произнася на китайски, докато на английски е Уонг)

(Списъкът не претендира за пълнота)

## ПРИЛОЖЕНИЕ 2

### В ПРОБНА ЕКСПЛОАТАЦИЯ

### НАЗВАНИЯ НА ЕЗИЦИ ЗА ПРОГРАМИРАНЕ

● Названията с курсива не са съкращения.

● Езиците с точка отпред са български.

● Винаги е възможно да се казва програма на едн-кой си език или да се пише название на език-програма.

Автокод за конкретен компютър (ако компютърът се подразбира — автокод; автокодове; автокодова програма; *компиляторът* от този език — също автокод

(книги по информатика, появили се в продажба от 1 до 28. 02. 1986 г.)

## БЪЛГАРСКИ КНИГИ

В. Боровския, Л. Давковския, К. Боянов и А. Егоров. СПРАВОЧНИК ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА. ЦИФРОВИ ЕЛЕКТРОННО-ИЗЧИСЛИТЕЛНИ МАШИНИ. С., Техника, 1985, 475 с.  
П. Сираков и О. Вълчава. КЛЮЧ ЗА КОМПЮТЪР. Въведение в Бейсик. С., Народна младеж, 1985, 80 с. (Поредница „Авагард“).

## КНИГИ, ПРЕВЕДЕНИ НА БЪЛГАРСКИ

К. Е. Огдин. ОРГАНИЗАЦИЯ И ПРОГРАМИРАНЕ НА МИКРОКОМПЮТЪРНИ СИСТЕМИ. С., Техника, 1985, 268.

## СЪБЕТСКИ КНИГИ

В. В. Александров и Н. Д. Горский. ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ИЗОБРАЖЕНИИ. РЕКУРСИВНИЙ ПОДХОД. Л., Наука, ЛО, 1985, 190 с.

А. С. Вомко. МИКРО-ЭВМ В СИСТЕМАХ УПРАВЛЕНИЯ ПРОИЗВОДСТВОМ. К., Техника, 1984, 128 с.

Н. Т. Демидович. ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И ПРОГРАМИРОВАНИЕ. Минск, Университетское, 1985, 272 с.

Г. А. Звенигородский. ПЕРВЫЕ УРОКИ ПРОГРАМИРОВАНИЯ. М., Наука, 1985, 208 с. (Библиотека „Квант“).

Б. В. Игнатенко и В. И. Павловский. ЯЗЫКИ МАНИПУЛИРОВАНИЯ ДАННЫМИ. МОДЕЛИ И СРЕДСТВА. К., КГУ, Вища школа, 1984, 126 с.

З. Г. Керимов и С. А. Вагиров. АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ КОНСТРУКЦИИ. М., Машиностроение, 1985, 221 с.

С. Д. Погорелый и Т. Ф. Слободянок. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ. СПРАВОЧНИК. К., Техника, 1985, 240 с.

СИСТЕМИ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ. М., Наука, 1985, 376 с.

А. И. Салтыков и Г. Л. Семашко. ПРОГРАМИРОВАНИЕ ДЛЯ ВСЕХ. М., Наука, 1986, 176 с.  
В. Г. Тоценко. АЛГОРИТМИ ТЕХНИЧЕСКОГО ДИАГНОСТИРОВАНИЯ ДИСКРЕТНЫХ УСТ-

Предлаганите бележки имат стойност преди всичко с това, че целят да регламентират начина на писане и изговаряне на част от терминологията, употребявана в областта на информатиката — названията на компютрите и на езиките за програмиране, като се съобразяват с действителната система от норми и правила в съвременния български иривонис и правоговор изобщо. По такъв начин новото, което се въвежда и утвърждава, няма да затруднява при усвояване, защото ще се свързва с онова, което е вече познато.

Препоръките имат за цел и да подпомагат някои граматически операции — избора на бройни форми и на форми за множествено число на названията и словообразователни процеси — образуване на прилагателни имена от съответните наименования. В този случай правилно от гледна точка на законите за езиковото развитие се работи на принципа на аналогията, тъй че разглеждайки терминологични единия съсем естествено биват включени в българския език като негова органична част, а и са подчинени на специалните изисквания на българската терминология.

Общото основно достойнство на препоръките е, че те търсят да постигнат системност и последователност при формулирането на правилата на отделните разнища (с ограничаване на изключенията), което значително ще облекчи тяхното възприемане и прилагане.

Възражение биди единствено идеята за т. нар. случаи в проба на осигуряване, които трябва според автора да бъдат разглеждани от Института за български език при БАН „след като целесъобразността им биде потвърдена в практиката“. Както показва и този първ опит, необходимо е предварително мнение на езиковедите при подбора на възможните решения и при тяхното иредставяне. Разбира се, това не изключва наместата на специалистите оценка и при оценката на допълнителните иредложения, които ще се ищиряват, за да се достигне до избора на най-подходящия вариант и той да се утвърди като единствен.

К. ф. н. ВЛАДКО  
МУРДАРОВ

Ада (Ада-програма)  
Алгамс (алгамсова програма)  
Алфърд  
Алгол-58, -60, -68, -W (алголска програма)  
АПЛ (чете се Ейпийл; ейпийелска програма)  
Асемблер за конкретен компютър; еайка Асемблер (ако компютърът се подразбира — асемблер; асемблери; асемблерски езици; не бива да се предпочита асемблеров, още повече — асемблери; асемблерска програма; компилаторът от този език — също асемблер)

Бал (Вал-програма)  
Бейсик (а не Бейзик; Бейсик-програма)

Джовинал  
Днамс (днамсова програма)  
Евклид (на английски се произнася Юклид)

Клу  
Кобол (коболска програма)  
Комит

Лисп (Лиса-програма)  
Лого

Макроасемблер за конкретен компютър (ако компютърът се подразбира — макроасемблер; макроасемблери; не бива да се предпочита макроасемблеров, толкова повече — макроасемблерен; макроасемблерска програма; компилаторът от този език — също макроасемблер)

Макро-11  
Модула, Модула-2

Паскал (паскалска програма)  
ПЛ-1 (а не ПЛ/1, но PL/1; чете се Пи-ел-едино; пийелска програма)  
ПЛ-М (а не ПЛ/М, но PL/M)  
ПЛ-М86  
Пролог

Рефал (рефалска програма)  
РПГ (може да се прочете Арпидж; РПГ програма — без тире)

С (предложение: Си; С-програма или Си-програма)  
Симула-67  
Снобол-4 (сноболска програма)

Фал (Фал-програма)  
Фокал (фокалска програма)  
Форт (Форт-програма)  
Фортран, Фортран-II, Фортран-IV, Фортран-77, Фортран-8х (фортранска програма)

(Списъкът не претегдира за излюновата)





Инж. ВЕСЕЛИН ВЪЛКОВ

# СИСТЕМИЗЪТ

Институтът за приложни системи „Системизът“ е създаден през 1978 г. към инженеринговото предприятие „Изоткомплект“, а от 1984 г. се обособява като специализирано звено в системата на ДСО „Изот“. Основните му дейности са производство, адаптиране, разпространяване и внедряване на приложни системи с електронноизчислителна техника. Главната задача на института е да бъде свързващо звено между разработчика и производителя (в рамките на ДСО „Изот“), от една страна, и потребителите на приложните системи, от друга.

В своята дейност ИПС „Системизът“ сътрудничи тесно с Централния институт за изчислителна техника и заводите производители в системата на ДСО „Изот“, като основно разработва програмни продукти за техническите средства, произвеждани в системата на обединението. По-точно казано, програмните продукти са ориентирани към следните електронноизчислителни системи:

1. Миникомпютри от типа Изот-1016С (СМ-4) с операционна система Джамс-В.
2. Персонални 8-битови компютри Изот-1031С под управлението на операционна система ДОС Умко, съвместима със СР/М.
3. Персонални 16-битови компютри Изот-1036С и Изот-1037С под управлението на операционна система ДОС ПК.
4. Локални мрежи от мини- и персонални компютри от типа Мрежа СМ-4 и Изотранг.

Главните направления на работа на института са:

- Разработка и внедряване на приложно програмно осигуряване за автоматизация на проектирането, на автоматизирани работни места на база професионални компютри, на приложни системи за автоматизация на учрежденската дейност и автоматизация на управлението на машиностроителните предприятия.
- Разработка на технологии за създаване на програмни продукти и внедряване на съвременни индустриални методи за програмиране и проектиране.

- Разпространяване и системно съпровождање на програмните продукти, разработени в подразделенията на ДСО „Изот“.

Към ИПС „Системизът“ е обособена като специализирано звено Фирмената програмна библиотека (ФПВ) на ДСО „Изот“. Тя осъществява връзката между производителите на програмно осигуряване и потребителите чрез следните дейности:

- Реклама на наличните и очакваните да постъпят във фонда на библиотеката програмни продукти.
- Производство на текстова документация и машинни носители на програмните продукти.
- Разпространяване на програмните продукти, производство на ДСО „Изот“.
- Оказване на висококвалифицирана помощ при внедряване на предоставените програмни продукти.
- Системно съпровождане на приложното програмно осигуряване в гаранционен и извънгаранционен срок.
- Даване под наем на електронноизчислителна техника, програмни продукти и машинно време; комплексни услуги за електронноизчислителна техника и кадри.

Всяка есен в рамките на Пловдивския панаир ИПС „Системизът“ организира и провежда международен симпозиум „Технически и програмни средства за ЕС ЕИМ и СМ ЕИМ и тяхното приложение в народното стопанство“.

За делови контакти се обръщайте на адрес:

1113 София  
бул. Ленин, 7-и км.  
номератори: 70-10-12  
70-10-13  
70-10-14

ФПБ: 74-11-72

РОИСТВ. М., Радно и связь, 1985, 239 с.  
ЭВМ В ПРОЕКТИРОВАНИИ И ПРОИЗВОДСТВЕ. Л., Машиностроение, ЛО, 1985, вып. 2, 303 с.  
ЯЗЫКИ ПРОГРАММИРОВАНИЯ. М., ВЦ АН СССР, Наука, 1985, 81 с. (Алгоритмы и алгоритмические языки).

## КНИГИ, ПРЕВЕДЕНИ НА РУСКИ

Дж. Мартин. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ И РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ДАННЫХ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, МЕТОДЫ И АРХИТЕКТУРА. Вып. 1. М., Финансы и статистика, 1985, 256 с.

СИСТЕМЫ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ. М., Мир, 1985, 415 с.

## КНИГИ, СВЪРЗАНИ С ИНФОРМАТИКАТА

РОБОТИЗАЦИЯ СВОРОЧНЫХ ПРОЦЕССОВ. М., Наука, 1985, 256 с.

Предлагаме ви кратки описания на някои програмни продукти, с които разполага фондът на Фирмената програмна библиотека на ДСО „Изот“. Подбрайте програмни продукти са за универсален микрокомпютър ИЗОТ-1031С под управлението на операционна система ДОС УМКО.

## ПАКЕТ ПРИЛОЖНИ ПРОГРАМИ „СТАТИСТИК-1031“

Предоставя възможност за решаване на различни статистически задачи: пресмятане на вероятности, регресии и корелационен анализ, аналитични и дискретни разпределения, анализ на Фурье, проверка на хипотези и непараметрична статистика. Статистик-1031 е диалогова система, с която могат да работят и потребители без специални познания по изчислителна техника. Изборът на конкретна програма от менюта се осъществява посредством специализирания менюта. Той може да намери приложението в лаборатория, учебни заведения, научни институти — навсякъде, където е необходима статистическа обработка на данни. Обработваните данни могат да се извеждат както чрез клавиатурата, така и чрез предварително създадени файлове. Резултатите могат да се извеждат под форма на диаграми и таблици.

## СИСТЕМА ЗА ОБРАБОТКА НА ИКОНОМИЧЕСКА ИНФОРМАЦИЯ

Системата представява комплекс програмни модули, предназначени за механизирани и автоматизирани на явков планово-икономически и управленски задачи в рамките на едно стопанско обединение и неговите подразделения. В състава на системата влизат седем пакета приложни програми с автономно действие, които извършват следните дейности: предварителни разчети при изготвянето на плана на индустриално-аграрната организация; планиране и отчитане на материално-техническата и развойната дейност; оперативно управление на производството; отчитане на реализацията на готови продукти; планиране и отчитане на ефективността от измаса на готови продукти; пресмятане на сметката за образуване и разпределение на общия доход; изготвяне на валутна калкулация на дадено изделие.

## ПАКЕТ ПРИЛОЖНИ ПРОГРАМИ ЗА ДЕЛОВА ГРАФИКА

С негова помощ могат да бъдат визуализирани всякакъв вид данни, отнасящи се до дейността на промишлените предприятия. Това се осъществява чрез графично представяне на данните чрез линии, кръгове, правоъгълни и други диаграми. Предвидена е възможност за изменение на мащаба и времеви диапазон на диаграмата, за извеждане на обяснителни текстове, за избор на шрифтове за надписите в прощолка на диаграмите. Управлението се осъществява с помощта на менюта. Изборът на командите и изпълнението на задачите е лесно и удобно за потребителя.

## ПАКЕТ ПРИЛОЖНИ ПРОГРАМИ ЗА ОТЧИТАНЕ НА ПАРИЧНИ, БАНКОВИ И РАЗЧЕТНИ ОПЕРАЦИИ

Кръгът на потенциалните потребители на този пакет е твърде широк, тъй като не зависи от отрасловата ви принадлежност. Пакетът дава възможност за счетоводно отчитане както на паричните средства и техните източници, така и на стопанско-правните отношения на потребителя на пакета с други физически и юридически лица. Изпълняваните функции са: създаване и поддържане на база данни; хронологично, систематично, аналитично и синтетично отчитане на стопанските операции; извеждане на сираци за състоянието и движението на счетоводно отчитаните средства.

## ИНФОРМАЦИОННА СИСТЕМА ЗА КОНТРОЛ ПО ИЗПЪЛНЕНИЕ НА РЕШЕНИЯТА

Системата автоматизира възлагането, отчитането и контрола по изпълнението на задачите в централните управления на министерства, комитети, обществено-политическа организация, обединения, предприятия и учреждения. Основните ѝ функции са:

- извеждане на данни за задачите във възлагането и;
- корекции и отчет на вече въведени задачи;
- извеждане на разнообразни справки.

## СИСТЕМА ЗА АВТОМАТИЗИРАН КОНТРОЛ НА РАБОТАТА НА БЕНИНОСТАНЦИЯ

Системата, наречена Бениност-1031С, автоматизира процесите на зареждане на производните средства в бензиностанцията и отговорността на горивото и смазочни материали. Нейната структура позволява да се обслужват до 16 бензинови колонки в няколко режима. Системата осъществява следните операции:

- предварително задаване на големината и стойността на дозимното количество за зареждане;
- режим с последващо заплащане;
- регистрация на получената стойност;
- получаване на информация за товароборота;
- текущ контрол на работата на бензиностанцията.

## УПРАВЛЕНИЕ НА ПЛАНОВО-ИКОНОМИЧЕСКАТА ДЕЙНОСТ В РАЙОННИТЕ НАРОДНИ СЪБЕТИ

Този пакет приложни програми следи за изпълнението на плана по 21 основни в 8 производни икономически показател по предприятията и отрасли в рамките на районния народен съвет. Пакетът дава информация за ритмичността на изпълнението на плана както по месеци, така и за по-дълги отчетни периоди.

## ПАКЕТ ПРИЛОЖНИ ПРОГРАМИ „ТЕКСТ-1031С“

Този програмен продукт е предназначен за автоматизация на работата в машиннописни и кореспондентски бора, в редакции, библиотеки, а също и машиннописните и кореспондентски отдели на министерства, обединения в предприятията. Текст-1031С е екранно-ориентиран пакет приложни програми, удобен за ползуване от потребители без специални познания по изчислителна техника. Модулът за редактиране позволява хоризонтално и вертикално преместване на текста по екрана, функции, позволяващи корекции на документите, автоматично установяване на страници, изравняване на текста отдясно, центровка, пренасяне на думите и други функции. Модулът за печат допуска форматирание на страницата, автоматична номерация и формиране на страници и редци специални ефекти — горни и долни индекси, подчертаване, печатане с курсив и пр. Пакетът съдържа и модул за обединение при печатането, предназначен за създаване на серия от писма с еднакъв текст, но с различен адрес на получателя.

## ИНФОРМАЦИОННА СИСТЕМА „ИНФО-1031С“

Системата е предназначена за създаване и поддържане на база данни. Тя позволява да се проецират входни екрани за извеждане на данни, като потребителят определя дължината на полето и може да контролира тиза на въвежданите данни. Системата осигурява следните възможности на живо запис:

- добавяне и изтриване на запис;
- локализиране на запис посредством ключ;
- сканиране и поддържане на файл;
- локализиране на записи с редактиращи маски;
- групово извеждане на данни.

## ГЕНЕРАТОР ЗА ОТЧЕТИ

Програмният продукт Отчет-1031С е предназначен за изготвяне на отчети с общо предназначение. Той обработва файлове, създадени от системата ИЗОТ-1031С. Възможностите на системата са: генерация на отчети чрез обработката на един файл; генерация на сложни отчети с възможност за редактиране на заглавията на страниците и колонки чрез обработката на няколко файла; нова сортировка при промяна на ключовото поле.

## КОДОВЕ ЗА ПРАВЕЦ-82

Знаците, които чрез клавиатурата се въвеждат или извеждат от компютъра, са кодирани в т. нар. ASCII код. Той е 7-битов и позволява  $2^7=128$  различни кодови комбинации. Тъй като микропроцесорът 6502 е 8-разреден и кодовете на знаците, които се въвеждат чрез клавиатурата, имат винаги единица в най-старшия бит, те започват не от 0, а от 128. Същевременно интерпретаторът кодира знаците с числата от 0 до 127, а кодовете от 128 нагоре използва за ключовите думи на Бейсик.

Знакът, чиито ASCII код е, да речем, X, се отпечатва с функцията CHR\$(X), а пък PRINT CHR\$(102) дава буквата Ф. Лесно можем да надникнем в таблицата на кодовете чрез малка програма.

```
10 FOR I = 32 TO 126
15 A$ = CHR$(I)
20 PRINT I; " "; A$
30 FOR Y = 1 TO 100: NEXT
40 NEXT : END
```

Ето и една „по-интелигентна“ програма, която при натискане на който и да е бутон от клавиатурата написва на екрана кода и съответния му смисъл.

```
10 CALL - 936
20 C = PEEK ( - 16384)
30 IF C < 128 THEN 20
40 POKE - 16368,0
50 PRINT C;
60 POKE 256 * PEEK (41) + PEEK
(40) + PEEK (36) + 4,C
70 CALL - 926
80 GOTO 20
```

Кодовете са разделени на няколко групи:

- контролни (управляващи) кодове;
- режим INVERSE;
- режим FLASH;
- режим NORMAL, в която група е поместена и кириллицата (главни букви). Това е причината кири-

лицата да не може да се отпечатва в режимите INVERSE и FLASH.

Таблицата на кодовете вероятно е едно от справочните пособия, към които програмистът ий-често посяга. При това е значително по-удобно да не разгръща постоянно дебели книги, а да я има отделно отпечатана и приложена към собствения ѝ справочник, така че винаги да му е под ръка.

Смятаме, че оптималното решение е сами да можете да си отпечатвате таблицата, за да не се страхувате, че ще се повреди при продължително ползване. За това ще е необходимо да наберете предлаганата по-долу кратка програма.

```
10 D$ = CHR$(4): PRINT D$ "PR#1"
20 PRINT CHR$(27) "E"
30 FOR I = 1 TO 69: L$ = L$ + "*"
: NEXT : HOME : PRINT
40 A$ = "=====" : HTAB 2
7: PRINT A$: HTAB 27: PRINT
"КОМПЮТЪР ЗА ВАС": HTAB 27: PRINT
A$: HTAB 4 : PRINT "++K040BE
ЗА ПРАВЕЦ-82++": PRINT
50 PRINT SPC(3); "INVERSE"; SPC(
8); "FLASH"; SPC(10); "CONTR
L (MK-)" : SPC(8); "NORMAL": PRINT
L$
60 FOR V = 0 TO 63: FOR N = V TO
V + 192 STEP 64
70 IF N > 159 AND N < 255 THEN X
$ = CHR$(N): BOTO 110
80 IF N = 255 THEN X$ = " ": GOTO
110
90 X = N: X = X + 192 * (N < 32) +
128 * (N > 31 AND N < 96) +
64 * (N > 95)
100 X$ = CHR$(X)
110 PRINT " "; : GOSUB 150: PRINT
" "; X$: " "; N: SPC( (N < 100)
+ (N < 10)); " "; : IF N > 12
7 THEN PRINT " ("; N - 128; " "
"; SPC( (N - 128 < 100) + (
N - 128 < 10));
120 PRINT " "; : NEXT N: PRINT :
IF V + 1 - INT ( (V + 1) /
16) * 16 > 0 THEN 140
130 PRINT : T = T + 1: IF T = 2 THEN
PRINT SPC(3); "INVERSE"; SPC(
8); "FLASH"; SPC(10); "NORMAL
"; SPC(15); "КИРИЛИЦА": PRINT
L$
140 NEXT V: FOR I = 1 TO 6: PRINT
: NEXT : PRINT D$ "PR#0": END
150 PRINT "$": NZ = N / 16: GOSUB
160: NZ = N - NZ * 16: SOSPUB
160: RETURN
160 PRINT CHR$(48 + NZ + 7 * (
NZ > 9)); : RETURN
```

Напечатаната от нея таблица е по-пълна и по-удобна за работа от всички таблици, публикувани у нас като приложения в книги. Тя съдържа всички кодове от 0 до 255 с тяхното десетично и шестнайсетично означение, включително и двойната номерация на кодовете от 128 до 255. Означени са и отделните групи кодове.

Програмата е предвидена за работа с принтер — редове 10 и 20. Командата на ред 20 е предназначена

за матричен принтер от типа на Епсон (за удебелен печат).

За тези, които имат принтер, отпечатваме втората част на таблицата. Тя включва контролните кодове и кодирането на кирилицата, латиницата и преприятелните знаци в режим NORMAL.

Ще допълним, че предлаганият подход за самостоятелно отпечатване на таблицата има и предимството, че ще знаете точно съответствие между кодовете и знаците за вашия компютър и принтер.

# КОДОВЕ ЗА ПРАВЕЦ-82

## CONTROL (МК-) NORMAL

\*\*\*\*\*

\$80 Ю 128 (0)	\$C0 Ю 192 (64)
\$S1 A 129 (1)	\$C1 A 193 (65)
\$82 B 130 (2)	\$C2 B 194 (66)
\$83 C 131 (3)	\$C3 C 195 (67)
\$84 D 132 (4)	\$C4 D 196 (68)
\$85 E 133 (5)	\$C5 E 197 (69)
\$86 F 134 (6)	\$C6 F 198 (70)
\$87 G 135 (7)	\$C7 G 199 (71)
\$88 H 136 (8)	\$C8 H 200 (72)
\$89 I 137 (9)	\$C9 I 201 (73)
\$8A J 138 (10)	\$CA J 202 (74)
\$8B K 139 (11)	\$CB K 203 (75)
\$8C L 140 (12)	\$CC L 204 (76)
\$8D M 141 (13)	\$CD M 205 (77)
\$8E N 142 (14)	\$CE N 206 (78)
\$8F O 143 (15)	\$CF O 207 (79)

\$90 P 144 (16)	\$D0 P 208 (80)
\$91 Q 145 (17)	\$D1 Q 209 (81)
\$92 R 146 (18)	\$D2 R 210 (82)
\$93 S 147 (19)	\$D3 S 211 (83)
\$94 T 148 (20)	\$D4 T 212 (84)
\$95 U 149 (21)	\$D5 U 213 (85)
\$96 V 150 (22)	\$D6 V 214 (86)
\$97 W 151 (23)	\$D7 W 215 (87)
\$98 X 152 (24)	\$D8 X 216 (88)
\$99 Y 153 (25)	\$D9 Y 217 (89)
\$9A Z 154 (26)	\$DA Z 218 (90)
\$9B <del>0</del> 155 (27)	\$DB Ш 219 (91)
\$9C <del>1</del> 156 (28)	\$DC Э 220 (92)
\$9D <del>2</del> 157 (29)	\$DD Ш 221 (93)
\$9E <del>3</del> 158 (30)	\$DE Ч 222 (94)
\$9F <del>4</del> 159 (31)	\$DF <del>223</del> (95)

## NORMAL

## КИРИЛИЦА

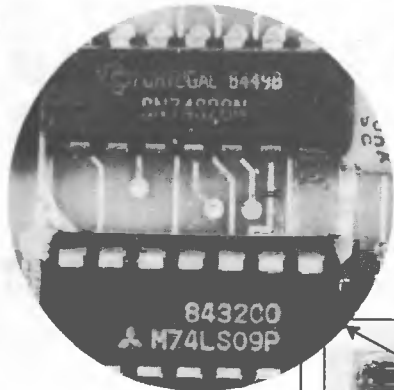
\*\*\*\*\*

\$A0 160 (32)	\$E0 224 (96)
\$A1 ! 161 (33)	\$E1 A 225 (97)
\$A2 " 162 (34)	\$E2 B 226 (98)
\$A3 # 163 (35)	\$E3 Ц 227 (99)
\$A4 \$ 164 (36)	\$E4 Д 228 (100)
\$A5 % 165 (37)	\$E5 Е 229 (101)
\$A6 & 166 (38)	\$E6 Ф 230 (102)
\$A7 ' 167 (39)	\$E7 Г 231 (103)
\$A8 ( 168 (40)	\$E8 X 232 (104)
\$A9 ) 169 (41)	\$E9 И 233 (105)
\$AA * 170 (42)	\$EA Я 234 (106)
\$AB + 171 (43)	\$EB К 235 (107)
\$AC , 172 (44)	\$EC л 236 (108)
\$AD - 173 (45)	\$ED М 237 (109)
\$AE . 174 (46)	\$EE Н 238 (110)
\$AF / 175 (47)	\$EF O 239 (111)
\$B0 0 176 (48)	\$FO П 240 (112)
\$B1 1 177 (49)	\$F1 Я 241 (113)
\$B2 2 178 (50)	\$F2 P 242 (114)
\$B3 3 179 (51)	\$F3 C 243 (115)
\$B4 4 180 (52)	\$F4 T 244 (116)
\$B5 5 181 (53)	\$F5 Y 245 (117)
\$B6 6 182 (54)	\$F6 Ж 246 (118)
\$B7 7 183 (55)	\$F7 B 247 (119)
\$B8 8 184 (56)	\$F8 Ь 248 (120)
\$B9 9 185 (57)	\$F9 Ъ 249 (121)
\$BA , 186 (58)	\$FA Э 250 (122)
\$BB ; 187 (59)	\$FB I 251 (123)
\$BC < 188 (60)	\$FC Э 252 (124)
\$BD = 189 (61)	\$FD I 253 (125)
\$BE > 190 (62)	\$FE - 254 (126)
\$BF ? 191 (63)	\$FF _ 255 (127)

# МОНТИРАНЕ

на

# DRAM ПЛАТКА



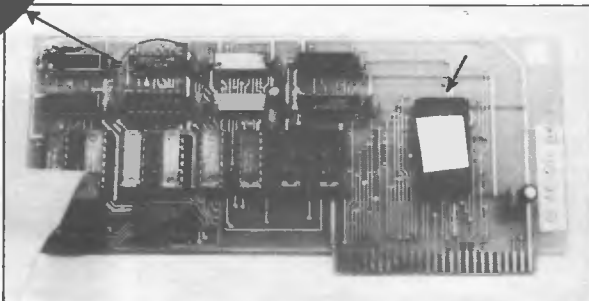
Поставянето на платката за разширение на паметта (Dynamic RAM, DRAM платка или езикова платка) често пъти затруднява специалистите, защото липсват съпътстваща укавания. В резултат не са редки случаите на повреждане на интегрални схеми при неправилното монтиране на платката.

На фиг. 1 е показана драм-платка от по-стария модел, която е и най-широко разпространена. За да се монтира платката, трябва предварително от основната (дълната платка) на компютъра Провец-82 да се извадят две интегрални схеми — EPROM 2716 (крайният откъм страната на захранването със зелен надпис F8) и RAM 4116, също краен в редицата си (фиг. 2). Епром-чипът се поставя в цокъла си на платката, като се внимава при монтираня платка белегът му да бъде насочен нагоре (фиг. 1). Рам-чипът се монтира на цокъла на платката, като белегът му сочи към съединителния кабел. Понякога на платката има такова интегрална схема и извадената ще на остане за резерв.

Самата платка се поставя в слот 0, т. е. най-левия, гледано откъм клавиатурата. Много важно е кулунгът, с който завършва кабелът, да се постави в освободения от рем-чипа цокъл така, както е показано на фиг. 3. Ако се обърне обратно, преместеният рем ще изгоря.

Ето и една хитрост — за да не местите епром-чипа, трябва да прекъснете една шина на печатната платка (нещо, което е сторено при новите платки), както е показано на изнесената позиция. Тази малка операция спестява неудобството от местенето на чипа, без да влияе върху работата на платката.

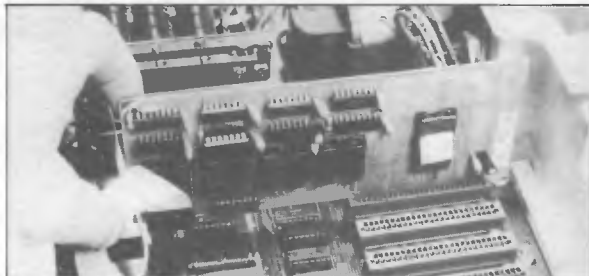
Фиг. 1



Фиг. 2



Фиг. 3



Основната функция на оператора IF е разклоняването на програмата при известно условие, продиктувано от логиката на решавания проблем. Традиционният му формат е:

IF условие  $\left\{ \begin{array}{l} \text{GOTO} \\ \text{THEN GOTO} \\ \text{THEN} \end{array} \right\}$  номер на ред от програмата

И трите конструкции са равностойни и изборът им зависи само от предпочитанията на програмиста.

Условието може да бъде просто или съставно — няколко прости условия, свързани помежду си с AND, OR или NOT.

Пример за просто условие:

```
20 *****
30 IF A=5 OR B>10 GOTO 80
   THEN GOTO 80
   THEN
40 *****
```

Пример за съставно условие:

```
20 *****
30 IF A=5 THEN GOTO 80
   THEN
40 *****
```

И в двата случая изпълнението на условието води до предаване на управлението на ред 80 от програмата. В противен случай се продължава с ред 40, т. е. със следващия ред.

Като допълнителна функция на оператора за условен преход IF може да се разглежда възможността за организиране на програмни цикли, т. е. многократно изпълнение на части от програмата. За това е нужно в тялото на цикъла да се включи увеличаващ или намаляващ брояч, определящ колко пъти да се изпълни цикълът. Броячът се реализира чрез въвеждане на променлива, която периодично (след всяко изпълнение) увеличава или намалява стойността си. Например:

```
60 I = 100
70 IF I < 1 GOTO 110
80 PRINT "ЦИКЪЛ"
90 I = I - 1
100 GOTO 70
```

# ПОВЕЧЕ ЗА ОПЕРАТОРА IF

СИМЕОН НИКОЛОВ

В този програмен фрагмент променливата I представлява увеличаващ брояч с първоначална стойност 1, крайна стойност 10,1 и стъпка 0,1. Създай ефект се постига и при използването на намаляващ брояч  $I=I-X$ , където X е стъпката на цикъла:

```
60 I = 1
70 IF I > 10 GOTO 110
80 PRINT "ЦИКЪЛ"
90 I = I + 0.1
100 GOTO 70
```

И в двата примера с оператора GOTO 70 приключва изпълнението на тялото на цикъла и се отива към оператора IF, който или продължава изпълнението на цикъла, или извежда програмата от това състояние след определен брой изпълнения, като предава управлението на подходящо място.

Важно е да се отбележи, че в оператора IF могат да бъдат включени редица други оператори, които разширяват приложението му.

IF (УСЛОВИЕ) THEN (ОПЕРАТОР)

В този случай използването на THEN е задължително. При този формат на IF не се прави преход, а се изпълнява включеният след THEN оператор, след което се продължава със следващия ред от програмата.

```
50 A = 0 + P
60 IF A = 4 THEN PRINT "ПРАВЕЦ-82"
70 B = 0 + B: GOTO 60
```

Ако в посочения пример условието за A е изпълнено, се отпечатва Правец-82 и се продължава с ред 70. В противен случай също се продължава с ред 70, но без да се отпечата нищо.

```
70 A = 5
80 A = 0 + P
90 IF A > 3 THEN INPUT "ВЪВЕДЕТЕ E=B*I"
   E=B*I
100 M = A + 3
```

Тук при условие, че  $A > 3$ , може да се изведе стойност за B и изчислението да продължат с ком. При неизпълнение на условието преходът за M се изчислява за  $B = 5$ .

По аналогичен начин се изпълняват и останалите оператори, включени след THEN, с изключение на GOSUB, който извършва преход към подпрограмата.

Възможно е и следното решение:

```
90 IF M = 0 THEN PRINT "ИЗРАЗЪТ
   НЯМА РЕШЕНИЕ.": GOTO 250
100 Y = (A + B) * SIN (3.14 / 6)
   / M
```

Тук при неизпълнение на условието управлението се предава не на оператора, следващ IF непосредствено (разделен с :), а на следващия ред от програмата, в случая 100. С други думи, ако условието не е удовлетворено, всички оператори, записани на същия ред с оператора IF, няма да бъдат изпълнени. Ако  $M = 0$ , ще бъде отпечатан зададеният текст и ще се премине към ред 250.

Всичко, описано дотук, увеличава свободата при решаването



на един или друг проблем и създава възможност за разнообразяване на логиката на програмата. Следващите два примера, които са части от програмата за решаване на квадратно уравнение, убеждават, че уметелото използване на възможностите на оператора IF в много случай прави програмата по-прегледна.

```

90 D = B * B - 4 * A * C
100 IF D > 0 GOTO 130
110 IF D = 0 GOTO 140
120 PRINT "УРАВНЕНИЕТО НЯМА РЕАЛНИ
КОРЕНИ" GOTO 150
130 PRINT "X1="; "1 - B + SQR (D)
/ (2 * A); "X2="; "1 - B - SQR
(D) / (2 * A); GOTO 150
140 PRINT "X1=X2="; "1 - B / (2 * A)
"
150 END

```

```

90 D = B * B - 4 * A * C
100 IF D < 0 THEN PRINT "УРАВНЕ
НИЕТО НЯМА РЕАЛНИ КОРЕНИ" GOTO
150
110 P = - B / (2 * A)
120 IF D = 0 THEN PRINT "X1=X2=
"; P; GOTO 150
130 Q = SQR (D) / (2 * A)
140 PRINT "X1="; P + Q; "X2="; P -
Q
150 END

```

### ДОПЪЛНЕНИЕ НА РЕДАКЦИЯТА:

Условия, които представляват изрази, включващи променливи и числови константи, могат да бъдат зададени, без да се използва знак за отношение (например = или <>). Използувани в оператор IF, тези условия се приемат за удовлетворени, ако резултатът от пресмятането на израза е нула. При всички други стойности се приема, че условиято не е изпълнено. Например

```
IF A = 12 THEN GOTO 100
```

е рязносилно на

```
IF A = 12 THEN GOTO 100
```

И в двата случая се осъществява преход към програмния ред 100 само ако променливата A е равна на 12.

# РАЗШИРЯВАНЕ НА ПАМЕТТА

## ГЕНЕРИРАНЕ НА СИГНАЛ

### \* INHIBIT

## В МОДУЛ 16 КБАЙТА

(продължение от брой 2—3/1986 г.)

Схемата за изработване на сигнал \*INHIBIT в модул 16 Кбайта е показана на фиг. 4. В таблица 1 са дадени означенията на използваните сигнали, перата на периферните мулдуци, на които те възникват, и тяхната функция.

Условието за генериране на сигнал \*INHIBIT е на четирите входа на схемата U7, реализираща функция и-не, да има високо ниво. За това е необходимо:

1) A15.A14.(A13+A12)=1, което означава, че адресът, пресметнат от компютъра, е от областта \$D000—\$FFFF. Това условие винаги е изпълнено, защото по принцип не е необходимо да се генерира сигнал \*INHIBIT при обръщане към друга област от паметта на компютъра.

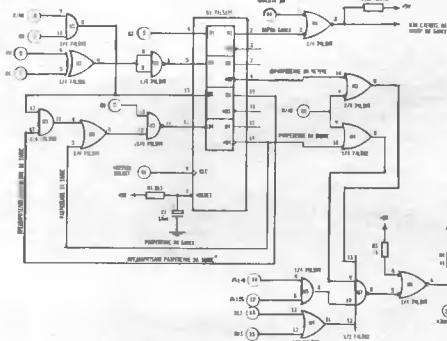
2) При операция четене (ниско ниво по шина R/\*W) тригер D2 да е установен.

3) При операция запис (високо ниво по шина R/\*W) тригер D4 да е нулиран.

Видно се, че схемата за изработване на сигнал \*INHIBIT се основава на трите тригера D2-D4. Тригерът D1 също играе важна роля в управлението на модул 16 Кбайта, защото определя кой от двете банки по 4 Кбайта е достъпна. При включване на захранването и четирите тригера се нулират от късия отрицателен импулс, изработен от групата R1C1. Това означава, че о избрана втора банка (наход 01 е нулирана), разрешен е записът в модула (наход \*Q4 е установен) и е забранено четенето от него (наход \*Q2 е установен).

Всичко превключване на ниво от тези тригери сменя режима на достъп до паметта на модула. И тъй като такава смяна е възможна само под програмния контрол, от съществено значение за правилното използване на модула е да се усвои механизъмът за неговото управление:

1) Информационната, подадена на че-



Фиг. 4 ПОРЪКОВАНИЕ НА СИГНАЛ \*INHIBIT В МОДУЛ 16 КБАЙТА

тирите D-входа, се записва в триггерите во предния фронт на сигнала \*DEVICE SELECT. С други думи, тези триггери са разположени на адреси \$C080-\$C08F.

2) На входа на тригер D2 след приемането през схема, реализираща функция изключващо или-не, се подават сигнали A0 и A1. Така той се установява с обръщение към който и да е от адресите \$C080, \$C083, \$C084, \$C087, \$C088, \$C08B, \$C08C и \$C08E, а се нулира с \$C081, \$C082, \$C085, \$C086, \$C089, \$C08A, \$C08D и \$C08E.

4) Тригерът D3 има спомагателна роля. Той се установява при операции четене от всяко от четните адреси на областта \$C080-\$C08F и се нулира отново при операции четене, но от четен адрес на същата област.

5) Управлението на тригер D4 е малко по-сложно, защото неговото установяване е свързано с предварителното установяване на тригер D3. И така тригер D4 се установява с две или повече последователни операции четене от всяко от четните адреси на областта, определена с \*DEVICE SELECT (\$C080-\$C08F) и се нулира с обръщение към който и да е от четните адреси на същата област.

В таблица 2 са обобщени условията за управление на режима на достъп до модул 16 Кбайта.

### III. ГЕНЕРИРАНЕ НА СИГНАЛ \*INHBIT В МОДУЛ 128 КБАЙТА

Схемата за генериране на сигнал \*INHBIT в модул 128 Кбайта не се различава съществено от схемата, използвана в модул 16 Кбайта. Малкото различие се дължи на допълнителния програмно управляем ключ, необходим за включване на 8-те банки по 16 Кбайта. Ето защо няма да се свършва подробно на принципната схема, а ще разгледаме само как се установява режимът на достъп до паметта на модула.

В заключение трябва още веднъж да се каже, че използването на двата модула се извършва само под програмен контрол, и то от програма, написана на машинен език. Друга особеност на модулите е, че натискането на клавиш RESET не променя режима на достъп до тях. Ако такава промяна е необходима, тя трябва да се извърши от подпрограмата, обръщаща RESET.

Инж. ПЕТЪР ПЕТРОВ  
ИТКР — БАН  
направление „Персонални  
компютри“

Таблица 1. СИГНАЛИ ЗА ИЗРАБОТВАНЕ НА \*INHBIT

Име	Перо	Значение
A0	2	Бит 0 от адресната шина на компютъра. Заедно със сигнали A1 и A3 установява програмно управляемите ключове, определящи режима на достъп до паметта на модула.
A1	3	Бит 1 от адресната шина на компютъра. Виж A0.
A3	5	Бит 3 от адресната шина на компютъра. Виж A0.
A12	14	Бит 12 от адресната шина на компютъра. Заедно със сигнали A13, A14 и A15 се използва за дешифриране на областта \$D000-\$FFFF от адресното пространство на компютъра.
A13	15	Бит 13 от адресната шина на компютъра. Виж A12.
A14	16	Бит 14 от адресната шина на компютъра. Виж A12.
A15	17	Бит 15 от адресната шина на компютъра. Виж A12.
R/*W	18	Сигнал четене/запис. Участва както в установяването, така и при дешифрирането на състоянието на програмно управляемите ключове.
*DEVICE SELECT	41	Сигнал, получен от дешифрирането на областта \$C000-\$C0FF от адресното пространство на компютъра. Той е номерът на периферния кулунг, в който е поставен модулът, плюс 8. Прето е модул 16 К да се поставя в кулунг #0, така че сигнала *DEVICE SELECT се получава от дешифрирането на адресите \$C080-\$C08F. В конкретния случай се използва за адресиране на четирите програмно управляеми ключа.

Таблица 2. Установяване на достъп до модул 16 Кбайта

Адрес		ПОСЛЕДИЦИ ОТ ОБРЪЩЕНИЕТО КЪМ ТОЗИ АДРЕС
ВТОРА БАНКА ПЪРВА БАНКА		
\$C080	\$C084 \$C088 \$C08C	Забранява запис в паметта на модула и разрешава четенето от нея.
\$C081	\$C085 \$C089 \$C08D	Забранява четенето от паметта на модула. Две или повече операции четене от този адрес разрешават запис в модула. Операция запис на този адрес забранява запис в модула.
\$C082	\$C086 \$C08A \$C08E	Забранява четенето и запис на модула.
\$C083	\$C087 \$C08B \$C08F	Разрешава четенето от паметта на модула. Две или повече операции четене от този адрес разрешават запис в модула. Операция запис на този адрес забранява запис в модула.

ТАБЛИЦА 3. УСТАНОВЯВАНЕ НА ДОСТЪП ДО МОДУЛ 128 КБАЙТА

АДРЕС		ПОСЛЕДИЦИ ОТ ОБРЪЩЕНИЕТО КЪМ ТОЗИ АДРЕС
ВТОРА БАНКА ПЪРВА БАНКА		
\$C080	\$C088	Забранява запис в паметта на модула и разрешава четенето от нея.
\$C081	\$C089	Забранява четенето от паметта на модула. Две или повече обръщания към този адрес разрешават запис в модула.
\$C082	\$C08A	Забранява четенето и запис в паметта на модула.
\$C083	\$C08B	Разрешава четенето от паметта на модула. Две или повече обръщания към този адрес разрешават запис в модула.
\$C084-\$C087	\$C08C-\$C08F	Младшите три бита (D2-D0) на байта, написан на който и да е от тези адреси, определя номера на избраната банка от 16 Кбайта.



## АСЕМБЛЕР

И

## МАШИНЕН

(продължава от бр. 2—3)

АДРЕСИРАНЕ  
НА ОПЕРАНДИТЕ

При адресирането на вътрешната памет минималната адресируема единица (клетка) за МП 6502 е с дължина един байт, използва се и поименното двойна дума, което означава последователни адреси. Характерното при МП 6502 е, че младшият байт от двойната дума е байтът с по-малък адрес. Например, ако в две последователни клетки са записани стойностите 2А и 6С, това представлява двойна дума със стойност \$6С2А.

Друга особеност при адресирането на паметта е специфичната употреба на клетки от нулевата страница (с адреси от 0 до 255), без които някои от начините на адресация на МП 6502 са невъзможни. С това се обяснява „загостота“ на нулевата страница от интерпретатора на Бейсик и монитора.

При извършването на аритметични и логически операции, при сравнения и други действия микропроцесорът обработва два операнда. Единият от тях обикновено е съдържанието на регистратора, на регистрите X или Y, а другият — съдържанието на клетка от паметта (ще го наричаме просто *адрес*). При посочването на непосредствен операнд (*написано по-нататък*) той се намира веднага след кода на операцията (КОП), така че за да го извлече, микропроцесорът трябва да изведе на адресните си шини действителния адрес, който е с 1 по-голям от адреса на байта с КОП.

В повечето от случаите обаче адресът на операнда или на клетката, в която микропроцесорът ще записва, е съвсем различен от адресите на програмата. Например една програма е разположена от адрес \$300 до \$350, а в нея има инструкция за четене от клетка с адрес \$200. Илюстрация, че по време на изпълнение на една машина програма, микропроцесорът

ПРОФЕСИОНАЛНО  
КРЪЩЕНИЕОРЛИН ВЪЛЧЕВ  
БОРИСЛАВ ЗАХАРИЕВ

## ЕЗИК

ът определя различен адрес, в зависимост от това, дали прочитан КОП и следващите го байтове от инструкцията, или адресира някакъв операнд от паметта. Конкретният начин за определяне на действителния адрес на операнда от паметта се нарича *начин на адресация*.

При описването на начините за адресация при МП 6502 ще използваме следните означения:

С КОП е отбелязана числената стойност на КОП (от 0 до 255), която е записана в паметта. Микропроцесорът прочита и разпознава КОП и използва съответната операция.

С МКОП е означен мнемоничният код на операцията, както е в езика Асемблер. Например мнемоничният код на командата Нуларий *бита* за адресиране е CLC. В паметта на компютъра тази команда се записва с двоичната стойност 00011000 (\$18).

Дължината на инструкцията показва конкретния брой на байтовете (1, 2 или 3) в паметта.

Асемблерският запис показва как изглежда инструкцията, написана на езика Асемблер.

С АДРЕС е означен някакъв адрес от паметта (от \$0 до \$FFFF), който в асемблерския запис може да бъде и именован.

БАЗА е адрес на клетка от нулевата страница (от \$0 до \$FF) или условно име на клетка от тази страница при асемблерския запис.

1. Адресиране с погрозбирване на акумулатора.

Формат: КОП

Дължина: 1 байт

Асемблерски запис: МКОП

Командата се отнася за акумулатора, без да е необходимо обръщане към паметта. Възможни са пренесване наляво, надясно и циклично пренесване (ротация) на съдържанието на акумулатора.

2. Адресиране с погрозбирване на регистър.

Формат: КОП

Дължина: 1 байт

Асемблерски запис: МКОП

Командата се отнася до някои от регистрите на микропроцесора (X, Y) и служи за установяване или ну-

диране на флаговете, при стекова операция и т. н.

3. Непосредствено адресиране

Формат: КОП ОПЕРАНД

Дължина: 2 байта

Асемблерски запис: МКОП ОПЕ-

РАНД

Вторият байт от инструкцията (след КОП) служи за операнд. Поради това, че не се губи време за определяне на адреса на операнда (той е стойността на програмния брояч РС плюс 1), тези инструкции се изпълняват бързо — за два машинни цикъла (такта), т. е. за две микро-секунди.

4. Относително адресиране спрямо РС

Формат: КОП МЛ. АДРЕС СТ. АДРЕС

Дължина: 3 байта

Асемблерски запис: МКОП АДРЕС

Вторият и третият байт от инструкцията образуват 16-битов действителен адрес. Ако инструкцията е за запис, се пише в клетката с този адрес. При останалите инструкции там е операндът.

5. Пръко адресиране в нулевата страница.

Формат: КОП АДРЕС НА КЛЕТКА ОТ НУЛЕВА СТРАНИЦА

Дължина: 2 байта

Асемблерски запис: МКОП БАЗА

Операндът се намира в клетка от нулевата страница, с адрес втория байт от инструкцията. Характерното е, че инструкцията се изпълнява по бързо от съответната инструкция при разширена адресация.

6. Пръко адресиране в нулевата страница с индексирване по X.

Формат: КОП АДРЕС НА КЛЕТКА ОТ НУЛЕВАТА СТРАНИЦА

Дължина: 2 байта

Асемблерски запис: МКОП БАЗА, X

Базата е адрес на клетка от нулевата страница, чиято съдържание се прибавя към съдържанието на регистъра X. Преносът от събирането се пренебрегва, поради което старшият байт на нововъзникващия адрес е винаги 0. Младшият байт от адреса е действителният адрес на операнда, например се в нулевата страница. Например, ако съдържанието на регистъра X е \$5, а вторият байт от инструкцията е \$20, адресът на операнда ще бъде \$20+\$5=\$25. Ако обаче съдържанието на регистъра X е \$F8, при същата инструкция сумата ще бъде \$20+\$F8=\$118, но тъй като преносът се пренебрегва, действителният адрес е \$13.

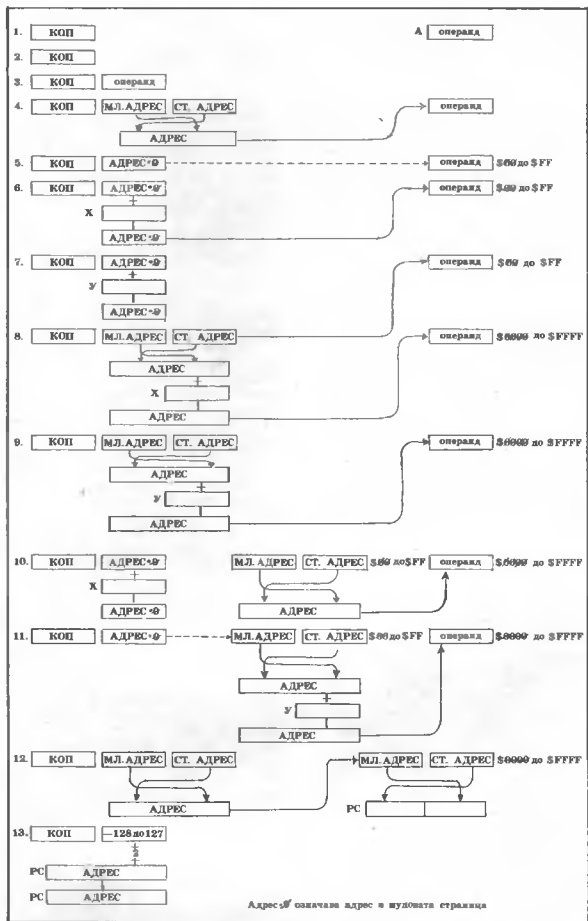
7. Пръко адресиране в нулевата страница с индексирване по Y.

Формат: КОП АДРЕС НА КЛЕТКА ОТ НУЛЕВАТА СТРАНИЦА

Дължина: 2 байта

Асемблерски запис: МКОП БАЗА, Y

Инструкцията е аналогична на предишната, само че за получаване на действителния адрес на операн-



Адрес/А означава адрес в нулевата страница

да се взема съдържанието на регистра У.

В. Пряко пълно адресиране.

Формат: КОП МЛ. АДРЕС СТ. АДРЕС

Дължина: 3 байта

Асемблерски запис: МКОП АДРЕС, X

Действителният адрес на операнда се получава, като към втория и третия байт се прибавя съдържанието на регистра X с отчитане на преноса.

Например инструкцията LDA \$2028 (зареди акумулатора със съдържанието на клетка с адрес \$2028) може да се запише като LDA \$2000, X, а съдържанието на X е \$26. Чрез промяна на регистра X програмистът има достъп до 256 последователни клетки от па-

метта (ограничението се налага от дължината на регистра — осем бита). Та могат да бъдат разположени в различни страници от паметта. Например, ако вторият и третият байт от инструкцията са \$A0 \$42 (адрес \$42A0), чрез промяна на X от 0 до 255 (\$FF) програмистът може да адресира клетките с адреси от \$42A0 до \$439F.

9. Пряко пълно адресиране с индексиране по X

Формат: КОП МЛ. АДРЕС СТ. АДРЕС

Дължина: 3 байта

Асемблерски запис: МКОП. АДРЕС, У

Инструкцията е аналогична на предходната, но за определяне на адреса се взема съдържанието на регистра У.

10. Пряко пълно адресиране с индексиране по Y

Формат: КОП АДРЕС НА КЛЕТКА ОТ НУЛЕВАТА СТРАНИЦА

Дължина: 2 байта

Асемблерски запис: МКОП (BA-3A, X)

Вторият байт от инструкцията се прибавя към съдържанието на регистра X, като преносът се пренебрегва. Полученият резултат посочва клетка от нулевата страница. Съдържанието на посочената клетка и съдържанието на клетката със следващ адрес образуват 16-битовия адрес на операнда.

Пример. Вторият байт от инструкцията е \$14, а съдържанието на регистра X е \$8. Адресът на операнда ще се получи така: \$14+\$8=\$1C. Съдържанието на клетка с адрес \$1C представлява младшите, а съдържанието на \$1D (адрес \$1C плюс 1) — старшите осем бита на действителния адрес. Така формиран адресът се изпраща от микропроцесора към паметта за прочитането на операнда или за запис в тази клетка.

Обръщаме внимание, че с тази инструкция може да се адресира цялата памет (от \$0 до \$FFFF), променяйки съдържанието на двете последователни клетки от нулевата страница, в които е записан адресът на операнда. Трябва много да се внимава за съдържанието на регистра X. Всяка негова промяна ще породи прешащисляване на адресите на двойката клетки. Голямо предимство на инструкцията е възможността програмистът да адресира различни области от паметта само с промяна на регистра X. Разбира се, двойните клетки от нулева страница трябва да бъдат предпазливо заредени с адресите на областите.

11. Косвено пълно адресиране

Формат: КОП АДРЕС НА КЛЕТКА ОТ НУЛЕВАТА СТРАНИЦА

Дължина: 2 байта

Асемблерски запис: МКОП (BA-3A), У

Като адресацията е предякдексирана, така и тази е едни от най-мощните начини на адресиране при МП 6502.

Адресът на операнда се получава така: от съдържанието на клетката от нулевата страница, посочена в инструкцията, и съдържанието на следващата клетка (само от нулевата страница) се формира 16-битов адрес. Към него се прибавя съдържанието на регистра У.

Например вторият байт от инструкцията е \$FB, а съдържанието на У е \$40. Прочета се съдържанието на \$FB и \$FC (\$FB + 1), да кажем, 00 20. Това представлява адрес \$2000, към който се прибавя \$40 (съдържанието на регистра У). Получава се адрес \$2040, който се изпраща от микропроцесора по адресните шини към паметта.

Обръщаме внимание, че промяната на У позволява адресирането на 256 последователни клетки, а промяната на съдържанието на двойката клетки от нулева страница, в които е за-



писан адресът, прави възможност адресирането на цялата памет.

12. Косвенно пълно адресиране с преобръщане на индексирани по X.

Формат: КОП МЛ. АДРЕС СТ.

АДРЕС

Дължина: 3 байта

Асемблерски запис: МКОП (АДРЕС)

Само една инструкция използва този начин за адресация — инструкция за безусловен преход (JMP). При нея не се адресира операнд, а се презарежда програмният брой (PC). Следващата инструкция от програмата, че се вземеле от клетка с адрес, равен на новото съдържание на PC, т. е. извърши преход. Вторият и третият байт от инструкцията посочват адреса на клетка, с чийто съдържание се зарежда младият байт на PC (PCL). Съдържанието на клетката със следващ адрес се зарежда в старшата част на PC (PCN).

Пример: JMP (\$5566). Съдържанието на клетка \$5566 (например \$20) се зарежда в PCL, а съдържанието на \$567 (например \$08) — в PCN. Следващият КОП ще се прочете от адрес \$0820, т. е. извърши се безусловен преход към адрес \$0820.

13. Косвенно пълно адресиране с последващо индексирани по Y.

Формат: КОП ОТМЕСТВАНЕ

Дължина: 2 байта

Асемблерски запис: МКОП АДРЕС

Възможна е само при инструкциите за условен преход. При тях се извършва преход в програмата при наличие на някакво условие (състо-

ние на флаг) в регистъра за състоянието. След разпознаване на инструкцията PC се увеличава с 2 (дължината на инструкциите за условен преход е 2 байта) и се проверява дали е изпълнено даденото условие. Ако то не е изпълнено, не се предприема нищо, т. е. продължава се със следващата инструкция. Ако е изпълнено, вторият байт от инструкцията — отместването, се прибавя към съдържанието на PC:

$PC = PC + 2 + \text{ОТМЕСТВАНЕ}$

Събираемият 2 трябва да се вземе предвид, защото PC е вече увеличен след разпознаването на инструкцията. Отместването е осемразрядно число от 0 до 255 (от \$0 до \$FF), но отчита се в допълнителен код, т. е. \$0...\$7F (0...127) са положителни числа (старшият знак бит нула), а \$80...\$FF (-128...-1) са отрицателни (старшият знак бит единичка). Иско е, че границите на прехода при относителната адресация са от PC + 2 — 128 до PC + 2 + 127 или от PC — 126 до PC + 129.

Пример. Като втори байт от инструкцията е записано числото \$11 и условието от проверката е изпълнено. PC ще се промени с  $\$2 + \$11 = \$13$  (19) и следващата инструкция ще се вземеле от адрес, с 19 по-голям от адреса на текущата инструкция (за условен преход), т. е. има преход напред в програмата при някакво условие. Ако вторият байт е числото \$F6, адресът на прехода ще бъде  $PC + \$2 + \$F6 = PC + \$F8$  (\$F8 в допълнителен код е числото — \$8) или че се извърши преход назад с 8 байта.

При асемблерския запис адресът, който може и да е именован, трябва да бъде в диапазона от —128 до 129 байта от адреса на инструкцията за условен преход. В противен случай че последва съобщение за грешка от транслятора.

Пример:

...  
BCC ET1

...  
ET1 ...

Командата BCC означава преход при бита за пренос (C)=0, към адрес, който е именован ET1.

Ако инструкцията за условен преход е разположена на адрес \$310, а адресът на имените ET1 е \$3B0, преходът не може да се извърши: искаме преход напред с SAO (160) байта, а най-големият възможен преход напред е \$7F + \$2 (129) байта.

Времето за изпълнение на инструкциите за условен преход е:

— 2 микросекунди, ако не е извършва преход (неизпълнено условие);

— 3 микросекунди при преход в текущата страница от паметта, т. е. без промяна на старшите 8 бита на адреса;

— 4 микросекунди при преход в съседна страница (например, ако инструкцията е на адрес \$51F7, а преходът — на адрес \$5215).

Инструкциите за условен преход дават възможност за написването на т. нар. *претемаваци програми*. Те се изпълняват независимо от началния адрес на зареждане, тъй като адресите на преходите се определят от текущото съдържание на PC.

**БЕЛЕЖКА  
НА РЕДАКТОРА**

## КЛЕТКА И ДУМА

Клетка в един компютър е минималното количество свързани (съседни) битове във вътрешната му памет, което може да бъде адресирано. Не се казва клетка, ако това е един бит.

Поради различната структура на паметовите вътрешни памет в един компютър може да има клетки с различна дължина. В ЕС микропрограмазната клетка (от постоянната памет) е 64-битова, а тази в ОП е с дължина 1 байт. В Интел-

432 адресацията в програмната ОП е до бит, а тази в ОП за данни — до байт. В 8-битовите микропроцесори съставящите на вътрешната памет — постоянната и оперативната памет — са с еднаква структура и се състоят от 8-битови клетки.

Дума в един компютър е количеството свързани битове, което може едновременно да бъде обработено. Обикновено в големите компютри има няколко различни по дължина думи. В ЕС има дума (4 байта,

т. е. думата се разполага в 4 последователни клетки на ОП), полудума (2 байта) и двойна дума (8 байта). В мощните 16- и 32-битови микропроцесори от III и IV поколение различните думи са с дължина 1, 2, 4 и 8 байта.

И така клетката и думата са две съществено различни понятия — клетката е място във вътрешната памет и е свързана с адресацията, докато думата е данна с фиксирана дължина, което е свързано с обработката на данни извън вътрешната памет например в акумулатор, служебен регистър или регистър с общо предназначение.

В 8-битовите микропроцесори дължините на клетката и думата (а не самите понятия) съвпадат — те са 1 байт. Това съпадение (на дължините) води най-често до объркване на двете понятия.

## ДИСК - ФИКСЕР

От дълго време търся указания за работа с програмата Дискфиксер, но не мога да намеря. Вихте ли ми помогнали?

Орлин Велев — Пловдив

За да помогнем, потърсихме нашите научни консултанти, които с доста усилия успяха да съставят таблица с командите за тази програма.

Дискфиксер е изключително полезна приложна програма и в ръцете на опитния програмист е мощен инструмент. На начинаещите тя също може да е много полезна, защото позволява да се види на физическо равнище как DOS организира записването на данните върху дискетата. Така например норедиката за DOS, която се отпечатва във вестник „Направи сам“, и особено статията в 8 и 9 брой от 1985 г. може да бъдат идеално овладени с тази програма.

Дискфиксер позволява да се четат отделни сектори от дискетата, както и да се прегледа избраният файл сектор по сектор. Тя показва съдържанието на дискетния указател и адреса на началния сектор (списъка виста-сектор), чете таблицата за съдържанието на тома (TST). Информацията за съдържанието на секторите се изобразява в шестнайсетичен код или в знаци.

Най-важното е, че с Дискфиксер могат да се внасят корекции или нова информация, които се записва обратно върху същата или върху друга дискета. С нея файлчето може да се преименува, отключат и заключат, да се смени известяващата програма, да се правят корекции и да се възстановява TST, да се открие и евентуално да се премести на обичайното му място не-обичайно разположеният дискетен указател (каталогът), да се прочетат и запишат отново проблематични за четене сектори и т. н.

Областите на приложение са твърде много и броят им е пропорционален на уменията на програмиста.

Ще добавям, че Дискфиксер не може да помогне при физически повредени сектори, както и не чете адресните полета на секторите.

В някои от следващите броеве ще разгледаме тънкости при работата с програмата.

### КОМАНДИ НА ДИСКФИКСЕР

01 0 до 7	- ИЗБИРА БУФЕР ЗА РАБОТА.
H	- ЧЕТЕ В ТЕКУЩИЯ БУФЕР УКАЗАНИЯ СЕКТОР ОТ ИЗБРАНАТА ЛИСТА.
W	- ЗАПИСВА СЕКТОР ОТ ТЕКУЩИЯ БУФЕР.
E	- ЗАПИСВА СЕКТОР ОТ ТЕКУЩИЯ БУФЕР ВЪРХУ ДРУГО ДИСКЕТНО УСТРОЙСТВО.
S	- ЗАДАВА НОМЕРАТА НА СЛОТА И ДИСКЕТНОТО УСТРОЙСТВО, С КОЕТО ШЕ СЕ РАБОТИ.
L	- НАСТРОБВА ПРОГРАМАТА ЗА РАБОТА С 13- ИЛИ 16-СЕКТОРНО ФОРМАТИРАНА ДИСКЕТА.
MK+C	- ИЗБИРА ЛОГИЧЕСКИ ИЛИ ФИЗИЧЕСКИ 16 СЕКТОРА.
B	- ИЗВЕЖДА ИНФОРМАЦИЯТА В ШЕСТНАЙСЕТИЧЕН И СИМВОЛЕН ФОРМАТ НА ДВА ЕКРАНА.
0	- ИЗВЕЖДА ЦЕЛИЯ ТЕКУЩ БУФЕР В СИМВОЛЕН ФОРМАТ.
h	- ИЗВЕЖДА ТЕКУЩИЯ БУФЕР В ШЕСТНАЙСЕТИЧЕН ФОРМАТ.
Q	- ИЗВЕЖДА ПЪРВИЯ ЕКРАН (БАЙТОВЕ \$00 ДО \$7F).
q	- ИЗВЕЖДА ВТОРИЯ ЕКРАН (БАЙТОВЕ ОТ \$80 ДО \$FF).
!	- ИЗВЕЖДА СЪДЪРЖАНИЕТО НА БУФЕРА КАТО ПОЗИЦИОНИРА УКАЗАНИЯ БАЙТ ПЪРВИ НА ЕКРАНА.
MK+D	- ПРОЧЕТА СЛЕДВАЩИЯ СЕКТОР ОТ ДИСКЕТАТА ИЛИ ФАЙЛА.
MK+H	- ПРОЧЕТА ПРЕДИШНИЯ СЕКТОР ОТ ДИСКЕТАТА ИЛИ ФАЙЛА.
I	- ДВИЖИ КУРСОРА НАГОРЕ.
J	- ДВИЖИ КУРСОРА НАЛЯВО.
K	- ДВИЖИ КУРСОРА НАДОЛНО.
M	- ДВИЖИ КУРСОРА НАОДОУ.
L	- ДИРЕКТНО ПОЗИЦИОНИРАНЕ НА КУРСОРА ВЪРХУ ИЗБРАН БАЙТ.
U	- ПОЗИЦИОНИРАНЕ НА КУРСОРА В НАЧАЛОТО НА ЕКРАНА.
<RETURN>	- НАЧАЛО НА ВЪВЕЖДАНЕ НА ПРИМЕНИ.
<ESC>	- ПРЕКРАТЯВА РЕЖИМА НА КОРЕКЦИИ. ОТМЕНЯ ПРЕДИШНАТА КОМАНДА.
:	- ПОЗВОЛЯВА ПРОМЯНА САМО НА ТЕКУЩИЯ БАЙТ.
A	- ЗАПЪЛВА ДО КРАЯ ТЕКУЩИЯ БУФЕР СЪС СТОЯНОСТТА НА БАЙТА. ВЪРХУ КОЙТО СЕ НАМИРА КУРСОРА.
Q	- ДУБЛИРА ТЕКУЩИЯ БАЙТ, КАТО ИЗМЕСТВА ВСИЧКИ СЛЕДВАЩИ БАЙТОВЕ С ЕДИН НАДЯСНО. ПОСЛЕДНИЯТ БАЙТ СЕ ГЪБИ.
I	- ИЗТРИВА БАЙТА, ВЪРХУ КОЙТО СЕ НАМИРА КУРСОРА. ПОСЛЕДНИЯТ БАЙТ В БУФЕРА СЕ ДУБЛИРА.
X	- ВЪЗСТАНОВЯВАНЕ НА ОРИГИНАЛНАТА ИНФОРМАЦИЯ /АНУЛИРАНЕ НА ПРОМЕНЕТЕ/ В БУФЕРА.
-	- ВЪЗСТАНОВЯВАНЕ НА ТЕКУЩИЯ БАЙТ.
N	- ИЗВЕЖДА ЧИСЛЕНАТА ИНФОРМАЦИЯ ДЕСЕТИЧНО ИЛИ ШЕСТНАЙСЕТИЧНО.
F	- ВКЛЮЧВА И ИЗКЛЮЧВА ФИЛТЪРА ПРИ ИЗВЕЖДАНЕ НА ГРАФИЧНИ СИМВОЛИ, СЪОТВЕТСТВАЩИ НА ШЕСТНАЙСЕТИЧНИТЕ СТОЯНОСТИ НА БАЙТОВЕТЕ.
T	- ПОЗИЦИОНИРА ФИЛТЪРА.
:	- ПОЗИЦИОНАТА УКАЗАН СЕКТОР В РЕЖИМ "ФАЙЛ" /ЛОГИЧЕСКО ТЪРСЕНЕ СЛЕД КОМАНДИ 0-6-<ФАЙЛ>.
7	- ИЗВЕЖДА В ТЕКУЩИЯ БУФЕР БАЙТОВЕ СЪС СТОЯНОСТИ ОТ \$00 ДО \$FF.
P	- ПАУЗА.
MK+Q	- ИЗХОД КЪМ МОНИТОР.



МК+У<RETURN>- ВРЪЩАНЕ ОТ МОНИТОР В ДИСКФЛОПДИСК.  
 <RST> - ИЗХОД КЪМ ДОС.  
 У - РАБОТА С ТАБЛИЦАТА ЗА СЪДЪРЖАНИЕ НА ТОМА  
 /VTOC/.

<ИНТЕРВАЛ>- ИЗВЕЖДА ТСТ В РЕДАКИРАН ФОРМАТ.  
 R - ПРОЧИТА ТСТ.  
 W - ЗАПИСВА ТСТ.  
 F - КОРИГИРА ТСТ СПОРЕД ДЕЙСТВИТЕЛНО ЗАЕТИТЕ ОТ  
 ФАЙЛЛОВЕ СЕКТОРИ.  
 M - ПОКАЗВА ЗЕТИТЕ ОТ ВСЕКИ ФАЙЛ СЕКТОРИ.  
 E - ПОЗВОЛЯВА РЕДАКИРАНЕ НА ТСТ.  
 - МАРКИРА СЕКТОРИТЕ ПОД КУРСОРА ОТ ТЕКУЩАТА ПИСТА  
 КАТО ЗАЕТИ.  
 - МАРКИРА СЕКТОРИТЕ ПОД КУРСОРА ОТ ТЕКУЩАТА ПИСТА  
 КАТО СВОБОДНИ.  
 / - ПРОМЕНЯ СЪСТОЯНИЕТО НА СЕКТОРИТЕ ПОД КУРСОРА  
 ОТ ПИСТАТА ОТ СВОБОДНИ В ЗАЕТИ И ОБРАТНО.

<ИНТЕРВАЛ>- ПРОМЕНЯ СЪСТОЯНИЕТО НА СЕКТОРА, ВЪРХУ КОЙТО СЕ  
 НАМИРА КУРСОРА ОТ ЗАЕТИ В СВОБОДНИ И ОБРАТНО.

D - ПОЗВОЛЯВА РАБОТА С КАТАЛОГА НА ДИСКЕТАТА.  
 <ИНТЕРВАЛ>- ИЗВЕЖДАНЕ НА КАТАЛОГА: ИМЕНА НА ФАЙЛЛОВЕТЕ, ТИП,  
 ДЪЛЖИНА, АДРЕС НА СПИСЪКА ПИСТА/СЕКТОР. ИЗВЕЖДА  
 И ИМЕНАТА НА МАРКИРАНИТЕ КАТО ИЗТРИТИ ФАЙЛЛОВЕ.

- ПРОЧИТА КАТАЛОГА.  
 W - ЗАПИСВА КАТАЛОГА.  
 S - СОРТИРА ФАЙЛЛОВЕТЕ В КАТАЛОГА СПОРЕД ТИПА ИМ.  
 C - ПОЗВОЛЯВА ПРЕИМЕНУВАНЕ НА ФАЙЛЛОВЕ.  
 F - ИЗЧИТА ФАЙЛЛОВЕТЕ СЕКТОР ПО СЕКТОР И КОРИГИРА  
 ОБЕКТУАЛНИ ГРЕШКИ В КАТАЛОГА.  
 L - ИЗВЕЖДА СПИСЪКА ПИСТА/СЕКТОР НА ИЗБРАНИЯ /B/  
 ФАЙЛ.

G - ПОЗВОЛЯВА РАБОТА С ОТДЕЛЕН ФАЙЛ - ЛОГИЧЕСКИ  
 РЕЖИМ НА РАБОТА.  
 - ИЗВЕЖДА ПОСЛЕДОВАТЕЛНО ВСИЧКИ ФАЙЛЛОВЕ ОТ КАТА-  
 ЛОГА. <RETURN> ОЗНАЧАВА ПРИЕМАНЕ НА ПРЕДЛОЖЕНИЯ  
 ФАЙЛ, А <ИНТЕРВАЛ> - ОТКАЗ.

<ИМЕ> - ЗАПОЧВА РАБОТА С ИЗБРАНИЯ ФАЙЛ.  
 <RETURN> - ПРЕИМЕНАВА ОТ РЕЖИМ "ФАЙЛ" В РЕЖИМ "ДИСКЕТА"  
 /ФИЗИЧЕСКИ/ /D-<RETURN>).

РЕДАКИРАНЕ НА СИМВОЛНАТА ИНФОРМАЦИЯ ВЪРХУ КОМАНДИЯ  
 /FED/.

МК+A - ПОЗИЦИОНИРА КУРСОРА В НАЧАЛОТО НА ПОЛЕТО.  
 МК+V - ПОЗИЦИОНИРА КУРСОРА В КРАЯ НА ДУМА.  
 МК+B - ПОЗИЦИОНИРА КУРСОРА В КРАЯ НА ПОЛЕТО.  
 МК+S - ВРЪЩА КУРСОРА ЕДИН СИМВОЛ НАЗАД.  
 МК+F - ПРЕМЕСТВА КУРСОРА ЕДИН СИМВОЛ НАПРЕД.  
 МК+I - ОСВОБОЖДАВА МЯСТО ЗА ВМЪКВАНЕ НА СИМВОЛ.  
 МК+J - ИЗТРИВА СИМВОЛА, ВЪРХУ КОЙТО СЕ НАМИРА КУРСОРА.  
 МК+Z - ИЗТРИВА РЕДА.

ПРИ ИЗБОР НА БУФЕР ВСИЧКИ БАЙТОВЕ СЪС СТОЙНОСТИ РАЗЛИЧНИ ОТ  
 ПРОЧЕТАНАТА ОРИГИНАЛНА ИНФОРМАЦИЯ СЕ ИЗВЕЖДАТ ИНВЕРСНО, ТОВА

ВАЖИ ПРИ РЕЖИМ В И РЕЖИМ N.  
 КОМАНДИТЕ НА ДИСКФЛОПДИСК МОГАТ ДА СЕ ВЪВЕЖДАТ КАКТО С  
 ЛАТИНИЦА, ТАКА И С КИРИЛИЦА.

# ПРАВЕЦ

# 8Д

ОРЛИН ВЪЛЧЕВ

Първият български домашен компютър е вече факт. Това е микрокомпютърът Правец-8Д, разработен в Базата за развятие и внедряване към Комбината по микропроцесорна техника в Правец. Както и известният вече Правец-82 домашният компютър е вграден на базата на микропроцесора 8502. Но приликите между двата компютъра свършват до тук. Веднага трябва да заявим, че програмите, предназначени за единия компютър, само в известни случаи ще се изпълняват от другия. Защо се получава така?

Всеки микрокомпютър притежава уникален интерпретатор на Бейсик. Вариантите на Бейсик (диалектите) се различават преди всичко по командите и функциите за работа в графичен режим, защото всеки компютър използва по свой начин разделителната способност на екрана. Някоя програма, дори на машинен код, не може без системни подпрограми и входно-изходни адреси. А те за всички компютри са на различно място, имат различни функции и се изпълняват по различен начин.

Тук ще се спрем само на различията в интерпретатора на Бейсик, а оттам и на отделните команди и функции на езика при двата компютъра. Така хората, които познават популярен Правец-82, ще свикнат бързо с домашния компютър. Действително, отнасящи се за Правец-82, ще бъдат означавани със (\*), а на Правец-8Д — с (D).

# ДОМАШЕН КОМПЮТЪР

1. Редактиране
  - 1.1. Придвижаване на курсора:  
(\*) < OCB >, последвано от I, J, K или M;  
(Д) клавиши със стрелки нагоре, надолу, наляво и надолу.
  - 1.2. Размер на буфер за клавиатурата:  
(\*) 256 байта;  
(Д) 60 байта.
  - 1.3. Въвеждане (изтриване) в (на) буфера от клавиатурата:  
(\*) MK-U (CTRL-U) за въвеждане, MK-X (CTRL-H) за изтриване;  
(Д) MK-A (CTRL-A) за въвеждане, клавиш DEL (<=>) за изтриване.
  - 1.4. Изтриване на група програмни редове:  
(\*) DEL N-M;  
(Д) няма.
  - 1.5. Спиране на въведеното върку екрана:  
(\*) MK-C (CTRL-S);  
(Д) клавиш за интервал
  - 1.8. Изчистване на екрана:  
(\*) HOME или OCB-Y (ESC-C);  
(Д) CLS или MK-L (CTRL-L).
  - 1.7. Изтриване на реда, където е курсорът:  
(\*) OCB-E (ESC-E) или CALL — 868 — изтрива до края на реда;  
(Д) MK-I (CTRL-N) — изтрива целия ред.
  - 1.8. Въвеждане на шестнайсетични цифри:  
(\*) няма;  
(Д) предшестват се от # (например # FF).
  - 1.9. Преобразуване на десетични числа в шестнайсетични:  
(\*) няма;  
(Д) функция HEX\$ — HEX\$ (дес. число).
  - 1.10. Фиксиране на режим латиница:  
(\*) жълт клавиш лат;  
(Д) MK-T (CTRL-T).
  - 1.11. Въвеждане на управляващи символы:  
(\*) едновременно натискане на MK (CTRL) и друг клавиш в режим картица;  
(Д) едновременно натискане на MK (CTRL) и друг клавиш в кой да е режим.
  - 1.12. Скриване на курсора:  
(\*) няма бърза начин;  
(Д) MK-Y (CTRL-Q).
  - 1.13. Скриване на въведената от клавиатурата текст:  
(\*) няма бърза начин;  
(Д) MK-C (CTRL-S).
  - 1.14. Зауков сигнал при натискане на клавишите:  
(\*) няма бърза начин;  
(Д) MK-F (CTRL-F).
2. Въвеждане на коментари в програмата:

- (\*) REM;  
(Д) REM и апостроф (').
3. Операции над цели програми
  - 3.1. Пълнота на запис върху касета:  
(\*) около 1500 бода (бит/сек);  
(Д) по избор 300 или 2400 бода.
  - 3.2. Прочитане на програми от касета:  
(\*) LOAD  
(Д) CLOAD *име*.
  - 3.3. Запис на програми на касета:  
(\*) SAVE;  
(Д) CSAVE *име*.
  - 3.4. Отпечатване на програми:  
(\*) PR#1 и след това LIST;  
(Д) LLIST.
  4. Логически оператори
    - 4.1. ИСТИНА:  
(\*) няма, стойност 1;  
(Д) TRUE, стойност -1 — побитово отрицание на 0 — 00000000 → 11111111 (FF), което е -1 а допълнителен код.
    - 4.2. НЕИСТИНА:  
(\*) няма, стойност 0;  
(Д) FALSE, стойност 0.
  5. Аритметични функции
    - 5.1. Природен логаритъм:  
(\*) функция LOG;  
(Д) функция LN.
    - 5.2. Десетичен логаритъм:  
(\*) няма;  
(Д) функция LOG.
    - 5.3. Системна променлива със стойност ПИ:  
(\*) няма;  
(Д) PI; стойност 3,14159265.
  6. Управление на изпълнението
    - 6.1. Оператор IF. След THEN може да има оператори, които ще се изпълнят, ако условието след IF е удовлетворено:  
(\*) IF ... THEN ...  
(Д) IF ... THEN ... ELSE ...  
Ако условието след IF не е удовлетворено, ще се изпълнят операторите след ELSE.
    - 6.2. Цикъл REPEAT/UNTIL. Цикълът ще се изпълнява, докато не бъде удовлетворено условието след UNTIL:  
(\*) няма;  
(Д) REPEAT, UNTIL, PULL — премахва адрес на връщане от стека за REPEAT/UNTIL.
    - 6.3. Потребителска обработка на грешки:  
(\*) ONERR GOTO ... , RESUME;  
(Д) няма.
    7. Помощни оператори
      - 7.1. Прочитане на цяло число без знак, записано в два байта:  
(\*) ZEEK(A)+256\*PEEK(A+1);  
(Д) DEEK(A).
      - 7.2. Запис на цяло число (N) без знак в два байта:  
(\*) M=INT (N/256),

- N = N-M\*256,  
POKE A, N,  
POKE A+1, M;  
(Д) DOKE A, N.
- 7.3. Поставяне на долна и горна граница за Бейсик-програми:  
(\*) LOMEM: ... ,  
HIMEM: ... ;  
(Д) HIMEM ...
- 7.4. Включване и изключване на режим на трасировка (проследяване) на изпълнението:  
(\*) TRACE,  
NOTRACE;  
(Д) TROIN,  
TROFF.
- 7.5. Използуване на наметка, заета от графичната страница:  
(\*) няма команда;  
(Д) GRAB — заема на наметка, RELEASE — освобождаване на наметка.
8. Входно-изходни операции
  - 8.1. Отпечатване:  
(\*) PR#1,  
PRINT ... ;  
(Д) LPRINT ...
  - 8.2. Четене на натиснат клавиш без чакане:  
(\*) X=PEEK (-16384)-128,  
POKE - 16368,0,  
KEY\$=CHR\$(X);  
(Д) KEY\$.
  - 8.3. Позиционни операции на знаците при въвеждане върху екрана:  
(\*) HTAB (X),  
VTAB (Y);  
(Д) PRINT@X, Y ... ;  
PLOT X, Y ...
  - 8.4. Атрибути за въвеждане на екрана:  
(\*) INVERSE,  
FLASH,  
NORMAL;  
(Д) Атрибутите се записват в една позиция от текстовия екран и вадят до края на реда или до нова промяна (всеки ред съдържа в първата позиция атрибут за цвета на фона или картата и във втората — за цвета на знака или молива). Атрибутите се задават по два основни начина:  
— чрез ESC (OCB), следван от буква @ се счита за буква);  
— чрез директно въвеждане на ASCII кода на атрибута в текстовата позиция.  
От клавиатурата кодът ESC се въвежда след натискане на клавиша ESC (OCB), а от програмата се отпечатва кодът на ESC (CHR\$(27)) и след това буквата. Например PRINT CHR\$(27) "A".  
ASCII кодът на атрибутите се въвежда най-лесно с команда PLOT. Горният пример ще изглежда така:  
PLOT X, Y, 1.  
С тази команда атрибутът ще бъде записан на текстовата позиция с координати X и Y.



Следват различните атрибути и начините на извеждането им.

Обяснение	ESC кодове	ASCII кодове
черен молив	@	0
червен молив	A	1
зелен молив	B	2
жълт молив	C	3
син молив	D	4
лилава молив	E	5
светлосин молив	F	6
бял молив	G	7
единична височина, стандартен набор	H	8
единична височина, втори набор	I	9
двойна височина, станд. набор	J	10
двойна височина, втори набор	K	11
единична височина, мигащ, станд. набор	L	12
единична височина, мигащ, втори набор	M	13
двойна височина, мигащ, станд. набор	N	14
двойна височина, мигащ, втори набор	O	15
черна хартия	P	16
червена хартия	Q	17
зелена хартия	R	18
жълта хартия	S	19
синя хартия	T	20
лилава хартия	U	21
светлосиня хартия	V	22
бяла хартия	W	23

9. Графика  
Разделителната способност на Промет-8Д е следната:  
режим TEXT — 28 реда по 40 знака;

режим LORES 0 — 28 реда по 40 знака — стандартен набор символи;  
режим LORES 1 — 28 реда по 40 знака — втори набор символи;  
режим HIREX — 200 реда по 240 знака.

В режим „висока разделителна способност“ има само една графична страница на адрес #A000.

9.1. Установяване на цвят:  
(\*) COLOR=... — ниска разделителна способност;  
HCOLOR=... — висока разделителна способност;

(J) PAPER... — цвят на хартията (0 до 7) и в двата режима (виж цифрите на атрибутите за молив при PLOT),

INK... — цвят на молива (от 0 до 7) и в двата режима.

9.2. Чертае в режим на висока разделителна способност:

(\*) HPLOT,  
SHLOAD,  
DRAW,  
XDRAW,  
SCALE,  
ROT;

(J) в списъка, който следва всяка команда, има наследен операнд със следното значение:

F=0 — избира цвета на фона (от PAPER).

F=1 — избира цвета на молива (от INK),

F=2 — обръща цветовете (за фона тога от INK и обратно),

F=3 — не извежда върху екрана.

CURSET X, Y, F — установява курсора на точка с координати X, Y.  
CURMOV X, Y, F — премества курсора с X точки по абсцисата и Y точки по ординатата.

CIRCLE R, F — чертае окръжност с център текущата точка и радиус R.

DRAW X, Y, F — чертае линия от текущата точка до точка, чийто координати се получават от координатите на текущата точка с нарастване X и Y.

CHAR N, S, F — извежда знак с ASCII код N, чийто горен ляв край е в текущата точка. Ако S=0, знакът е от стандартния набор; при S=1, знакът е от втория (псевдографичния) набор.  
Организацията на графичния екран е следната.

Има 200 реда с по 240 точки. Всяка група от по 6 точки има общ атрибут. Така на един ред има 40 групи по 6 точки.

FILL A, B, N — запълва A реда по B групи със стойност N. Обикновено N е атрибут. По този начин правятелна област от екрана се запълва лесно с желан цвят.

POINT (X, Y) — позволява да се провери какъв е цветът на точката с координати X, Y. Ако стойността е 0, цветът ѝ отговаря на цвета от PAPER; при стойност 1 — на цвета от LNK.

PATTERN N — задава типа на линията, извеждана от DRAW и CIRCLE. Стойността на N е от 0 до 255 (от #00 до #FF). Нормално се извежда непрекъсната линия N=255 — #FF → 11111111. При маска 10101010 (#AA или 170) ще се изведе нунтирана линия.

На 22 април в присъствието на кандидат-члена на Политбюро на ЦК на БКП, зам. председател на Министерския съвет и председател на Съвета за духовно развитие Георги Йорданов бе учреден клуб "Компютър" към НДК "Людмила Живкова". Клубът ще стане център за популяризиране и пропагандиране на развитието на компютризацията в България. Към него ще бъдат изградени няколко постоянно действащи секции. Предвижда се ежемесечните сборки на клуба да се предават по Българската телевизия.

В клубния съвет са представени всички държавни и обществени организации, научни звена, имащи отношение към създаването на компютърната техника и програмната индустрия и нас, към нейното разпространение и подготовка на кадрите за работа с нея.

Списание "Компютър за вас" е съучредител на клуба, член на неговия съвет и активно ще участва в дейността му.

На добър час

клуб "Компютър" към  
НДК "ЛЮДМИЛА ЖИВКОВА".

## 10-20 ТРИМЕСЕЧЕН ОТЧЕТ НА ВЪЛНЕНИЯТА

Драги състезатели, изминаха три месеца, откакто обявихме постоянноредяващия суперминиконкурс „10—20“, и вече се събраха над 100 двуредови програми. Не ви съобщаваме точния им брой, защото те ще станат два пъти повече, докато списанието влезе в печатницата, а никой не може да предскаже колко ще бъдат, когато то попадне в ръцете ви. Но това не пречи да споделим мисли за същността на конкурса и да дадем няколко напътствия, за да продължим успешно и по-нататъшната си съвместна работа.

Както сте усетили, това наглед забавно състезание не е самоцелно, а се стреми да подтикне читателите към дълбоко усвояване на законите на информатиката и към нестандартно мислене. Затова при класирането сме дали предпочитание на програмите, в които има истинско творчество и отминаваме тривиалните предложения. Всъщност, както и очаквахме, конкурсът действува като моментален тестер за равнище в информатиката. Уж два реда, а веднага излявяват майсторството в програмирането.

Може би не е излишно да декларираме в явен вид нашето отношение към плагиатите. Ние посрещаме всички еднакво: вярваме абсолютно на всеки автор и публикуваме неговата програма така, както е изпратена (ако се наложи, внасяме корекции при очевидни и незначителни пропуски). Това е ключът на защитата срещу преписвачите: нас някой може и да успее да ни излъже, но 20 000 читатели — никога.

Затова, ако откриете, че някоя от отличените програми е заимствувана от публикация или може да се намери някоя фирмена дискета, пишете ни. Тогава ще обявим името на „автора“ с черна точка и нека той да му бере срама. (Между другото същото е отношението ни и към всички други автори, които публикуват в „Компютър за вас“.) Колкото до наградата — бъдете спокойни. Няма да му я дадем, щом не заслужава!

След тези кратки бележки, можем да обявим как върви класирането дотук.

### ВРЕМЕННО КЛАСИРАНЕ:

АНТОН ХЛЕБАРОВ	— ПЪРВА НАГРАДА ЗА ДЕКЕМВРИ
ПАВЕЛ ПЕЕВ	— ПЪРВА НАГРАДА ЗА ЯНУАРИ 1986 Г.
КАМЕН КАНЕВ	— ПЪРВА НАГРАДА ЗА ФЕВРУАРИ 1986 Г.
ВЛАДИМИР КРИЧОВ	— ЕДНА ДИСКЕТА
НИКОЛАЙ ХРИСТОВ	— ЕДНА ДИСКЕТА

### А сега да продължим.

В този брой ще съведем един нов елемент в състезанието. Освен програмата на инж. Веселин Бончев, получила първа награда, публикуваме една задача от математика Камен Канев. Тя представлява разгърната на 17 реда графична програма, но може да стане двуредова с нетривиални и не особено леки преобразувания. Нека видим колко души ще успеят да преодолеят барьерата. За тази допълнителна задача наградата е морална — ще публикуваме само пълния списък на програмистите, които са успели да осъществят същата идея на два реда.

И така, конкурсът продължава. За ИВСД „Авангард“ напълнихме първата дискета „Антология 10—20“, том 01. Пожелаваме на всички читатели успех в състезанието — и техните програми да бъдат увековечени в следващите токове.

Предложеният разгърнат вид на програмата ясно отразява алгоритъма и е удобен за експерименти. За да бъде приведен във вид на „двуредовка“ обаче, са необходими малки, но нетривиални изменения, които са по силите на всеки, който добре познава тънкостите на езика Бейсик за ПК Праец-82.

**ЗАБЕЛЕЖКА.** Имайте предвид, че изискването за двуредова програма е съществено, тъй като то наклочва възможността за използването на оператора IF.

### ПРОГРАМА „КАМЕННИ КРИСТАЛИ“

```

10 DIM A(300), B(300)
15 DIM R(300), X(30), Y(30)
20 P = RND (- 1) *
  9 + .2B32
30 HCOLOR= 3
40 FDR Z = 45 TO 5
  STEP - 13
50 FOR N = 3 TO 7
60 J = 0
  70 A(O) = 140
  80 S(O) = 80
  90 R(O) = 40
  100 FOR I = 0 TO N
  110 X(I) = COS (I *
    P / N)
  120 Y(I) = SIN (I *
    P / N)
  130 NEXT I
  140 HGR
  150 A = A(J)
  160 B = B(J)
  170 R = R(J)
  180 J = J - 1
  190 HPLLOT A + X(O)
    * R, B + Y(O)
    * R
  200 FOR I = 1 TO N
  210 X = A + X(I) * R
  220 Y = B + Y(I) * R
  230 HPLLOT TO X, Y
  240 J = J + 1
  250 A(J) = X
  260 B(J) = Y
  270 R(J) = R / 2
  280 IF R < Z THEN J
    = J - 1
  290 NEXT I
  300 IF J > = 0 THEN
    150
  310 NEXT N
  320 NEXT Z
  330 P = RND (1) * 9
    + .2B32
  340 GOTO 30

```



## КОПИРАЦИ ПРОГРАМИ И АВТОМАТИ

На запад често пъти делят програмистите на две основни групи. Едините пишат програми и с различни прийоми защитават труда си от неправомерно присвояване, а другите пък разработват копията на програми, които да се справят със защитите. Твърдят се, че и двете групи са еднакво талантливи и работливи и за това съревнованието помежду им става все по-ожесточено. Стигнало се дотам, че някои автори смятат като добро постижение, ако броят на продадените оригинални програми достигне 10% от всички, които се предлагат на пазара.

Колкото по-съгласна е защитата, толкова по-мощни стават и копирачните програми. Нещо повече, производителите на някои от тях (Локсмит например) твърдят, че програмите им са толкова гъвкави, че ще могат да се справят и със защити, които още не са създадени.

Тъй като конкуренцията не отрича и производителите на копиращи програми, фирмата „Сентръл нойт софтуер“ пусна на пазара нова версия COPY II PLUS 5. 3. на известната си програма, която в допълнение на вече популярното BIT COPY съдържа и SEKTOR COPY. Най-интересното обаче е, че тя предлага и режим на автоматично копиране, т. е. в два текстови файла е записана информация и алгоритми за копиране на около 150 програми продукта. Предвидени са възможности за обновяване и допълване на тази ин-



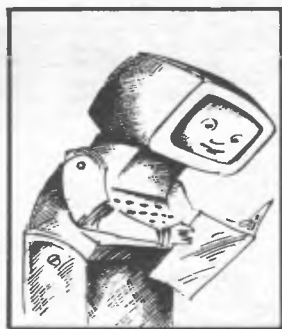
формация, а и самата фирма (само за клиентите, които законно са си купили фирмената дискета) на всеки три месеца обновява информацията с алгоритмите, необходими за преодоляване на защитата на новопоявили се на пазара програмни продукти. Трябва да се воюва за честта на фирмата и за доверието на клиентите.

Нов момент в битката за софтуерните пазари е появяването на копиращи автомати, за чието обслужване не е нужна никаква квалификация. Достигнато е в двата процента да се поставят оригиналната и празната дискета и натисне единственото копче на панела. Всичко останало е грижа на автоматa, който на всеки 40—60 секунди прави на едно висококачествено копие, на което, отгоре на всичко, поставя и собствена защита! Неговите производители твърдят, че автоматът им не се спира пред нито от съществуващите защитни системи. Това твърдение според нас е малко преувеличено, но факт е, че този нов член на компютърното семейство вече се продава на пазара и вероятно всъщност в противникови лагер.

## КОМПИЛАТОР ЗА СТРУКТУРНО ПРОГРАМИРАНЕ НА БЕЙСИК

Фирмата „Майкрософт“ е разработила един компилатор Куинбейк, който повишава скоростта при изпълняване на програми на Бейсик и позволява да се използват методите на структурното програмиране. Компилаторът е предназначен за персонални компютър PC на фирмата IBM и ускорява изпълнението на програмата от 3 до 10 пъти в сравнение със случаете, когато се използва стандартен интерпретатор на Бейсик. Въведени са и подобрения в самия Бейсик. Например номерът на реда на програмата е заменен с цифрово-буквено означение, което облекчава използването на съответните оператори. Големите блокове от програмата вече могат да се разбиват на подпрограми, които могат да се компилират независимо, да се отделят от основната програма и след това отново да се обединяват.

## ЧЕТАЩ АВТОМАТ



Създадено е устройство за въвеждане на документални данни в персоналните компютри. То четете със скорост 180 думи и минути документи, напечатани най-разпространените шрифтове за пишещи машини. Размерите му са 122 на 272 на 406 милиметра, а масата — 1,18 килограма.

## НОВ НОСИТЕЛ НА ИНФОРМАЦИЯ

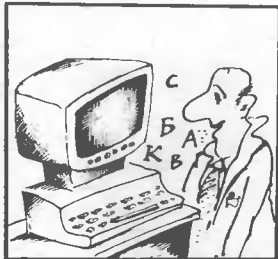
Той прилича на дебела магнитна картичка, но не се използва за разплащане, а за запазване на информация. Новата картичка открива широки хоризонти при разпространяването на научни, технически, икономически и игрови програми за персонални компютри. Тя е резултат от тригодишната работа на японската фирма „Мицубуши пластик индъстри“ и изготвена по метода на леене с подналягане от акрилугодиестирол. В нея е поместено еднокристално запаметено устройство с капацитет един Мбайт. Новият носител на информация с успех ще конкурира касетите и дискетите. Засега проблемът е в масовизирането на необходимото за целта четящо устройство.

## КОМПЮТЪР ПРЕВОДАЧ

Фирмата „Тошиба“ е произвела компютър, който превежда текстове от английски на японски с 90-процентна точност. 32-битовият микрокомпютър UX700 превежда по 5000 думи на час, което е три пъти повече от възможностите на професионален преводач. По време на превода компютърът изобразява върху екрана едновременно английския текст и японския му превод. При думи с повече значения той предлага няколко варианта. Паметта му може да съхранява 130 000 думи, от които 30 000 с общо предназначение, 50 000 специални термини и 50 000, които могат да се въвеждат по избор.

## РЕЧЕВ АДАПТОР

Новосъздаденият речев адаптор позволява на персоналните компютри РС/XT да анализират и редактират речени съобщения. За да може базовият модел с оперативна памет 256 Кбайта да работи с адаптор, в него са монтирани три допълнителни платки. Програмното осигуряване позволява осем интерактивни режима на работа. На всеки режим съответствува меню от команди, които се възвеждат със съответния клавиш.



## СЛЪНЧЕВАТА КЪЩА

Джек Търнър от Кеймбридж не е изключвал четири години персонални си компютър APPLE//e. Чрез него той управлява слънчевата отоплителна инсталация на къщата си в непрекъснат режим. Компютърът следи показанията на различните температурни датчици, монтирани в стаята, навън и в тръбопроводите и резервоарите на слънчевата инсталация. Търнър е електроинженер и сам е съставил програмата на Бейсик за управление на цялата система.

При нормални условия водата от централния резервоар циркулира през слънчевите колектори на покрива с площ 45 м<sup>2</sup>. Програмата следи температурата на водата на входа и на изхода на колектора.

Щом температурата на водата в централния резервоар падне под минималната (27°С), програмата включва топлинна помпа, която продължава да изсмуква топлината от хладната вода в резервоара. Ако

## СВРЪХБЪРЗ ПРИНТЕР

Става дума за принтер от определен тип — матричен. Основната му характеристика е високото бързодействие — 400 знака на секунда при тестове с типографско качество и графична изобразяемост в режим на висока разделителна способност. В печатащата глава на принтера има 18 игли и това осигурява разделителна способност 3,7 точки на милиметър. В графичен режим принтерът четрае със скорост 38 сантиметра и секунда.

## ВРЪЗКА МЕЖДУ МИКРО — И МАКРОЕИМ

Разработен е пакет приложни програми за включване на микрокомпютри към големи ЕИМ, като между тях се организира канал за обмен на данни. С пакета може да се осигури достъп практически до всяка файлова система и база от данни на голяма ЕИМ, производство на IBM. Той е с открита архитектура, която му позволява да се обръща към бази от данни от различен тип.

и тези запаси се свършат, се включва нафтов котел. От съображения за пожаробезопасност инженерът все пак е оставил обикновен термостат да дублира управлението на котела — ако компютърът се повреди.

Финансовата страна на тази пилотна по същности инсталация е трудно да се съпостави с нашите условия поради различните цени на горивата и елементите. Като база за сравнение може да се приеме срокът за нейното изплащане от икономия на електроенергия и горнина. По данните на Търнър този срок е от порядъка на 15 години, което показва, че слънчевото отопление все още не е прекрачило прага на масовото използване. Итересно е, че Търнър е проектирал системата така, че да може да ползува компютъра си. За времето, когато го изключва, слънчевата система продължава да работи в режим, който е „запомнила“. След като си свърши работата, Търнър зарежда отново програмата и управлението продължава в режим на реално време.

## ЕФЕКТЕН END

Предложената по-долу малка програма е от тези, без които може, но си струва да я включите в някоя своя програма, за да получите ефектен завършек, когато искате да излезете от нея. При подаване на съответната команда курсорът отива и горния десен ъгъл на екрана и започва да се движат скокообразно към долния ляв със съответен музикален съпровод. Щом го достигне, се изпълнява операторът END.

Преди да включите тази подпрограма към вашата програма, проверете дали адресното пространство \$303—\$315 (771—789) няма да се използва. На тази (трета) страница от папета са свободни за малки машинни програми клетките с адреси до \$3CF(975).

```

1000 FOR I = 771
      TO 789
1010 READ Z
1020 POKE I,Z: NEXT

1030 DATA 173,
      48,192,136,2
      08,4,198,1,2
      40,8,202,208
      ,246,166,0,7
      6,3,3,96
1040 FOR V = 1 TO
      23
1050 H = 25 - V: A
      = 128 * V +
      H - (984 * INT
      ((V - 1) / 8
      )) + 895
1060 B = A + 1: X =
      PEEK (A): Y =
      PEEK (B)
1070 POKE A,Z21:
      POKE B,96
1080 POKE 0,10 *
      V: POKE 1,9:
      CALL 771
1090 FOR J = 1 TO
      25: NEXT : POKE
      A,X
1100 POKE B,Y: NEXT
      : VTAB 23: END
    
```

# СЪДЪРЖАНИЕ

## СУПЕРМИНИ - КОНКУРС „10—20“

```
10 DATA 169,46,162,3,141,242,3,
143,243,96,111,291,76,0,0
20 FOR I = 768 TO 783: READ N$: POKE
I,N: NEXT I: CALL 768

0 REM РАЗГРЪД 7200:"СФОРНИИ"06
+ВАЛЕНТИН ДИМИТРОВ
10 DIM F(158):E(158):HGR2:=MCOLOR-
3:FOR Z=0 TO 6:28 STEP .0
48: P = 118*PI = 140 + 30 *
SIN (Z)*E(P) = 96 + 30 * COS
(Z): NEXT Z
20 FOR P = 1 TO 158: HPL0T F(P),
E(P): NEXT P

0 REM СЪБИЯ 1715:КВ:"ИМАДОСТА"
91,405,81,47,2,1,2,0,1,3
11:АСЕН ТОДОРОВ
10 FOR I = 896 TO 914: READ J: POKE
I,J: NEXT J: POKE 1010,138: POKE
1011,3: POKE 1012,186
20 DATA 32,27,253,201,195,144,2,
75,52,76,149,128,133,56,169,
3,133,57,76,105,258
```

### ДОПУСНАТА ГРЕШКА

В брой 2—3 в статията „След RESET-RUN“ са допуснати две печатни грешки, за което молим да бъдем извинени.

Вместо „RESET-сектор“ да се чете „RESET-вектор“ и вместо „при регистриране на монитър“ — „през регистриране на Монитър“.

## КОМПЮТЪР ЗА ВАС

Издаване на ЦК на ДКМС

### СПИСА РЕДАКЦИЯ „ОРБИТА“

Главен редактор  
г-р ДИМИТЪР ПЕЕВ 88-51-68

1000 София СОФИЯ  
БУЛ. „ТОЛБУХИН“ № 51 А  
ТЕЛ. 87-78-04

Приемни часове от 14 до 16 ч.

НЕПУБЛИКУВАНИ РЪКОПИСИ И ПРОГРАМИ НЕ СЕ ВРЪЩАТ.

РЕДАКЦИОНЕН СЪВЕТ: чл.-кор. Ангел Ангелов, проф. Анаел Писарев, ст.н.с. к.т.н. инж. Александър Александров, академик Благовест Сенгов, Веселин Спиридонов, доц. Димитър Шишко, инж. Иван Маранозов, инж. Пенчо Сираков, чл. кор. Петър Кендеров, н.с. к.т.н. инж. Пламен Вацков, Рашко Анавацинов

XIII конгрес на БКП	
НЕВИДИМАТА ДИПЛОМА, Николай Дюлгерев . . . . .	2
компас	
КОМПЮТЪРЪТ В ОБРАЗОВАНИЕТО, Недялко Тодоров	4
школа 10—18	
ЗИМНИ ПРАЗНИЦИ НА ИНФОРМАТИКАТА, Павел	
Азълов . . . . .	6
теория	
ЕКСПЕРТНИ СИСТЕМИ, Станислав Димов . . . . .	7
софтуер	
РЕШЕНИЯ НА ЗАДАЧИТЕ, Павел Азълов . . . . .	8
речник	
ЕЗИКОВИ БЕЛЕЖКИ, Димитър П. Шишков . . . . .	10
запозвайте се	
СИСТЕМИЗОТ, Веселин Вълков . . . . .	14
за вашия справочник	
КОДОВЕ ЗА ПРАВЕЦ 82, Георги Мирчев . . . . .	16
практика	
МОНТИРАНЕ НА DRAM ПЛАТКА . . . . .	18
ПОВЕЧЕ ЗА ОПЕРАТОРА IF, Симеон Николов . . . . .	19
РАЗШИРЯВАНЕ НА ПАМЕТТА, Петър Петров . . . . .	20
самоучител	
АСЕМБЛЕР И МАШИНЕН ЕЗИК, Орлин Вълчев и Борис	
Захариев . . . . .	22
отговори	
ДИСКФИКСЕР . . . . .	25
домашен компютър	
ПРАВЕЦ 82 . . . . .	26
състезателни програми	
10—20-ТРИМЕСЕЧЕН ОТЧЕТ НА ВЪЛНЕНИЯТА . . . . .	29
КАМЕННИ КРИСТАЛИ, Камен Канев . . . . .	29
панорама . . . . .	30

ЗАМ-ГЛАВЕН РЕДАКТОР  
И ЗАВ. СПИСАНИЕТО  
инж. Георги Балански 87-09-14

ОТГОВОРЕН СЕКРЕТАР  
инж. Борис Ачков 80-23-18

ДЕЖУРЕН РЕДАКТОР  
Кънчо Кожухаров

ДИЗАЙНЕР  
Васил Пенгев

КОРЕКТОР  
Бистра Ботева

ТЕХНИЧЕСКИ РЕДАКТОР  
Люба Калкачиева

Предадено за печат  
15 март 1986 г.

Подписано за печат  
6 мхи 1986 г. В6 г.

Печатни коли 4

Формат 60/90/8

Тираж 20 000

Цена 0,60 лв.  
Годишен абонамент 7,20 лв.  
\*\*\*\*\*  
ДП "А. Благоев"  
София, ул. "Ракитин" 2  
Телефон 46-31  
\*\*\*\*\*

(38-39)	(56-57)	KSWL KSWH	Формират входния регистър на куплюните на компютъра. RST , 0 МК-K и IN #0 установяват в тези клетки адрес \$FD1B (подпрограма за въвеждане от клавиатура - KEYIN). S МК-K и IN #S установяват в тези клетки адрес \$CS00, където S е номерът на съединителя (KEYBOARD INPUT SWITCH; L - младши байт, H - старши байт).
(3A-3B)	(58-59)	PCL PCH	Работна област, където се съхранява съдържанието на програмния брояч при обработка на прекъсванията (PROGRAM COUNTER; L - младши байт, H - старши байт).
(3C-3D)	(60-61)	A1L A1H	Работна област за параметри на подпрограми на Монитора (L - младши байт, H - старши байт).
(3E-3F)	(62-63)	A2L A2H	Работна област за параметри на подпрограми на Монитора (L - младши байт, H - старши байт).
(40-41)	(64-65)	A3L A3H	Работна област за параметри на подпрограми на монитора (L - младши байт, H - старши байт).
(42-43)	(66-67)	A4L A4H	Работна област за параметри на подпрограми на Монитора (L - младши байт, H - старши байт).
(44-45)	(68-69)	A5L A5H	Работна област за параметри на подпрограми на Монитора (L - младши байт, H - старши байт).
45	69	ACC	Съхранява съдържанието на акумулатора A при изпълнение на инструкцията BRK (\$00) - (ACCUMULATOR).
46	70	XREG	Съхранява съдържанието на индексен регистър X при изпълнение на инструкцията BRK (X REGISTER).
47	71	YREG	Съхранява съдържанието на индексен регистър Y при изпълнение на инструкцията BRK.
48	72	STATUS	Съхранява съдържанието на регистъра на състоянието при изпълнение на инструкцията BRK.
49	73	SPNT	Съхранява съдържанието на указателя на стека при изпълнение на инструкцията BRK (STACK POINTER).

СЛЕДВА

СТУДЕНТСКИ  
КЛУБ  
"КОМПЮТЪР"  
РАЙОН "ХРИСТО БОТЕВ"



СТУДЕНТСКИ  
КЛУБ  
"КОМПЮТЪР"  
РАЙОН "ХРИСТО БОТЕВ"

