

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

3 1989



КУВТ-86 на колесах

Одна из важных задач, стоящих перед школой,— предоставление одинаковых возможностей в получении образования учащимся столиц и маленьких городов и деревень. Обеспечение уроков информатики — особая проблема. Для полноценных занятий необходим компьютерный класс, а техники не хватает.



В Марийской АССР решили воспользоваться идеей создания компьютерного класса на колесах. КУВТ-86 разместили в серийном автобусе марки ЛИАЗ 677. Работу по заказу Министерства просвещения республики выполнил временный творческий коллектив НТО под руководством А. В. Куканова, председателя областного правления НТОРЭС им. А. С. Попова. Авторы проекта, они же и исполнители, успешно справились с поставленной задачей.

По подсчетам экономистов, работа коллектива обошлась республике втрое дешевле, чем стоило бы выполнение заказа традиционным путем через государственное предприятие. Более того, класс был изготовлен менее чем за месяц и принят экспертной комиссией с оценкой «отлично». Теперь школьники даже самых отдаленных школ республики работают с компьютерами. Удивительно, что за полгода ни одна машина не сломалась.

Передвижной компьютерный класс был продемонстрирован на школе-семинаре «Экспертные системы и ПРОЛОГ в учебном процессе» в Йошкар-Оле и вызвал большой интерес участников.



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

OldPC.su

7002

музей компьютеров

Содержание

Общие вопросы

- Логико-психологические основы использования компьютерных учебных средств в процессе обучения 3
Белошапка В., Лесневский А. Основы информационного моделирования 17

Методика обучения

- Гейн А., Линецкий Е., Сапир М., Шолохович В. Информатика: модели, алгоритмы и исполнители 25
Каймин В., Кугель Л., Кузницкий Е., Точенная И. Компьютерная грамотность в примерах и задачах 34
Глаголева Н., Юерман Н. Основные понятия языка Рапира 39
Поливаный И. Исполнение Бейсик-программ 44
Белов Б., Белова А. Активизация работы учащихся при изучении табличных величин и массивов 52

КВТ

- Гриценко А. Система подготовки текстов для КУВТ-86 55
Дементеев В. «Агат» рисует 59
Караваев А. Электронный коммутатор видеосигнала для дисплейного класса 62
Зинюк В., Баранаскас Э. ПМК на уроке труда 67
Картинка — на принтере 68
Подробнее о «Микроше» 70
Новые возможности БК 71
Модем для ПЭВМ — реальность 73

Педагогический опыт

- Штернберг Л. Уроки с программируемыми микроалькуляторами 75
Ускова О. Школьная информатика глазами первокурсников 79
Абдукадыров А., Источникова М., Вайнштейн Е. Компьютерный кроссворд в учебных целях 80

Внеклассная работа

Крылова Е., Петров М. III Московская олимпиада по информатике 82

Молодежная инициатива

93

Точка зрения

Дименштейн Р., Яковлев А. Информатика или компьютерное дело? 105

ЭВМ в народном хозяйстве

Исаева Н., Мироносецкий Н., Парфенова Л., Щеглов Ю. Компьютер и хозрасчет 108

Зарубежный опыт

Тихонов В. Можно ли выиграть... знания? 114

Нам пишут

Нужен ли школе совместитель? 120

Мы верим! 121

Информатика и русский язык 122

Информация

Семинар на ВДНХ 123

Международная встреча в Ленинграде 124

Два слова о съезде 125

Книги

127

Обложка Э. Бажилина

В оформлении номера принимали участие: Э. Бажилин, Н. Горбунова, С. Расторгуев

Главный редактор
академик

В. А. МЕЛЬНИКОВ

Редакционная
коллегия

И. Н. АНТИПОВ

В. Н. АФАНАСЬЕВ

И. М. БОБКО

Г. В. ГОДЖЕЛЛО

С. А. ЖДАНОВ

Б. В. ЛОМОВ

Ю. В. ЛУИЗО

(зам. главного
редактора)

Н. Г. МЕЛЬДИАНОВ

И. С. ОРЕШКОВ

О. К. ПАВЛОВА

А. Ю. УВАРОВ

А. И. ФУРСЕНКО

В. О. ХОРОШИЛОВ

К. В. ШЕХОВЦЕВ

(редактор отдела)

Редактор отдела *А. Кравцова*

Научный редактор *Т. Драгньш*

Зав. редакцией *Н. Игнатова*

Художественный редактор *Л. Розанова*

Корректор *О. Пурлова*

Сдано в набор 21.03.89. Подписано в печать 27.04.89. А 03574. Формат 70×100/16.
Бумага офсетная № 1. Печать офсетная. Усл. печ. л. 10,40. Усл.-кр. отг. 42,88.
Уч.-изд. л. 13,49. Тираж 83 650 экз. Заказ 722. Цена 60 коп.

Издательство «Педагогика» Академии педагогических наук СССР и Государственного
комитета СССР по делам издательств, полиграфии и книжной торговли

Адрес для переписки: 107005, Москва, Лефортовский пер., 8.

Адрес редакции: Студенческая ул. 37.

Телефон редакции: 249-97-77

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат
Государственного комитета СССР по делам издательств, полиграфии и книжной
торговли
142300, г. Чехов Московской обл.

© «Педагогика», «Информатика и образование», 1989

Логико-психологические основы использования компьютерных учебных средств в процессе обучения

Проект психологической концепции проектирования новых технологий обучения и развития детей. Основные разработчики: В. В. Рубцов (науч. рук., докт. психол. наук), В. Н. Каптелинин, В. А. Львовский, В. К. Мульдаров, Л. Ю. Невуева, Н. И. Поливанова, И. М. Улановская. Научный консультант — действительный член АПН СССР В. В. Давыдов.

Современные тенденции в исследовании и разработке новых информационных технологий обучения

Февральский (1988 г.) Пленум ЦК КПСС наметил широкую программу различных мероприятий, направленных на коренное улучшение работы советской школы. Одной из основных задач этой программы является оснащение школьных учреждений современными средствами вычислительной техники, которые призваны способствовать повышению качества учебно-воспитательного процесса.

Научно обоснованное применение ЭВМ в практике школьного образования, разработка перспектив и прогнозов внедрения новых технологий обучения требуют проведения фундаментальных и прикладных психолого-педагогических исследований, предвещающих проникновение новых технологий в массовую школу.

Анализ целого ряда отечественных и международных проектов, опыт использования компьютеров в образовании

свидетельствуют о том, что процесс компьютеризации становится важнейшим рычагом внедрения достижений научно-технической революции в социальную практику. Новые технологии обучения порождают новые формы учения и специфическое учебное содержание. Последнее приводит к появлению новых учебных предметов, интегральных междисциплинарных компьютерных курсов, новых подходов к организации обучения и самому процессу формирования знаний, умений, действий учащихся, новых средств оценки эффективности обучения. Актуальной и принципиально значимой с этой точки зрения представляется проблема организации целостного учебно-воспитательного процесса, ориентированного на использование новых технологий обучения и развития детей. В настоящее время можно указать на следующие основные тенденции в ее решении, имеющие значение не только для советской системы народного образования, но и важное международное значение:

1. Понимание того, что проблема будущего образования, основанного на использовании новых информационных технологий, не может быть решена только за счет развития техники, ибо компьютеры сами по себе не определяют реальной среды и культуры обучения. Необходимо научное обоснование педагогических технологий но-

3

вого типа, разрабатываемых на базе средств ЭВТ и обеспечивающих развитие детей, их творческой активности и введение инноваций в процесс учебной деятельности.

2. Формирование двух основных и наиболее перспективных подходов к решению проблемы использования компьютеров в обучении. Первый связан с проектированием и компьютерной реализацией предметно-ориентированных учебных сред, обеспечивающих развернутое моделирование содержания объектов усвоения и создание интегральных учебных предметов (НРБ, США, Франция, Япония), второй — с созданием на основе этих сред моделей совместной и индивидуальной учебной деятельности, опирающихся на процессы коммуникации и широкое взаимодействие учителя и учащихся, самих учащихся (СССР, США, Англия, Франция, Финляндия).

3. Наряду с разработкой новых педагогических технологий и их внедрением в образование получают распространение системы контроля за влиянием компьютерного обучения на психическое и умственное развитие детей. Создаваемые в этом направлении методы психодиагностики и психолого-педагогической коррекции являются частью соответствующих педагогических технологий.

Представленный проект психологической концепции использования компьютерных учебных средств в процессе обучения, реализующий в определенной степени указанные тенденции психолого-педагогических исследований, разработан коллективом сотрудников лаборатории психологии компьютерного обучения в соответствии с направлением 6 Комплексной программы АПН СССР — АН СССР «ЭВМ в школе» и по заданию Временного научно-исследовательского коллектива «Школа» Госкомитета СССР по народному образованию. В основу концепции положен деятельностный подход к обучению, раскрывающий своеобразие компьютера как средства организации и развития учебной деятельности. В рамках этого подхода в концепции представлена точка зрения, согласно которой эффектив-

ное применение компьютерных учебных средств и соответственно новых технологий обучения внутренне связано с изменением содержания образования, выраженным в появлении целостных интегральных областей знания, создании интегральных учебных предметов, что позволяет преодолеть эмпиризм узкотематического проектирования школьных дисциплин путем передачи в обучении «...опыта различных форм и видов деятельности, опыта эмоционально-ценностного отношения к миру, опыта общения и т. д.» (Концепция общего среднего образования: Проект. М., 1988. Разд. «Содержание, формы и методы обучения». С. 17).

1. Деятельностный подход к обучению и проблема создания компьютерных учебных средств

1.1. Роль машин в человеческой деятельности. Компьютеризация обучения является одной из форм интенсивного проникновения современных человеко-машинных систем во все сферы общественной жизни. Согласно теории таких систем, их ведущим компонентом является человеческая деятельность, а машинный компонент выступает как орудийное средство ее эффективной реализации. Человеческая деятельность имеет много разных видов и форм, генетически исходной основой которых является трудовая деятельность. Ее историческое развитие породило другие виды деятельности, например игровую, учебную, научную и пр. Все виды деятельности при различии своего конкретного содержания имеют общую структуру, включающую следующие основные составляющие:

- 1) потребности и мотивы;
- 2) задачи;
- 3) действия;
- 4) операции.

Если действия человека соответствуют целям какой-либо деятельности, то операции, входящие в те или иные действия, соответствуют условиям достижения этих целей. Когда в процесс выполнения человеческого действия включается такое средство, как машина, то человек, осуществляя целепо-

лагание, передает машине операторную реализацию своего действия. При этом соотношение действий и операций подчиняется такой закономерности, что потеря каким-либо действием своей цели превращает его в операцию, а приобретение некоторой операцией цели превращает ее в соответствующее действие. Эти взаимопереходы находят свое отражение в процессе построения человеко-машинных систем.

В человеко-машинной системе, соответствующей конкретному виду деятельности, человек, руководствуясь определенными потребностями и мотивами, ставит перед собой задачи, а операторную часть действия, направленного на решение этих задач, выполняет посредством машины. Иными словами, при функционировании такой системы цель деятельности определяется человеком, а достижение самой цели, т. е. получение некоторого реального продукта, осуществляется машиной.

Но любая человеко-машинная система эффективна только при достигнутой согласованности своих компонентов, при правильном «вписывании» машинного звена в целостную деятельность человека, потребности, мотивы, цели и действия которого определяют в конечном счете производительное функционирование всей этой системы.

Условия достижения цели, с которыми неразрывно связана операторная часть человеческого действия и, следовательно, использование его орудийно-машинных средств, могут быть самыми разнообразными и относиться, например, к качествам требуемого продукта, к скорости его получения, к физическим и психическим возможностям действующего человека, к изменчивости самих этих условий. Человеко-машинные системы создавались тогда, когда человек по тем или иным причинам не мог выполнить операторную часть своих действий, которую вместо него могла осуществить машина. Традиционные машины выполняли по преимуществу операции человека, связанные с такими относительно постоянными и устойчивыми условиями достижения цели, как физические возможности человека, требования к скорости созда-

ния внешнего продукта и к его качеству и т. п. (однако машины, ориентированные на такие условия, не могли осуществлять операции, которые предполагали учет, например, быстрой смены самих условий выполнения человеческих действий).

Положение решительно изменилось, когда современная электроника и вычислительная техника привели к возникновению информатики, вторгнувшейся в выполнение человеком умственных действий. Эти действия, как правило, требуют ориентации человека на огромное количество быстро меняющихся и неповторяющихся условий, многие из которых связаны с особенностями психики самого человека, с особенностями функционирования ее продуктов (например, таких, которые сейчас называют интеллектуальными системами). Вычислительная машина (ВМ) как раз и может фиксировать в своем устройстве определенную часть этих условий и тем самым выполнять операции соответствующих умственных действий человека. При этом любая ВМ остается лишь средством выполнения операторной части этих действий, задачи для которых всегда составляет человек.

1.2. Человеко-машинные системы учебной деятельности. Учебная деятельность в соответствии со своим специфическим содержанием состоит из учебных потребностей и мотивов, учебных задач, учебных действий и учебных операций. Содержанием этой деятельности является овладение человеком теоретически обобщенными по содержанию знаниями и умениями, опираясь на которые можно успешно решать различные конкретно-практические вопросы. Потребности и мотивы учебной деятельности связаны с устремлением человека овладеть именно такими знаниями и умениями еще до того, как он столкнется с практическими вопросами, чтобы быть подготовленным к их правильному решению. Специфика учебных задач состоит в том, чтобы при их решении человек мог раскрыть процесс происхождения содержания теоретических знаний и умений и овладеть обобщенными способами дей-

ствий в конкретных практических ситуациях.

Состав учебных действий, которые человек выполняет при решении учебной задачи, достаточно сложен, и их необходимо специально перечислить:

преобразование условий учебной задачи с целью обнаружения общего отношения, лежащего в основе изучаемой системы теоретических знаний;

моделирование выделенного отношения в графической и знаковой форме; преобразование модели отношения с целью изучения ее общих свойств; выделение и решение конкретно-практических вопросов на основе общего способа;

контроль за выполнением предыдущих действий;

6 оценка усвоения общего способа решения данной учебной задачи.

При выполнении человеком этих учебных действий он овладевает определенной системой теоретических знаний и обобщенным способом решения некоторого класса конкретных практических задач.

Учебные операции, связанные с условиями выполнения учебных умственных действий, весьма многообразны и изменчивы, поскольку эти условия соотносятся с содержанием различных учебных предметов (математикой, языком, физикой, историей и т. д.). Если учесть взаимопревращаемость учебных действий и операций, то определить сколько-нибудь однозначно состав операций очень трудно.

Именно с этим обстоятельством были связаны серьезные проблемы создания человеко-машинных систем в области учебной деятельности, хотя и в ней издавна использовались элементарные и достаточно сложные технические средства (например, счетные устройства при овладении арифметикой, различные аудио-визуальные средства и т. д.). В 20-е гг. нашего века в США были предприняты попытки создания настоящих учебных машин, облегчающих человеку усвоение знаний, а в 40—50-е гг. возникло программированное обучение, осуществление которого в определенной мере было связано с необходимостью применения вычислительных машин,

т. е. современных человеко-машинных обучающих систем.

Опыт программированного обучения с использованием вычислительной техники до сих пор сохраняет научное и практическое значение. Однако следует иметь в виду, что оно возникло на основе теоретических установок бихевиоризма, ограниченно представлявшего процесс обучения, не раскрывавшего подлинного содержания и структуры учебной деятельности и не сумевшего определить настоящее место в ней машинных компонентов. Бихевиористическая теория обучения абсолютизировала момент упражнения в целостном процессе усвоения человеком знаний и умений. Упражнение можно в той или иной степени соотнести с такими учебными действиями, как решение конкретно-практических вопросов и результативный контроль, но оно мало связано с другими важными учебными действиями, в частности с действием моделирования.

На основе этой теории с применением вычислительной «начинки» были созданы обучающие машины тренажерного типа, реализующие процесс упражнения и проверки усвоения знаний и умений по разным предметам. Но есть основания полагать, что при этом были механизированы не столько операции учебных действий самого человека, овладевающего знаниями и умениями, сколько некоторые операции учительского труда (что, конечно, само по себе существенно). В итоге при введении учащихся в новые области знания и организации учебной деятельности такие системы не преодолевают, а, наоборот, усугубляют противоречия, свойственные традиционному, безмашинному обучению. Схема процесса передачи знаний, принятая в данной технологии обучения, существенно затрудняет освоение учащимися основ рефлексивно-теоретического мышления, поскольку в рамках этой схемы обучение приобретает характер своеобразного «программирования» действий и операций учащихся.

Применение компьютера в учебном процессе является не только предпосылкой совершенствования обучения,

но и потенциальным источником ряда негативных последствий. В частности, использование компьютера может послужить причиной распада целостной системы деятельности «учитель — класс» на отдельные элементы типа «ученик — компьютер», контролируемые учителем. Поэтому специальной задачей проектирования компьютерных технологий обучения является поиск способов организации общения и сотрудничества учителя и учащихся, самих учащихся. Разработка такого рода способов должна осуществляться по следующим направлениям:

создание условий учебного сотрудничества между школьниками и учителем во время их работы, опосредованной применением компьютера;

организация коллективных «проектов», требующих взаимодействия группы учащихся с компьютером и групп учащихся между собой;

определение оптимального соотношения компьютерных и безкомпьютерных форм обучения.

При этом целостность организации учебного процесса в условиях компьютерного обучения достигается за счет трех основных аспектов:

управления познавательной активностью отдельных школьников;

управления учебной деятельностью как системой «учитель — компьютер — ученик»;

управления взаимодействиями и сотрудничеством учителя и учащихся, самих учащихся.

Опора на психологические закономерности каждого из этих видов управления является необходимым условием разработки человеко-машинных систем учебного назначения.

2. Новые информационные технологии обучения — основа развития современного образования

2.1. Интеграции учебных предметов в условиях использования компьютерных учебных средств. Анализ мировых тенденций показывает, что период информатизации общества во всех развитых странах закономерно привел к формулированию новых требований,

предъявляемых к системе образования и к самим принципам организации процесса трансляции культурно-исторического опыта подрастающему поколению.

Появление в культуре нового интеллектуального учебно-познавательного средства с неограниченной сферой применения практически повсеместно выявило неадекватность существующих традиционных форм обучения, его целей, содержания и способов организации. В связи с этим постепенно сложилась и к настоящему моменту стала доминировать более конструктивная и гуманитарная по своей сути ориентация в разработке и прогнозе возможностей использования компьютерной техники в обучении, которая выразилась в смещении акцентов с самих машин на собственно субъекта («учитель — ученик — ученики») учебной деятельности как ключевого фактора проектируемых технологий обучения.

Целенаправленные попытки осмысления полифункционального учебного применения компьютера, определение его реального влияния на процессы обучения и воспитания детей, на закономерности формирования у них знаний, умений и навыков привели к возникновению междисциплинарной области исследований на стыке психологии, педагогики и техники по широкому кругу проблем когнитологии, информатики, лингвистики, социологии, искусственного интеллекта и, самое главное, психологии развития. В этой новой социокультурной ситуации задача перестройки сферы образования и разработки новых технологий обучения (с применением или без применения машин) может решаться только за счет вклада всех перечисленных исследовательских областей, и прежде всего тех, которые ориентированы на человека.

Основные особенности компьютера как инструмента человеческой деятельности и принципиально нового учебного средства состоят в следующем:

1. Компьютер обеспечивает доступ к практически неограниченному объему информации и ее аналитической обработке. Резкое количественное увеличение объема потенциально доступ-

ной информации и скорости ее получения приводит к качественному скачку — возникновению феномена «непосредственной включенности» человека в информационную культуру общества.

2. Компьютер представляет собой универсальное средство познавательно-исследовательской деятельности человека.

3. Компьютер обеспечивает новую — активную — форму фиксации продуктов психической деятельности. Все предшествующие средства объективации психической деятельности создавали лишь предпосылки для выполняемых самим человеком преобразований, таких, как анализ отдельных аспектов содержания знания, сличение с уже имеющимися данными, дополнение данных новой информацией из некоторой области, использование для организации практических действий и др. Компьютер впервые позволяет полноценно выполнять и частично автоматизировать эти преобразования.

4. Компьютер является вторым по значимости, после традиционной письменности, знаковым орудием, с помощью которого возможен оперативный обмен информацией по содержанию выполняемой деятельности. Таким образом, появляется принципиально новая область использования человеком языка и знаково-символических средств деятельности в целом.

5. Компьютер в ряду других орудий и инструментов человеческой деятельности обладает особым коммуникативным свойством, отличающим его от любого иного средства способностью «вступать в конструктивно-содержательный диалог» с пользователем и составлять с ним единую функциональную предметно-ориентированную среду. Специфика такого деятельностного способа организации «мира объектов» состоит в том, что компьютер не просто и не только усиливает интеллектуальные возможности человека, воздействует на его память, эмоции, мотивы и интересы, но изменяет и перестраивает саму структуру познавательной, а затем и производственной деятельности человека.

Известно, что активное и самостоя-

тельное построение или изменение деятельности доступно человеку в условиях, когда он имеет возможность целенаправленно и смыслообразно обращаться к основаниям своих собственных действий, осуществлять их планирование и рефлексии, трансформировать и самостоятельно конструировать предметное содержание, с которым работает. Компьютер, опосредствующий познавательную деятельность, обеспечивает такую возможность благодаря тому, что любое ответное действие или воздействие пользователя может быть индексировано, обозначено схемой или моделью, сохранено, возвращено и фиксировано для анализа, оценки и контроля. Кроме того, каждое действие может быть свернуто до операции и, наоборот, развернуто или перестроено в соответствии с намерениями и возможностями пользователя и условиями динамического пространственно-временного представления объектов. В этом смысле компьютер является настолько эффективным и тонким средством объективации всех компонентов учебной деятельности, что ему трудно подобрать аналог в практике обучения. При этом компьютер как носитель знаково-символического универсума деятельности по самой своей природе ориентирован на интегральный, заведомо идеализованный способ представления объектов, моделирование которых невозможно вне целостного полисемантического развертывания, адекватно отражающего содержание реальной действительности.

Отсюда следует, по крайней мере, два вывода. Первый из них касается возможности интегрального представления содержания предметной среды, которая создается с помощью компьютера и в которой органично соединяются конкретные структуры знаний (гуманитарных и естественнонаучных), полноценно представляющие содержание соответствующих объектов усвоения. Второй вывод связан с возникновением благодаря интегральной предметной среде наиболее эффективных условий для формирования обобщенных способов действия, обуславливающих развитие у детей полноценных форм реф-

лексивно-теоретического мышления.

Сама интеграция может осуществляться при этом по двум взаимосвязанным направлениям. Во-первых, путем объединения материала из нескольких предметных областей, позволяющего выделить и сформулировать наиболее общие принципы и закономерности своего разветвления. Во-вторых, такая интеграция может осуществляться по способам действий учащегося с предметным содержанием. При этом формулируется система обобщенных стратегий поиска решения задач, структурирования информации, постановки проблем и т. д., одновременно обеспечивается как познавательное развитие учащихся, так и эффективное усвоение учебного материала. Отметим, что и интеграция по содержанию объектов усвоения, и интеграция по способам преобразования объектов усвоения должны быть основаны на логико-психологическом анализе предметных или операциональных структур знаний, задающих по существу новое содержание образования.

Вместе с тем использование компьютера в системе деятельности, нацеленной на продуцирование только «символических продуктов» (т. е. в сугубо «квазипредметной» деятельности), создает опасность отрыва знания от предметно-практической основы его происхождения. Решение этой проблемы обеспечивается за счет способов использования самих учебных средств. Эта сфера взаимодополняющего, комплексного применения всего спектра выработанных в культуре средств организации учебной деятельности и представления содержания знаний. В этой связи можно говорить о комплексном использовании различных информационно-технологических средств для создания полифункциональной предметной среды — объединения компьютера, видео, телевизора, кино, а также о взаимодействии этой полифункциональной среды с собственно предметными «компьютерными формами обучения».

2.2. Функции учебного компьютера как средства моделирования деятельности. Как показывает современная педагогическая практика, использование

компьютера в учебном процессе направлено на решение по преимуществу следующих четырех типов задач.

Во-первых, компьютер используется в качестве вспомогательного средства для более эффективного решения уже имеющейся системы дидактических задач. Содержанием объекта усвоения в компьютерной обучающей программе этого типа является справочная информация, инструкции, вычислительные операции, демонстрации и др. Примером использования компьютера в этой функции являются экспертные системы.

Во-вторых, компьютер может быть средством, на которое возлагается решение отдельных дидактических задач при сохранении общей структуры, целей и задач безмашинного обучения. При этом само учебное содержание не закладывается в компьютер (он выполняет функции контролера, тренажера и т. п.). Эта функция компьютера широко представлена в разветвленных диалоговых системах, моделирующих деятельность учителя.

В-третьих, использование компьютера позволяет ставить и решать новые дидактические задачи, не решаемые традиционным путем. Характерными являются, например, компьютерные программы по имитации эксперимента. В этих программах в качестве объекта усвоения выступают: а) внешние параметры того или иного процесса; б) закономерности, которые не доступны наблюдению в естественных условиях; в) связи имитируемых явлений с теми параметрами, которые автоматически задаются программой; г) поиск параметров, оптимизирующих протекание имитируемого процесса, и т. д. Исследуются также возможности сопряжения реального и компьютерного учебного эксперимента.

И наконец, в-четвертых, компьютер может использоваться в качестве средства, моделирующего содержание объектов усвоения путем его конструирования. При этом реализуются принципиально новые стратегии обучения. Характерным примером этого направления разработок в области компьютеризации являются так называемые «компьютерные обучающие среды», представляю-

щие модели осваиваемых областей знания (см., например, С. Пейперт, США; Б. Сендов, НРБ; и др.). Функционально-ориентированная учебная среда создает предпосылки для порождения целей и планов действия, что открывает ученику возможность выступить в роли субъекта своей деятельности. Вместе с тем проблемы и ограничения этой технологии обучения, основанной на принципе конструктивизма в психологии (в соответствии с концепцией интеллектуального развития Ж. Пиаже), обусловлены спонтанным характером деятельности учащихся, связанным с тем, что эта деятельность осуществляется через систему игровых действий пользователя с содержанием предметной среды. Поэтому актуальной в разработке данных технологий обучения остается проблема преобразования игровой мотивации деятельности детей в полноценно осуществляемую и внутренне мотивированную учебную деятельность.

Мы полагаем, что в основу проектирования новых компьютерных развивающих технологий обучения должен быть положен принцип компьютерного моделирования деятельности, в которой воссоздаются условия для поиска, отображения в моделях и анализа содержания сущностных характеристик объекта усвоения. Компьютер как специфическое учебное средство реализует несколько основополагающих функций, а именно выступает в качестве средства:

а) моделирования предметного содержания объектов усвоения;

б) моделирования соответствующих обобщенных способов действия;

в) моделирования взаимодействий и организации совместной деятельности (типа «обучаемый — группа учащихся», «ученик — ученик», «учитель — ученик»);

г) реализации адекватных структуре совместной деятельности и содержанию объектов усвоения форм контроля и оценки действий учащихся.

Во взаимосвязи указанных функций компьютерные системы обучения представляют собой предметно и коммуникативно направленную, рефлексивно управляемую учебную среду, организованную как целостная система дея-

тельности и включающую контроль в качестве необходимого условия полноценного функционирования.

Попытки экспериментальной реализации указанных функций компьютера позволяют говорить о ряде важнейших психологических особенностей использования компьютерных средств на различных этапах обучения. Так, построение с помощью компьютера моделей содержания объектов усвоения дает возможность учащимся самостоятельно ставить и решать новые проблемы, что позволяет учителю руководить продвижением учащихся, управлять переходом от одних форм организации учебной деятельности к другим, осуществлять управление движением учащихся в логике учебного материала. Возможность опосредованной оценки своих действий способствует развитию у школьников основ рефлексивно-теоретического отношения к действительности, умения самостоятельно организовывать, планировать и корректировать свою учебную работу. Наконец, необходимо отметить эффективность использования компьютерных средств в процессе контрольно-оценочной деятельности школьников.

Вместе с тем при оценке роли моделей и выделении критериев их применения для создания компьютерных учебных сред необходимо руководствоваться специальными требованиями, определяющими своеобразие новых технологий как системы организации и управления полноценной учебной деятельности.

3. Психологические требования к проектированию компьютерных учебных средств и систем обучения

Учитывая общий смысл перечисленных теоретических положений, можно определить требования к проектированию компьютерных учебных средств и систем обучения.

Первая группа требований связана с отбором и характером рекомендуемого к освоению материала, определяющего предметно-содержательную основу проектируемой учебной среды. Выполнение этих требований позволяет обес-

печить в процессе обучения единство логико-психологической основы конкретного учебного предмета или его раздела (системы соответствующих понятий и схем как единиц усваиваемого материала) и тех учебных действий и операций, посредством которых с помощью компьютера учащиеся могут успешно выявить происхождение теоретически обобщенных знаний, способы графического и знакового моделирования и изучения их содержания.

Требования этой группы включают: выявление и описание области знания, задающей конкретный объект изучения через определение его исходных элементов и их отношений; определение типов связей элементов (аксиоматика), характеризующих объект как некоторую систему;

определение логико-предметного контекста существования объекта усвоения при помощи описания адекватных условий его происхождения, преобразования и конструирования;

определение типа знаковых средств, обеспечивающих реализацию всех видов связей на экране дисплея; выбор динамического типа моделей, позволяющих изучать свойства исходного отношения в «чистом виде», изменять, трансформировать и конструировать объект изучения (предметные, графические и знаково-символические модели);

определение и описание системы учебных и конкретно-практических задач как условий освоения объекта через его моделирование; определение концептуального содержания учебных задач и средств освоения понятий;

проблематизацию концептуального содержания задач таким образом, чтобы, с одной стороны, исключались или существенно ограничивались эмпирические стратегии поиска («пробы и ошибки»), а с другой — обеспечивались доступность и достижимость продуцирования средств, способов и приемов решения («вскрываемость» всеобщего отношения).

Удовлетворяющая названным требованиям компьютерная предметная среда представляет собой сложную многоуровневую систему, включающую:

исчерпывающий набор элементов «на-

турального» уровня предметности, позволяющий воспроизвести в наглядно-образном плане существенные свойства объекта изучаемой области знания;

исчерпывающий набор «естественных» для данной предметности действий для преобразования предметного содержания с целью открытия основного отношения, а также выделения границ, в которых найденные связи и отношения являются адекватными содержанию;

исчерпывающий набор элементов разного уровня обобщения, «естественных» для изучаемой области знания, позволяющих учащемуся строить модель объекта усвоения на том уровне обобщения, который ему доступен на данном этапе овладения материалом;

«естественные» для конструирования способы оперирования элементами модельного уровня предметной среды; возможности опосредованного воздействия на «натуральные» элементы предметной среды и способы их взаимодействия (например, числовое задание параметров, их перемещение, построение таблиц, графическое отображение изменения параметров, построение формул и т. д.);

простые способы синтеза последовательностей элементарных операций с натуральными и модельными элементами среды в более сложные команды, выполняющие целостные преобразовательные действия (выработка собственного языка), управление элементами среды;

простые способы перехода от одного уровня предметной среды к другому и возможность совмещения работы с элементами натурального и модельного уровней предметности.

Вторая группа требований касается операционального обеспечения учебных действий учащихся, а также режимов обмена действиями между пользователем и ВМ. Эта группа требований включает:

планирование действий и операций учащихся в соответствии с логико-психологическим анализом деятельности и структурой объекта усвоения;

соблюдение принципа развертывания учебной деятельности от освоения ис-

ходных действий и операций к овладению их сложной совокупностью через предъявление системы учебных и конкретно-практических задач в определенной и заранее обусловленной последовательности;

обеспечение возможностей программирования собственных действий, свертывания систем операций и действий, возврата к исходным действиям, замены действий, ограничения действий партнеров и т. д.;

обеспечение возможностей произвольного выбора типа действия, темпа его выполнения, уровня сложности задачи, вида модели, при помощи которой воспроизводится, трансформируется и конструируется объект усвоения;

обеспечение выбора между дискретным и непрерывным режимами трансформации объектов на экране и возможности комбинирования автоматизированного и автоматического режимов управления действиями с объектом в зависимости от достигаемого уровня анализа объекта и этапа решения учебной задачи;

специальное планирование поисковых и пробующих действий при освоении инструкций к выполнению учебных или конкретно-практических задач и обращение к базе данных с целью обеспечения условий для формирования обобщенных способов действий, мотивационных и смыслообразующих компонентов деятельности;

обеспечение возможности оперативно представлять содержание и результат действия. Для создания такой ситуации следует «снимать» заданность частных целей, характерных для этапа решения конкретно-практических задач, и дать возможность учащимся осуществлять широкий спектр преобразований объекта и рефлексивных действий (различных фиксаций, схем, обозначений и т. д.);

обеспечение операционального представления решения задач и возможности фиксировать при помощи вычислительных средств каждую отдельную операцию (пооперациональный контроль). Компьютер может использоваться здесь как эффективное средство протоколирования хода решения, которое позволяет воспроизвести его повторно в авто-

матическом режиме, и, кроме того, как диагностическое средство оценки уровня сформированности как отдельных составляющих учебной деятельности, так и обучающего (результаты освоения предметного содержания, умений и навыков) и развивающего (сформированность специальных компонентов мышления, психических функций) эффектов. Такая диагностическая подсистема может быть непосредственно встроена в компьютерную систему обучения либо частично и функционально быть обособленной от нее.

Специальное требование связано с обеспечением должной связи предметных преобразований, разворачивающихся на дисплее, и их операциональной реализации на кнопочной панели пульта. Так, при выполнении преобразования желательна максимальная простота исполнительных операций, отсутствие избыточных звеньев, опосредствующих отношение дисплея и пульта. Для этого может потребоваться специальное введение промежуточных средств, связывающих предметный и операторный планы учебных действий. В этом качестве могут использоваться различные схемы исполнительских операций (нажатия на клавиши) в виде конфигураций — трафаретов или разного рода знаковых изображений, накладываемых на клавиатуру.

Конкретная реализация компьютерной обучающей программы, соответствующей перечисленным требованиям, должна опираться на адекватное вхождение учащихся в учебные ситуации, что предполагает эффективные способы взаимодействия учителя, ЭВМ и учащихся. К этим способам относятся, например, демонстрация учащимся образцов работы с компьютером, разделение действий и операций при решении учебной задачи между учащимися и их кооперация, взаимный контроль и оценка действий и операций и т. д. Поэтому третья группа требований относится к организационному обеспечению учебной среды, состоящему из: а) конфигурации системы и б) перечня способов взаимодействия учителя и учащихся (индивидуально-групповое, межгрупповое, внутригрупповое). Конфигу-

рация системы включает описание пространственно-временных характеристик учебных ситуаций (расположение участников, их число, состав группы), а также требования к компьютерной сети для обеспечения распределения действий и операций между участниками, способов обмена действиями, а также средств коммуникации и обмена.

Требования к организационному обеспечению компьютерной учебной среды включают:

проектирование учебных ситуаций как динамических общностей учителя и учащихся, самих учащихся; эти общности могут иметь различные организационные формы: живое учебное сотрудничество, предваряющее совместное компьютерное моделирование объекта, опосредованный компьютером диалог участников совместной деятельности, интерактивное взаимодействие ученика и компьютера, группы учащихся и компьютера и т. п., опосредованное вычислительным средством, групповое моделирование целостного объекта, совместное моделирование связей и законов движения, совокупностей объектов и т. д.;

обеспечение при помощи вычислительных средств: а) целенаправленной организации коммуникативных учебных взаимодействий в группах учащихся, которые приводят к анализу предметных оснований самой формы организации совместной работы и поиску новых способов взаимодействий; б) машинной фиксации процесса и результатов учебных взаимодействий; в) оценки и контроля совместных действий в ходе обучения;

осуществление на одной машине (при разделении частей клавиатуры между участниками или наличии двух клавиатур) либо на разных машинах совместной работы обучаемых с общей моделью-задачей. При этом программирование управления учебным взаимодействием должно предусматривать следующие элементы: 1) звено репрезентации (экран дисплея); 2) звено объективных операций и рефлексивно-коммуникативных операций (клавиатура терминала); 3) звено коммуникации и кооперации (процедуры согла-

сования взаимодействий). Все части контура взаимодействия должны быть взаимосвязаны и либо подчиняться управляющим воздействиям учащегося, который также должен иметь поле репрезентации (представления), либо управляться автоматизированно;

создание специфически учебного конфликта при осуществлении учащимися преобразовательных действий с целью стимулирования аналитического и рефлексивного компонентов деятельности в процессе группового обучения;

создание ограничений во взаимодействии учащихся, достигаемое специальным программированием различий в степенях свободы индивидуальных действий: от их полной независимости друг от друга до существенных ограничений взаимными влияниями.

Особая группа требований связана с обеспечением правильного контроля в процессе функционирования компьютерных систем обучения. Такие системы должны обеспечивать:

самоконтроль для учащихся;
оперативный сбор информации о процессе выполнения учебных заданий и его предварительную обработку;

представление этой информации на мониторе в соответствии с системой диагностических критериев и показателей, оформление результатов обработки в виде протоколов;

генерацию коррекционных батарей заданий для индивидуального или совместного режима выполнения.

Режим контроля и оценки должен варьироваться от полностью автоматизированного (пошагового или по конечному результату) до текущего и отсроченного. Наиболее высоким уровнем контроля, обеспечивающим функционирование системы компьютерного обучения, является контроль за организацией взаимодействий обучающего и обучаемых, когда группа участников совместно решает задачу моделирования объекта через конструирование способов взаимодействия, т. е. когда вычислительное средство осваивается как средство построения самой предметной среды и управления ею.

О перспективах организации НИР в области компьютерного обучения на принципах деятельностного подхода

Научно-исследовательские и конструкторские работы по компьютеризации обучения на основе деятельностного подхода и психологической теории учебной деятельности развернуты у нас пока очень слабо, и для этого есть свои веские причины. Они связаны прежде всего с тем, что этот подход предъявляет серьезные требования к действительному анализу содержания объектов усвоения и структуры учебных действий и учебных операций, поскольку именно на этой основе может осуществляться интеграция предметного материала и его реализация с помощью компьютерных учебных средств. У нас еще сравнительно мало сведений об учебных действиях и операциях, об их взаимопревращениях, поэтому в настоящее время трудно реализовывать стратегию создания новых технологий обучения на основе деятельностного подхода, хотя, на наш взгляд, эта стратегия является сейчас наиболее перспективной с точки зрения разработки новых методов и нового содержания образования. Так, необходимо прежде всего определить такие стороны учебных действий и операций, которые действительно нуждаются в механизации, дающей существенное повышение эффективности обучения. Отсутствие соответствующего метода в разработке компьютерных учебных средств сводит проблему компьютеризации обучения в лучшем случае к созданию тренажеров, в худшем — к использованию современной техники для организации внешних моментов традиционного обучения.

Некоторые наши данные дают основание полагать, что компьютеры крайне необходимы при выполнении учащимися двух учебных действий — при прослеживании самого процесса происхождения определенных знаний и при моделировании выделяемого исходного отношения, задающего эти знания. Уже давно и многократно были описаны внешние особенности первого учебного действия (например, при изложении особенностей знаменитого сокра-

товского или эвристического метода обучения). Однако долго не было технических средств, позволяющих учащимся «своими руками» предметно (а не в словесном плане) реконструировать внутреннее содержание процесса происхождения тех или иных понятий. На наш взгляд, этими средствами сейчас могут стать ЭВМ, правильно «вписанные» в целостную учебную деятельность. Отдельные моменты учебного моделирования также хорошо известны и используются при построении различных графических схем, но в них выражаются внешние особенности усваиваемых знаний. Их внутреннее освоение может быть смоделировано в особых компьютерных системах, причем само их функционирование может стать моделью учебных действий. Таким образом, одна из главных психологических проблем компьютеризации обучения, на разработке которой следует сосредоточить исследовательские и конструкторские усилия, связана с анализом содержания объектов усвоения и определением тех учебных операций в некоторых учебных действиях, которые наиболее эффективно могут осуществляться человеком с помощью компьютера. Результаты таких разработок могут стать действительной основой достаточно развернутой теории компьютерного обучения.

Имеющиеся сейчас у нас тенденции конструировать и производить компьютерные системы обучения без опоры на такую теорию могут, конечно, через некоторое время привести к наполнению ими наших учебных заведений. Их применение позволит несколько улучшить учебный процесс за счет, например, быстрого поступления к учащимся необходимой информации, за счет быстрого сбора и обработки сведений об имеющихся у них знаниях и умениях, за счет интенсификации упражнений в этих умениях. Но включение таких систем в уже давно сложившуюся схему учебного процесса не сможет перестроить его сколько-нибудь существенно, что, естественно, не приведет и к ожидаемым результатам в усвоении учащимися содержания тех или иных объектов.

Введение компьютеров в целостную учебную деятельность предполагает серьезный семантический анализ «знаний и умений» с точки зрения содержащихся в них свернутых умственных действий и операций — они являются внутренней основой этих «знаний и умений», которую как раз и нужно развернуть в программах работы учебных компьютеров.

Сформулируем ряд общих положений, конкретизирующих стратегию разработки и использования в обучении компьютерных учебных средств на основе деятельностного подхода.

1. Компьютерные системы обучения должны целенаправленно создаваться для включения в целостную учебную деятельность с учетом всех ее составляющих (особенно с учетом учебных действий и операций). Это позволит предотвратить деформацию и даже разрушение этой деятельности при использовании ЭВМ в процессе обучения (что нередко в настоящее время и случается). Компьютеры должны быть предназначены не только для обучения человека тем или иным знаниям и умениям, но и для организации его учебной деятельности и управлению ею.

2. Компьютерные системы обучения должны создаваться на основе предварительного анализа содержания соответствующих знаний и умений как объектов усвоения: разному содержанию должны соответствовать и разные программы компьютерного обучения. Но один учебный компьютер может удовлетворять требования различных учебных предметов.

3. Каждая программа создается применительно к усвоению содержания, представленного на языке определенных действий и операций. Это позволяет строить учебную деятельность по принципу движения мысли человека от освоения исходных действий и операций к овладению их сложной совокупностью; последнее является предпосылкой интеграции учебных предметов.

4. Компьютерные системы обучения должны соединить в себе качества динамических и семиотических (знаковых)

моделей: осваивая эти модели, человек тем самым выполняет соответствующие учебные действия и усваивает открываемое ими содержание определенной предметной области. Работая с такими системами, человек не адаптируется к ним, а действует с ними, осуществляя преобразование некоторого предметного материала, а также контролирует эти преобразования с точки зрения поставленных задач.

5. Сама по себе компьютерная система не является «учителем», не она представляет «аппарат управления», регулирующий учебный процесс, она органически входит в решение человеком систем учебных задач. При этом компьютер выступает как средство организации совместной деятельности учителя и учащихся, самих учащихся и обеспечивает следующие формы их взаимодействия:

разделение действий и операций при решении учебной задачи между разными участниками и их кооперацию; взаимный контроль и оценку действий и операций учащихся в ходе решения учебных задач определенной последовательности;

совместное моделирование задаваемых учителем схем преобразования объекта;

отображение и представление одним учащимся способа решения задачи, осуществленного другим.

6. Эти формы организации взаимодействия учащихся позволяют учителю использовать компьютер для организации учебной деятельности в системе коллективного диалога, т. е. проектировать учебные ситуации как динамически моделируемую коммуникативно-организованную среду, обеспечивающую широкое взаимодействие и сотрудничество участников деятельности.

7. Отдельные виды компьютерных систем должны использоваться для диагностики уровня сформированности отдельных составляющих учебной деятельности, а также для контроля и оценки (в том числе текстовой оценки) результатов усвоения содержания определенных знаний и умений.

8. Учебные компьютеры должны учитывать возрастные аспекты развития че-

ловека: разным возрастным периодам должны соответствовать различные способы представления содержания в компьютерных системах обучения (от квазипредметных игровых форм в младшем школьном возрасте до квазиисследовательских и исследовательских творческих форм в среднем и старшем школьном возрасте и т. д.).

9. Создание компьютерных систем обучения должно осуществляться путем развернутого изучения способов их применения в различных учебных ситуациях; соответствующие разработки и исследования должны выступать основой для изучения возможностей каждой системы и ее использования в преподавании учебных предметов.

10. Применение компьютерных систем обучения должно способствовать формированию у человека основ рефлексивно-теоретического мышления, использующего логико-математические средства для программирования и планирования человеком своих познавательных действий и анализа основания их выполнения.

Изложенные выше требования и положения были сформулированы на основе теоретических и экспериментальных разработок, посвященных анализу проблем компьютеризации обучения на деятельностной основе и проведенных в НИИ общей и педагогической психологии АПН СССР сотрудниками лаборатории психологии компьютерного обучения. В настоящее время эти положения и требования уже нашли определенную реализацию:

в разработке пробных образцов компьютерных систем обучения, основанных на моделировании содержания объектов русского языка, природоведения, физики, английского языка;

в разработке компьютерных методик, обеспечивающих организацию совместной деятельности учителя и учащихся при постановке и решении учебных задач, в том числе на основе использования компьютерных сетей;

в разработке компьютерных методик обследования уровня сформированности

у различных возрастных контингентов школьников учебных действий, а также компьютерных методик диагностики развития основ рефлексивно-теоретического мышления учащихся.

Вместе с тем необходимо углубить исследовательскую работу в этом направлении, создав для этого благоприятные условия. Опираясь на деятельностный подход в обучении и фактические экспериментальные данные, требуется построить теорию конструирования и использования компьютеров в системе целостной учебной деятельности, а затем насытить учебные заведения именно такими компьютерными системами обучения, применение которых, на наш взгляд, даст требуемый учебный эффект.

Пояснения к тексту

Бихевиоризм — направление в психологии, согласно которому предметом научного исследования являются лишь объективно наблюдаемые акты поведения, т. е. зависимости между воздействующими на организм стимулами и ответными реакциями.

Знаково-символический универсум деятельности — способы специфически информационного, т. е. полностью знаково-символического, отражения реального предметного мира с точки зрения как знания, так и умений действовать, осваивая знания и трансформируя их.

Коррекционная батарея заданий — система учебных и развивающих заданий, выбор которых осуществляется индивидуально для каждого учащегося на основе данных диагностики.

Объективация психической деятельности — выражение психической деятельности во внешних, доступных объективному наблюдению формах (движениях, речи, материальных продуктах деятельности и т. д.).

Операциональные знания — знания об оптимальных приемах и способах действий.

Полисемантическое развертывание — раскрытие сущности объектов усвоения (предметов, явлений, способов действий) в широком контексте значений.

Свертывание действий — переструктурирование действий в процессе их освоения. При этом часть промежуточных звеньев переходит на неосознаваемый уровень, часть сокращается. Таким образом осуществляется переход от действия к операции. Обратный процесс — **развертывание** — необходим для модификации сложившихся операций; используется также для контроля за степенью освоенности действия.

Основы информационного моделирования

Проект курса в стиле brain

Социальный заказ на компьютерную подготовку выпускников средних школ меняется в связи с тем, что компьютерная техника становится все более доступной, программные продукты все более эргономичными, с естественным интерфейсом, ориентированным на традиционные, «докомпьютерные» навыки человека. В силу этих причин от пользователя требуется все меньше специальных знаний и умений, связанных с использованием компьютера. Круг этих знаний и умений постепенно сужается и приобретает вполне «бытовую» окраску (почувствовав запах газа, немедленно сообщите по телефону 04; уходя из дома, отключите электроприборы; обнаружив, что система не загружается, замените дискету и т. д.). Требования к программистской подготовке пользователя снижаются. Опыт развитых стран убедительно показывает, что поголовная компьютерная грамотность, как она понимается у нас в стране, не нужна. Нужна лишь элементарная подготовка пользователя (о том, что это такое, будет сказано чуть ниже).

Необходимо наконец признать, что существующий сейчас в школе обязательный курс ОИВТ, особенно в его безмашинном варианте, явился одним из уродцев, рожденных административно-командной системой.

На самом деле умение использовать компьютер для решения практических (не программистских) задач основывается на глубоком понимании смысла звеньев основной технологической цепочки решения (объект—модель—алгоритм—программа—результат) и отношений между ними. Ключом же к умению правильно и эффективно использовать компьютер является понимание отношения «объект — модель».

Это, так сказать, прагматическая

сторона социального заказа. Если же иметь в виду его общекультурный аспект, то достоянием культурного человека сегодня является представление о методах современной науки, среди которых метод формального моделирования занимает одно из ведущих мест. Нам представляется, что соответствующий комплекс знаний, умений и навыков является фундаментальным компонентом современного среднего образования.

Если ставить цель воспитать творца, а не исполнителя и потребителя, то необходимо признать, что мы должны дать возможность учащимся узнать о методах современной науки, о научных основах современных производственных процессов. Всем известно о росте наукоемкости производства. Говоря о наукоемкой технологии, мы каждый раз подразумеваем, что в ее основе лежит некоторая формальная модель реальных процессов. Естественно предположить, что подавляющее большинство выпускников школы будут иметь дело с наукоемкой технологией. Непонимание научной основы такой технологии не оставляет никаких надежд на то, что человек будет творчески относиться к своей работе, не говоря уже о том, что такое «информационное невежество» не позволит правильно ориентироваться в социокультурных явлениях современного мира.

Сформулируем теперь определение элементарной подготовки пользователя. Это комплекс, состоящий из следующих компонентов:

практических навыков, позволяющих школьникам самостоятельно использовать общеупотребительные виды программного обеспечения (включение—выключение с соблюдением правил техники безопасности, загрузка с внешнего запоминающего устройства, просмотр оглавления (директория), вызов про-

граммы, копирование и стирание файлов);

элементарных сведений об устройстве компьютера;

элементарного знакомства с понятиями «алгоритм» и «программа».

Курс, позволяющий школьникам овладеть указанными знаниями и умениями, будем называть курсом элементарной компьютерной грамотности.

Такой курс в качестве самостоятельного предмета, а скорее — в качестве вкрапления в другой предмет, может быть включен в учебные планы средней или даже младшей ступени общеобразовательной школы.

После его изучения можно использовать компьютеры с соответствующим программным обеспечением для преподавания широкого спектра школьных предметов.

В старших же классах следует подходить к продолжению обучения информатике дифференцированно. Мы полагаем, что логика вещей требует формирования следующих специализированных курсов (в соответствии с членением hardware — software — brainware):

«hard-курс» — с акцентом на инженерно-физических принципах функционирования компьютеров и автоматизированных систем, а также их проектирования;

«soft-курс» — с акцентом на принципы и технологию программирования;

«brain-курс» — с акцентом на теоретические принципы и практические навыки построения формальных моделей и работы с ними.

Именно о курсе третьего типа пойдет речь ниже. Прежде чем привести программу, охарактеризуем его основные черты.

Цель курса — научить методу информационного моделирования и применения его в различных предметных областях.

Курс рассчитан на учащихся, владеющих основами компьютерной грамотности.

Особенностью курса является перенос акцента со средства (компьютер и его программное обеспечение) на цель (решение конкретных задач), т. е. техноло-

гическая цепочка «объект—модель—алгоритм—программа—результат» изучается во всей ее полноте с акцентом на ведущем звене «объект—модель».

Методической особенностью является широкое использование коллективных и самостоятельных форм работы учащихся.

Теоретическим ядром курса является изучение основ системно-информационного языка описания формальных моделей. Сюда входят отношение «объект—модель» в связи с представлением о формализованном языке как о моделирующей знаковой среде, концепция системы как понятия родового по отношению к понятию модель, арсенал средств описания систем в терминах элемент—свойство—отношение, а также понятие структура; концепция формального процесса и связанных с ним понятий алгоритм, информация, операция, инвариант, представление о конструировании систем, в частности алгоритмов, из элементарных составляющих (элементов, операций, команд), а также о декомпозиции систем.

Курс состоит из теоретического ядра и системы практических работ по реализации учебных проектов с использованием компьютера.

Структура курса: введение обзорно-ознакомительного характера, затем основное содержание — 4 раздела, каждый посвящен определенному типу моделей (систем). Специфика метода информационного моделирования (по отношению к конкретному типу систем) раскрывается на примерах моделирования в различных предметных областях. В заключении — явное вычленение конструкций системно-информационного языка.

Во введении дается обзор основных проблем и понятий курса, важнейшими из которых являются: информатизация как новый этап развития общества; информатика как научная основа информатизации; информация; знак; слово как единство смыслового и знакового аспектов; компьютер как средство представления и преобразования знаковых конструкций; отношения «объект—модель»; средства построения моделей; шкала языков; понятийный аппарат формального моделирования; понятие системы.

Первый раздел посвящен моделям классификационного типа (введение в когнитологию). Рассматриваются дискретные и непрерывные модели. Теоретическое содержание раздела иллюстрируется примерами автоматизированных систем: речевой ввод информации; экспертные системы; статистический анализ; информационные системы и базы данных.

Второй раздел посвящен динамическим моделям (введение в кибернетику). Теоретическое содержание раздела иллюстрируется примерами систем и процессов: кибернетические системы (техника, биология); компьютер, исполнитель; глобальное моделирование; алгоритм как формальная модель дискретного процесса; системы передачи информации.

Третий раздел посвящен логиколингвистическим моделям (введение в структурную лингвистику). Теоретическое содержание раздела иллюстрируется примерами систем: автоматический перевод; языковые средства программирования; языки представления знаний.

В заключение рассматривается история становления информатики, обсуждаются конструкции системно-информационного языка, значение метода информационного моделирования для построения компьютерных (информационных) технологий, перспективы информатизации общества.

Общеобразовательный потенциал курса повышения за счет активизации межпредметных связей, в том числе с предметами гуманитарного цикла.

Программа курса*

Введение (информатизация как новый этап познания и освоения мира человеком)

Общая характеристика феномена информатизации. Место компьютера в данном процессе (от топора до исполнителя). Информатика — научная основа информатизации.

* Теоретический материал подразделяется на основной и дополнительный. Дополнительный материал выделен курсивом. Списки практических работ даются для примера, т. е. чтобы указать их характер и направленность.

Роль информационного моделирования в основной технологической цепочке. Отношения «объект—модель». Средства построения модели как язык и язык как средство моделирования.

Проблема построения формальной (знаковой, информационной) модели. Понятийный аппарат формального моделирования: элемент—свойство—отношение. Понятие «система». Графы как средство представления формальной модели.

Знать:

содержание процесса информатизации общества и роль в нем средств вычислительной техники;

содержание понятий «информация», «знак», «система», «элемент», «свойство», «отношение».

Уметь:

описывать простейшие модели в терминах «система», «элемент», «свойство», «отношение»;

использовать графы для представления простейших моделей.

19

Классификационные модели (введение в когнитологию)

Обзор автоматизированных систем, основанных на классификационных моделях. *Классификация и описательные дисциплины.*

Общая постановка задачи классификации. Дискретные модели: древовидные структуры, иерархия, подобие (систематика в биологии, таблица Менделеева). Задача распознавания образов — непрерывная модель. *Самообучающиеся процедуры, перцептрон.*

Практические работы:

разработка структуры, реализация (на основе готовых программных средств), наполнение и изучение предметно (тематически) ориентированных справочных систем (биология, химия, физика, математика, история, география и пр.);

разработка автоматизированных процедур распознавания и классификации (математические фигуры, буквы, животные и растения и пр.).

Знать:

основные типы классификационных систем.

Уметь:

строить простейшие базы данных с предметным наполнением;

разрабатывать элементарные процедуры распознавания.

Динамические модели (введение в кибернетику)

Обзор автоматизированных систем, основанных на динамических моделях.

Основные кибернетические модели: «черный ящик» со входом и выходом, гомеостаз; обратная связь. Задача управления объектом, оптимальное управление. Конечный автомат, формальные нейронные сети. Проблемы передачи информации (формальная модель коммуникации). *Дискретность и непрерывность в моделировании.*

Практические работы:

построение и изучение модели космического полета;

построение и изучение модели экологической системы;

моделирование эволюции клеточной популяции.

Знать:

содержание понятий «гомеостаз», «управление»;

основные типы автоматизированных систем, построенных с использованием кибернетических представлений;

основные способы измерения количества информации.

Уметь:

строить простейшие кибернетические модели;

использовать кибернетические модели для простейшего анализа динамических объектов.

Логико-лингвистические модели (введение в структурную лингвистику)

Обзор автоматизированных систем, основанных на логико-лингвистических моделях.

Формальное моделирование мышления и речи. Элементы математической логики: высказывания, связки, импликация, кванторы, законы. *Интерпретация построенной модели в терминах теории множеств. Логические парадоксы, программа Гильберта, результаты Гёделя.*

Лингвистический анализ языков программирования.

Принципы структурного анализа текста. Частотный анализ как метод изучения текста.

Практические работы:

разработка языка управления исполнителем;

частотный анализ текста.

Знать:

назначение и сферы применения систем искусственного интеллекта;

основные понятия математической логики (высказывание, связка, импликация, квантор);

содержание и сферы применимости моделей естественного языка;

принципы структурного анализа текста.

Уметь:

записывать и преобразовывать простейшие высказывания средствами математической логики;

осуществлять структурный анализ текста.

Заключение

История становления основных идей и методов информатики. Три точки зрения на мир: вещественная, энергетическая, информационная. Формирование и реализация этих точек зрения как три фазы единого процесса.

Что такое информация. Информация, данные, сведения и знания в процессах коммуникации. Слово как единство смыслового и знакового аспектов. Формально-знаковый характер информации. Понятие «знак». Компьютер как средство представления и преобразования дискретно-знаковых конструкций.

Соотнесение понятий «модель» и «образ», формальное — неформальное (содержательное), языки искусства, шкала языков. Что такое информатика? Системно-информационный язык, его основные конструкции, их универсальность. *Аналогия как научный метод.*

Информационное моделирование как метод познания и как основа компьютерных технологий.

Перспективы информатизации общества.

Знать:

основные этапы становления идей и методов информатики;
сущность метода информационного моделирования;
основные конструкции системно-информационного языка;
содержание понятий «информация», «знак».

Примерный список программных средств, пригодных для использования при изучении данного курса (в обсуждении данного списка приняли участие Я. З. Гольц и В. А. Сазонов).

Введение:

ПС для конструирования и исследования схем с конкретной предметной ориентацией (электрические, логические и т. д.).

Классификационные модели:

игра «Угадай объект в иерархической классификации»;

система распознавания простых графических образов с возможностью демонстрации процесса самообучения; классификация в пространстве признаков (2—3 признака);

простой пакет программ для статистического и графического анализа данных; справочные системы по школьным предметам, выходящие по своему объему за рамки школьной программы.

Динамические модели:

компьютерно реализованные динамические модели (экология, космический полет, двигатель, телевизор, химическое производство, гибкое автоматизированное производство, система массового обслуживания);

игровые модели с возможностью задания стратегии и ее автоматического выполнения.

Логико-лингвистические модели:

учебные экспертные системы (диагностика неисправностей в некоторой схеме, система для управления исполнителем в сложной ситуации);

программное средство для обучения элементарной математической логике (высказывания, связки, кванторы, предикаты);

учебное средство для автоматизированного анализа текста;

система для распознавания типа текстовой информации.

Несколько слов о преподавании такого курса. Как видно из приведенной программы, типичная глава имеет следующую структуру: обзор—теория—практикум. Сам термин «практикум» не вполне нейтрален и ассоциируется, скорее, с практикумом по физике. Однако свой тип практических работ порождает каждый учебный предмет, точнее, каждая концепция учебного предмета. И именно тип, потому что, несмотря на все многообразие задач по математике и вопросам по литературе, тем и другим соответствует свой вполне сложившийся типаж. А как выглядит типичное задание по информатике? Ответ, который мы даем, органически связан с развиваемой нами концепцией этого предмета (см.: ИНФО. 1987. № 6; 1988. № 5) Он вкратце таков: информатика в совокупности научных дисциплин занимает весьма сбалансированную и интегрирующую позицию по отношению к основным противостояниям (например, естественнонаучное — гуманитарное), эта же сбалансированность характерна и для прикладной информатики, так что было бы правильно, если бы тип учебного задания по соответствующему учебному предмету отражал эту сбалансированность. Таким образом возникает концепция «полномерного» учебного задания по информатике, среди особенностей которой отметим следующие:

ориентация не на отдельные фрагменты и навыки (например, составление алгоритмов) в области информатики, а на полный цикл работ (постановка задачи — моделирование — алгоритмизация — компьютерная реализация — работа с готовой программой — анализ результатов — создание отчета);

задание рассчитывается на выполнение в течение длительного времени (наподобие курсового проекта в институте);

задание выполняется рабочей группой (2—4 человека), возможны консультации у преподавателя.

Опишем теперь в общих чертах начало изучения каждого из разделов (обзор соответствующих автоматизированных систем).

1. Имеется ряд задач, где профес-

сиональные представления специалиста в некоторой области оказались в достаточной степени формализуемыми, что позволило создать соответствующие экспертные системы, в частности, для диагностики неисправностей технических устройств (например, телевизоров);

диагностики болезней;

определения названий растений и т. д.

Каждая такая система создается в результате кропотливой работы системного аналитика (специалиста по формальным системам и формальному моделированию) со специалистом экстракласса в соответствующей области, создающих формальную модель данной предметной области.

Пользователем такой системы будет предметный специалист (например, механик по тепловозам) более низкого класса, не обладающий навыками, позволяющими квалифицировать ситуацию самостоятельно. Работа с системой выглядит как некоторый диалог; после ответов на поставленные системой вопросы (достаточно элементарные) ситуация будет ею квалифицирована (неисправность тепловоза, название растения) и даны дополнительные рекомендации (по устранению неисправности, использованию растения).

2. В связи с компьютеризацией меняется характер библиотечного дела. Поиск книги или статьи в каталоге — задача, в которой использование вычислительной техники уместно и эффективно. При реализации такой библиотечной системы большое значение имеет способ построения каталога, организация справочной информации, способ доступа к ней. В связи с появлением носителей информации большой емкости (лазерные диски), а также новыми средствами коммуникации (сети ЭВМ, спутниковая связь) возникает и отчасти уже реализуется возможность переноса текстовой, графической и звуковой информации на эти новые носители и предоставления тех или иных ее порций (книг, фильмов и пр.) по запросам с удаленных терминалов. Хотя здесь есть ряд нерешенных проблем (как технических, так и социальных), но в принципе это возможно и работа в

данном направлении ведется.

Аналогичные системы могут использоваться в научных исследованиях. Например, в органической химии за последние 100 лет исследовано и синтезировано такое количество соединений, что обычными способами (используя статьи, монографии, справочники) с соответствующей информацией работать очень трудно, поток же новых данных все возрастает. Первый шаг в решении этой проблемы средствами ВТ — создание системы реферативно-библиотечного типа (справки о статьях на данную тему). Однако поиск конкретной информации с помощью такой системы порождает ссылки на десятки публикаций, что сильно усложняет работу. Гораздо более эффективна в данной ситуации база данных, в которой в определенном формате хранятся данные, характеризующие каждое известное химическое соединение (а их более миллиона): названия соединения, его физические, химические, биологические и технические свойства. Работа по созданию таких баз данных, разработке их интеллектуального, программного и аппаратного обеспечения ведется.

3. Компьютерное моделирование оказалось эффективным методом исследования в таких областях, как метеорология, экология и т. д. Особенности подобных систем в том, что они являются высокоинтегрированными, т. е. в них не удастся вычленить простые части и изучать их по отдельности, без учета всей системы в целом (феномен целостности). Так, например, исчезновение (вымирание, истребление) какого-либо одного вида живых существ отражается на всем биоценозе в силу богатства взаимных связей и зависимостей.

Одна из ведущих проблем в этой области — прогнозирование, т. е. предсказание на основе компьютерного моделирования значения тех или иных параметров. Пример — прогноз погоды или же долгосрочные прогнозы изменения климата Земли в связи с возросшим (за 100 лет на 25 %) содержанием углекислого газа в атмосфере. Как известно, углекислый газ, прозрачный для видимого света, силь-

но поглощает длинноволновое излучение (тепло) Земли, создавая так называемый парниковый эффект. Большое впечатление на мировую общественность произвели результаты моделирования климатических последствий ядерной войны. Основной вывод следующий: в результате массовых ядерных взрывов (суммарно 5000 Мт) в атмосферу поднимается такое количество твердых частиц (дым, пепел, пыль), что ее прозрачность резко снижается и температура падает на 20—40 °С («ядерная зима»), причем такое похолодание должно продлиться многие месяцы.

После «крупномасштабной» демонстрации возможностей применения ЭВМ в какой-либо сфере вниманию учащихся предлагается некоторый пример, который разбирается подробнее. Во время этого обсуждения формируется предварительная постановка учебного задания. Например, при разборе использования ЭВМ в космических исследованиях может прозвучать такая предварительная постановка: расчет режима полета космического аппарата на Луну. Причем здесь же можно начать обсуждение и уточнение постановки. В конце урока учитель комплектует группу учеников (3—4 человека), выразивших желание заняться этой задачей серьезно и систематически. Группа получает эту задачу на длительный период, не меньше четверти. Вся дальнейшая работа, за исключением ее завершающего звена, о котором мы скажем ниже, идет в режиме консультаций группы и ее представителей с учителем в отрезки времени, отведенного на практикум. При этом учитель может либо сразу рекомендовать группе определенный план работ, либо делать это постепенно, следуя логике задачи. Вот один из таких планов.

Углубление познаний в данной теме предметной области. В рассматриваемом случае это физика полета космического аппарата.

Уточнение постановки задачи. Уточняем критерий успешности полета (нулевая скорость на поверхности Луны, отсутствие столкновений с небесными телами, отсутствие запредельных перегрузок и пр.), характер управляющих воздействий (тяга, рули); оговариваем имеющиеся ограничения (ресурс топлива) и т. д.

Этап формального описания (моделирования)

данной ситуации. Здесь делаются упрощающие предположения, которые позволят впоследствии перейти к алгоритмам и их реализации на учебной ВТ: естественно принять, во-первых, все те предположения, которые обычно принимаются при решении школьных задач по физике (орбиты круговые, планеты — шары и т. д.); во-вторых, ряд специфических предположений (принимаем во внимание только Землю, Солнце и Луну, орбиты считаем лежащими в одной плоскости и пр.). Затем осуществляется переход к численному описанию ситуации: введем координат, числовых параметров, после чего можно выписать основные соотношения — три силы притяжения (Солнце, Луна, Земля) на основании закона тяготения плюс сила тяги двигателя и второй закон Ньютона для космического аппарата. Не следует забывать, что масса аппарата — величина переменная, она уменьшается по мере расходования топлива. Силы, как известно, складываются векторно, поэтому ясно, как вычислить результирующую. Теперь пора сделать важное предположение, позволяющее заменить решение дифференциальных уравнений обычными школьными расчетами: время в данной модели будем считать дискретным, т. е.

$$t = t(n) = t_0 + n \cdot \Delta t.$$

Это означает, что какие-либо изменения могут происходить только в эти моменты, а между ними силы и тем самым ускорение считаем постоянными. Следует обсудить проблему выбора шага Δt , а также весь комплекс вопросов, связанный с адекватностью модели.

Возможно, в процессе построения модели возникает необходимость изучения некоторых дополнительных вопросов по информатике. В данном случае это может быть общая постановка задачи автоматического управления. Учитель может рекомендовать соответствующую литературу.

Итак, модель готова, можно начинать продуцировать алгоритмы. Здесь важно выявить три класса величин: аргументы (время старта, масса аппарата, характеристики двигателя), промежуточные величины (координаты аппарата, параметры управляющих воздействий) и результаты (мера успешности полета и пр.). Затем производится разработка некоторого алгоритма управления аппаратом и запись его теми или иными средствами.

Теперь можно переходить к реализации алгоритма в виде программы для учебного компьютера и ее отладке. При этом учитель выполняет роль консультанта.

Программа заработала. Начинается самое интересное: получение результатов для различных исходных данных, их анализ, построение гипотез, их проверка и пр.

Завершают работу оформление результатов исследования, их защита на выступлении перед всем классом, оценка работы группы.

Такое учебное задание обладает той полнотой, которая характерна для реальных работ с использованием ВТ. Даже сотни упражнений на составление алгоритмов и программ не дадут

столь полного и правильного понимания проблем информатизации, как выполнение одного такого задания. При работе, построенной по приведенной методике, учебная деятельность принимает тот творческий и исследовательский характер, который она никогда не получит при отработке конструкций языка программирования. Такое полное и сбалансированное учебное задание реально направлено на общеобразовательные задачи, стоящие перед школьным курсом информатики.

Представленный здесь проект курса ориентирован на факультативные занятия, однако его гибкая модульная структура (теоретическое ядро + практикум) создает возможность ориентации на различные контингенты учащихся и типы учебных заведений: профильное обучение в старших классах средней

школы (математика + компьютер, естествознание, гуманитарный и другие профили), высшие учебные заведения, институты усовершенствования учителей, курсы повышения квалификации и т. д. Единственное, для чего подобный курс совершенно не пригоден, — это принудительное обучение.

Авторы проекта обращаются ко всем организациям и лицам, заинтересованным в его реализации, с призывом откликнуться. В частности, мы ищем: учителей, которые согласны принять участие в экспериментах по преподаванию одного из курсов такого типа;

организации, которые могут выступить в роли спонсоров проекта.

Мы надеемся, что реализация данного проекта послужит целям перестройки системы образования на гуманистической основе.

Компьютер создает компьютер

Следя примеру С. Джобса, который в гараже вместе со своим другом собрал компьютер, переросший потом в популярную персональную ЭВМ «Макинтош», многие сегодняшние радиолюбители пытаются сделать то же самое и порой даже в худших условиях. Однако любому, кто взялся за подобное дело, вскоре становится понятно, что с помощью одного паяльника что-либо стоящее соорудить невозможно. Как невозможно построить прецизионный станок, имея только гайки М4 и рашпиль.

Для тех, кто все-таки решил себя самостоятельно разработать, собрать и отладить свой собственный компьютер, западные фирмы, производящие электронное оборудование, предлагают разнообразные инструментальные системы. В их основе обычно лежит мощный персональный компьютер с набором программ, позволяющих не только промоделировать работу еще не существующего компьютера, но и превратить идеи авто-

ра в чертежи электрической схемы и рисунки печатных плат.

Но и готовый компьютер требует еще немало труда, прежде чем его можно будет отдать в руки программистов. Отладить новую ЭВМ поможет все тот же инструментальный комплекс.

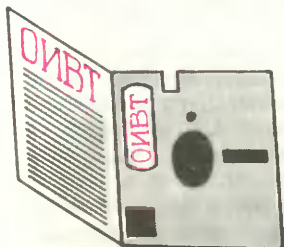
ЧТО?
МОЖЕТ
ЭВМ

Только в этом случае его необходимо пристыковать к собранному компьютеру, чтобы в работе проверить правильность выполнения всех его команд и выявить ошибки сборки и проектирования.

Чтобы не мешать работе опытной ЭВМ, компьютер-«нянь-

ка» должен успевать перерабатывать получаемую от налаживаемого компьютера информацию в том же темпе, в котором она поступает. Поэтому к инструментальным ЭВМ предъявляются особые требования по быстродействию, а фирмы, предлагающие подобные средства разработчикам компьютерной техники, в первую очередь стремятся продемонстрировать максимальную тактовую частоту, на которой может работать их инструментальный компьютер. Сейчас эта частота у наиболее популярных образцов достигает 25 мегагерц, что позволяет отлаживать разрабатываемый микропроцессор в режиме с нулевым временем ожидания (т. е. в реальном масштабе времени) с трассировкой выполняемых на нем программ.

Вторым показателем, обеспечивающим коммерческий успех подобных системам, является возможность работы с различными процессорами и ЭВМ на их основе. Подавляющее число современных инструментальных компьютеров обеспечивает эмуляцию процессоров таких фирм, как «Моторола», «Интел», «Зиллог» и «Хитачи».



А. ГЕЙН, Е. ЛИНЕЦКИЙ, М. САПИР, В. ШОЛОХОВИЧ

Информатика: модели, алгоритмы и исполнители

Данная статья — вторая в серии публикаций, посвященных методике преподавания курса ОИВТ с использованием учебника [1]. Изложение соответствует поурочному планированию курса, приведенному в [2]. В предыдущей статье начато описание методики изложения темы «Этапы решения задач с помощью ЭВМ» (§ 1, 2 гл. 1 [1]). В настоящей статье мы завершим это описание и расскажем о методике изложения темы «Алгоритмы и исполнители» (гл. 2 [1]). Мы сохраняем нумерацию глав и параграфов, принятую в [1].

§ 3. Пример математической модели

В этом параграфе на примере конкретной задачи подробно обсуждается построение математической модели. Выбор задачи продиктован следующими соображениями.

1. Задача должна занимать важное место в одном из школьных курсов. Желательно, чтобы эта задача имела прикладную направленность.

2. Это должна быть одна из тех задач, при решении которых в соответствующем курсе фактически создавалась приближенная математическая модель. Тогда при выборе упрощающих предположений можно сослаться на уже изученный школьниками материал.

3. Задача не должна быть слишком простой, чтобы соответствующий вычислительный эксперимент был достаточно содержателен.

4. Задача должна быть такой, чтобы соответствующий школьный курс предусматривал натуральный эксперимент для проверки ответа.

Это позволит сопоставить натуральный эксперимент с вычислительным.

5. Вычислительный эксперимент по этой задаче должен убеждать в предпочтительности вычислительных экспериментов перед натурными.

6. Задача не должна быть «изолированной», чтобы можно было продемонстрировать достаточно широкий класс задач, решаемых с помощью ЭВМ, и один из распространенных методов построения математических моделей.

7. Задача должна быть такой, чтобы построение математической модели делало ее решение более понятным (в частности, за счет явного разбиения решения на этапы).

8. Задача должна допускать линейный алгоритм решения, поскольку к моменту проведения вычислительного эксперимента учащиеся не будут знакомы с другими способами организации действий. В этом алгоритме должны использоваться все допустимые действия исполнителя **ВЫЧИСЛИТЕЛЬ** (см. [2]), изученные до соответствующей лабораторной работы.

9. Разбор задачи должен занимать не более двух учебных часов.

Как видно, перечисленные условия ограничивают учителя в выборе задачи весьма существенно. Тем не менее приводимая ниже задача удовлетворяет всем этим условиям (авторы будут благодарны каждому, кто предложит другую, более простую задачу).

Задача. Тело движется прямолинейно с ускорением a м/с² и начальной скоростью v м/с. Требуется определить, какой путь пройдет тело за T секунд.

В статье [2] дан подробный текст объяснений учителя в ходе построения математической модели этой задачи.

В результате изучения материала параграфа и решения задач к нему учащиеся должны уметь построить простейшую математическую модель.

Перейдем к разбору задач для самостоятельного решения из § 3.

Задача 1. Завершите составление математических моделей в задачах 1а) — 1г) § 2.

Комментарий к решениям

В задачах 1 и 3 § 2 реализованы первые два этапа построения математических моделей: высказаны упрощающие предположения и указаны исходные данные и результаты (см. [3]). Осталось записать математические соотношения, связывающие исходные данные с результатами. Это не представляет особых трудностей для учащихся, поэтому в целях экономии места мы опустим текст решений (в [4] решения приведены полностью).

Задача 2. Выполняя утреннюю зарядку, девятиклассник подошел к стене, на которой был закреплен пружинный эспандер, и оттянул его на некоторое расстояние. Какую работу совершила при этом сила натяжения пружины?

При построении для этой задачи математической модели были сделаны следующие предположения: сила натяжения пружины подчинена закону Гука; если весь промежуток движения тела разбить на большое число равных маленьких промежутков, то можно считать, что сила на каждом из них постоянна и меняется «мгновенно» в конце каждого промежутка; при неограниченном увеличении числа промежутков величина работы получается с любой точностью. Завершите построение математической модели.

Разбор задачи

Здесь надо провести аналогию с задачей о равноускоренном движении (см. [1], § 3 или [2], с. 14—15). В ней равномерно увеличивалась скорость, в этой равномерно увеличивается сила натяжения пружины. В той

3



задаче «измельчался» промежуток времени, в этой «измельчается» отрезок перемещения тела. Там отыскивалась величина пути, здесь — величина работы. После таких пояснений становится очевидным, что фактически при решении этой задачи можно полностью повторить решение задачи о равноускоренном движении, заменив «скорость» на «силу», «время» на «перемещение», «путь» на «работу». Приведем подробный текст решения.

Определим, что считать исходными данными и результатом нашей модели. Исходными данными являются путь S , коэффициент жесткости пружины k . Результатом является работа A , совершаемая силой натяжения пружины.

Математическое соотношение, которое мы должны получить, будет зависеть от количества частей, на которое разбивается промежуток движения тела.

Разобьем весь промежуток на N равных частей длины $r = S/N$. Силу, действующую на i -м промежутке ($1 \leq i \leq N$), обозначим через F_i . Сила $F_i = r \cdot i \cdot k$.

Работа, совершаемая телом, будет равна $A = F_1 \cdot r + F_2 \cdot r + F_3 \cdot r + \dots + F_N \cdot r$.

Сила натяжения F меняется от 0 на первом промежутке разбиения до $k \cdot S$ на последнем.

Последовательность $F_1, F_2, F_3, \dots, F_N$ является арифметической прогрессией с первым членом 0 и разностью $d = k \cdot r$.

Воспользуемся формулой для суммы N членов арифметической прогрессии:

$$S = ((2F_1 + d \cdot (N-1)) \cdot N / 2) \cdot r = (d \cdot (N-1) \cdot N / 2) \cdot r = k \cdot r \cdot (N-1) \cdot N / 2 \cdot r.$$

Подставляем вместо r значение S/N :

$$S = k \cdot (S/N) \cdot (N-1) \cdot N \cdot (S/N) / 2 = k \cdot S^2 \times (N-1) \cdot N / (2 \cdot N^2) = k \cdot S^2 \cdot (N-1) / (2 \cdot N).$$

Получили математическое соотношение, связывающее исходные данные и результат. Построение математической модели закончено.

Задача 3. Составьте математическую модель для решения следующей задачи.

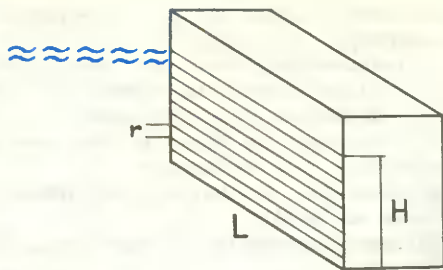
Плотина прямоугольной формы перегораживает реку. Определить силу давления воды на плотину.

Решение

Напишем предположения, лежащие в основе модели.

1. Давление на глубине x на единичную площадку равно $\delta \cdot g \cdot x$, где δ — плотность воды, которую примем равной 1000 Н/м^3 , g — ускорение свободного падения.

2. Если всю глубину H разбить на большое



4

количество маленьких промежутков длины $r = H/N$, то давление на i -й площадке со сторонами r и l будет постоянно и равно $P = \delta \cdot g \cdot x_i \cdot l \cdot r$, где $l \cdot r$ — площадь i -й площадки; $x_i = i \cdot r$.

3. При неограниченном увеличении числа площадок мы получим величину давления с любой точностью.

Снова воспользуемся аналогией с задачей о равноускоренном движении. Роль времени играет глубина, роль скорости — сила давления на данной глубине, роль пути — сила давления воды на плотину.

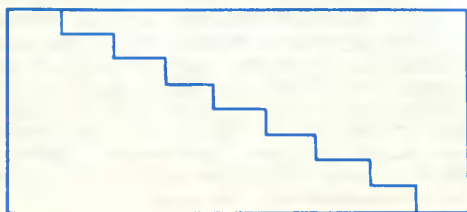
Определим, что считать исходными данными и результатом нашей модели. Исходными данными являются размеры плотины: l — длина, H — высота, δ — плотность воды, g — ускорение свободного падения. Результатом является P — сила давления воды на плотину.

Математическое соотношение между исходными данными и результатом строится точно так же, как в предыдущей задаче. Вот это соотношение (в нем N — число отрезков, на которые разбивается высота плотины):

$$P = \delta \cdot g \cdot l \cdot H^2 \cdot (N-1) / (2 \cdot N).$$

Задача 4. Школьник, не знающий теоремы Пифагора, составил следующую математическую модель для определения длины диагонали произвольного прямоугольного стола: предположения а) поверхность стола считаем прямоугольником; б) длины отрезков, на которые ломаная разбивает диагональ, можно считать равными сумме длин горизонтальной и вертикальной ступенек ломаной, если, конечно, эти длины достаточно малы;

исходные данные длины сторон стола a и b ;



5

результат длина диагонали d ; связь между исходными данными и результатом $d = a + b$ (действительно, длина ломаной в точности равна $a + b$ независимо от того, сколько в ней ступенек).

Почему получился неверный результат: диагональ прямоугольника равна сумме его смежных сторон?

Разбор задачи

Сравнив эту задачу с предыдущими двумя и с задачей о равноускоренном движении, учащиеся должны заметить, что в построенной школьником математической модели отсутствует важное предположение: при неограниченном уменьшении размера ступенек длина ломаной может быть сделана сколь угодно близкой к длине диагонали. Без этого предположения применять идею «измельчения» не имеет смысла, поэтому мы всякий раз высказывали подобные предположения, решая две предыдущие задачи и задачу о равноускоренном движении. Однако в данном случае это предположение не выполняется (что можно проверить, например, измерением). Потому и получился неверный результат.

Отметим, однако, что и построенную школьником модель можно считать математической моделью задачи о нахождении диагонали стола (ведь сделанный из бумаги самолетик тоже можно считать моделью самолета!).

Задача 1 завершает построение моделей, начатое в задачах 1 и 3 к § 2. Решая задачи 2 и 3, учащиеся осваивают общий прием построения приближенных математических моделей, основанный на «измельчении». Учащимся предлагается действовать по аналогии с решением задачи, разобранный в объяснительном тексте учебника [1]. При этом учащиеся должны сделать вывод, что для задач из разных предметных областей применим один и тот же метод построения математических моделей.

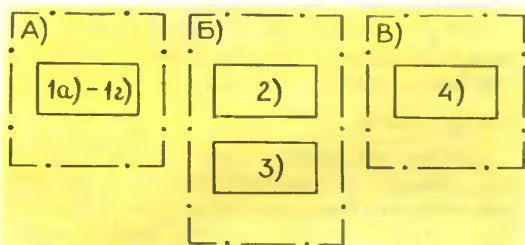
Задача 4 демонстрирует важный методологический принцип: *правильность и полнота предположений, на которых основана математическая модель, должны быть проверены на практике или строго доказаны.* В этой задаче описана математическая модель, которая приводит к неверным результатам не в силу каких-либо математических ошибок, сделанных при выборе соотношений, а в силу неправомерности исходных предположений.

И еще один вывод можно сделать после решения задачи 4: если мы заменяем реальную задачу ее моделью, то и ответ относится к модели и лишь опосредованно к исходной задаче (это общее свойство любых моделей, не только математических).

27

Задачи 1, 2, 3 и 4 относятся соответственно к первому, второму, третьему и четвертому уровням сложности.

Разбиение на блоки и группы:



Таким образом, изучение материала § 3 и решение задач к нему позволяет учащимся освоиться на практике с основными особенностями каждого из этапов построения математических моделей, а также «почувствовать» прием построения математических моделей, основанный на «измельчении», что дает возможность заменить «непрерывный» объект (скажем, отрезок прямой) «дискретным» объектом (например, конечным множеством точек). Это один из основных приемов, позволяющих применять ЭВМ: ведь в реальных задачах объекты, как правило, «непрерывны», а с помощью ЭВМ можно обозреть лишь конечное число случаев, лишь «дискретные» объекты. Впрочем, «измельчение» далеко не единственный способ сведения «непрерывного» к «дискретному». Например, задача нахождения площади фигуры может быть решена и с помощью «измельчения», и с помощью интегрирования в символическом виде (когда по формуле, задающей функцию $f(x)$, находится формула, задающая

$\int f(x)dx$), и с помощью метода Монте-Карло (см. [2]). Разумеется, приведенное рассуждение о соотношении «непрерывного» и «дискретного», скорее всего, недоступно учащимся IX класса, но, по нашему мнению, должно присутствовать во время объяснений учителя как бы «за кадром».

Глава 2. Алгоритм и его свойства

Эта глава начинает раздел «Алгоритмизация».

Перечислим соответствующие требования к учащимся, несколько видоизменив привычную методическую схему «знания, умения, навыки».

В результате изучения этого раздела учащиеся должны:

1) понимать, что такое алгоритм (т. е. понимать, что алгоритм — организованная

последовательность действий некоторого исполнителя);

2) понимать, что такое исполнитель алгоритмов (как сочетания «рабочего инструмента» и устройства управления);

3) понимать, что такое достижимые цели исполнителей алгоритмов;

4) понимать, что такое имитация исполнителей на ЭВМ*;

5) знать основные способы организации действий;

6) знать основные способы организации данных (переменные и таблицы);

7) уметь составлять «протоколы» выполнения алгоритмов, мысленно совершая действия алгоритма и комментируя их;

8) уметь находить ошибки в алгоритмах;

9) уметь записывать алгоритмы, не допускающая двусмысленности записи (мы не требуем строгого соблюдения определенной формы записи, но требование отсутствия двусмысленности считаем *обязательным*; скажем, из записи алгоритма должно быть понятно, где начинается и кончается ветвление, цикл или вспомогательный алгоритм);

10) уметь применять основные способы организации действий и данных при решении задач (при переходе от математической модели к алгоритму);

11) уметь использовать метод пошаговой детализации алгоритмов;

12) уметь переводить алгоритм на язык соответствующего исполнителя, симитированного на ЭВМ;

13) уметь проводить машинные эксперименты с помощью симитированных на ЭВМ исполнителей алгоритмов.

§ 4. Понятие алгоритма

Хотя эта тема на первый взгляд не вызывает методических трудностей, в процессе ее изложения велик риск неожиданных встреч с «подводными камнями». Укажем

* В принципе мы не являемся противниками «безмашинного варианта» обучения информатике. Конечно, если «безмашинно» обучать языку Бейсик или другому алгоритмическому языку, то курс ОИВТ будет просто вреден. Но в то же время ЭВМ не являются необходимыми для обучения умению ставить задачу, создавать математические модели, а также (что, может быть, более важно) умению планировать свои действия. Последнее умение, безусловно, является элементом общей культуры человека и должно быть сформировано еще в школьные годы. Но из всех школьных предметов курс ОИВТ наиболее приспособлен для этого. В других курсах крайне редко встречаются задачи, в решении которых планирование действий представляет самостоятельную трудность (хотя бы потому, что больше 3—4 действий обычно и не требуется).

примерный план соответствующего урока, позволяющий избежать «кораблекрушения» в самом начале изучения алгоритмизации.

Основная цель урока — показать, что алгоритм есть организованная последовательность действий. Начать, естественно, с нескольких примеров алгоритмов из обыденной жизни. Правда, к выбору этих примеров надо подходить осторожно. Дело в том, что четкая запись даже очень простой, понятной и привычной школьнику инструкции может оказаться непосильной для большинства школьников. Типичная *методическая ошибка* — предложение расхожей задачи о записи алгоритма перехода через улицу, с которой часто начинают первый урок неопытные учителя. Школьник сразу сталкивается с целым рядом трудностей.

Во-первых, это — отсутствие настроя на четкость мышления. Такой настрой необходим для словесной формулировки интуитивно ясных вещей: в голове у школьников есть целостный образ, а требуется дать его формальное, препарированное описание. При этом школьникам еще не продемонстрированы в явной форме примеры подобной препарации.

Во-вторых, учащиеся попросту не располагают необходимыми средствами для организации действий: они не знакомы с записью ветвлений, циклов и т. п.

В-третьих, в задачах такого типа трудно выделить «элементарные» действия, из которых будет состояться алгоритм. Произвол в выборе действий также создает весьма существенные трудности для школьников (вообще, задачи с неоднозначным решением сложнее для школьников, чем с однозначным).

Все это создает у школьника ложное впечатление, что от него требуют излагать сложным языком очевидные вещи. Фактически школьник попадает в положение сороконожки, которой предложено составить алгоритм своего передвижения.

Значит, учитель должен прежде всего настроить мышление учащихся на максимальную четкость. Здесь лучше всего применить ролевой метод обучения (этот метод весьма эффективен на уроках информатики; мы постараемся продемонстрировать его и в дальнейшем). Когда от человека требуется четко сформулировать алгоритм? Если от него ждут подробного и точного объяснения. Поэтому можно попросить учащихся представить себя в роли объясняющего. Полезно также предложить учащимся роль человека, которому предстоит воспользоваться чьим-либо алгоритмом.

Ясно, что первые алгоритмы, записанные учащимися, должны быть линейными и ко-

роткими. Ведь цель урока — лишь подвести учащихся к «определению» алгоритма как организованной последовательности действий.

Алгоритмы должны быть такими, чтобы простейшее изменение порядка действий приводило к невыполнимым алгоритмам или к алгоритмам, дающим «неправильные» результаты. Вот примеры заданий ученикам: «Представьте себе, что вам надо объяснить младшему брату, как открывать дверь ключом» или «Представьте себе, что вас пригласили в гости и вы не знаете, как проехать» (такие примеры подробно разобраны в [1]).

Еще одна типичная методическая ошибка — уделение слишком большого внимания так называемым свойствам алгоритмов и полному и строгому определению понятия алгоритма. Дело в том, что полного и строгого определения алгоритма не существует. Это фундаментальное и потому неопределяемое понятие информатики. Такие понятия существуют в любой науке: точка и прямая в геометрии, пространство и время в физике и т. д. Многие так называемые свойства алгоритмов — массовость, результативность и т. п. (см., например, [5], с. 31—34) — также не являющиеся ни четко формализуемыми, ни универсальными. Для того чтобы почувствовать это, можно попытаться ответить на следующие вопросы: если алгоритм содержит ошибки, то является ли он «массовым» или «результативным» (напомним известную истину: любая достаточно сложная программа содержит скрытые ошибки)? Являются ли «конечными» BASIC-программы для рисования фигур на экране «Ямахи» (напомним, что каждая графическая программа для «Ямахи» должна заканчиваться строкой типа `100 GOTO 100`, т. е., по существу, должна заикликоваться)?

Рискуя окончательно смутить читателя, скажем, что единственное, что можно сказать об алгоритме определенного, — это то, что он представляет собой каким-то образом организованную последовательность действий. Какие действия при этом имеются в виду и каковы способы их организации? — вот две основные проблемы раздела алгоритмизации. Естественно поэтому закончить урок указанием на возникшую проблемную ситуацию.

Приведем «комплект» задач к этому параграфу и разберем решения некоторых из них.

Задача 1. Приведите примеры неопределяемых понятий в следующих школьных предметах: а) «литература»; б) «русский язык»; в) «история»; г) «биология».

Решение

Вот некоторые примеры неопределяемых

понятий в школьных предметах: а) «литература» — сюжет, образ; б) «русский язык» — осмысленное слово; в) «история» — личность, время; г) «биология» — живое существо, вещество.

Задача 2. Некий злоумышленник выдал следующий алгоритм за алгоритм получения кипятка:

- Налить в чайник воду.
- Открыть кран газовой горелки.
- Поставить чайник на плиту.
- Ждать, пока вода не закипит.
- Поднести спичку к горелке.
- Зажечь спичку.
- Выключить газ.

Исправьте алгоритм, чтобы предотвратить несчастный случай.

Задача 3. Имеются цинк, 96 %-ная серная кислота, вода, а также колба и пробирка. Исправьте ошибки в алгоритме получения водорода.

- Поставить колбу на стол.
- Налить в колбу кислоту.
- Налить в колбу воду.
- Собрать выделяющийся газ в пробирку.
- Бросить в колбу цинк.

Комментарий к решению. Помимо очевидной перестановки четвертой и пятой строк необходимо поменять местами вторую и третью строки: это требование техники безопасности, с которым учащиеся знакомятся на уроках химии.

Задача 4. Какие действия вы бы добавили, чтобы был выполнен следующий алгоритм переправы через Волгу в районе г. Саратова?

- Подойти к реке.
- Войти в реку.
- Идти по дну, пока не выйдешь на другой берег.

Решение

Мысленно выполняя действия исходного алгоритма, приходим к выводу, что алгоритм невыполним, так как исполнитель утонет. Понятно, что нужно добавить некоторые действия, чтобы успешно переправиться через Волгу. Приводим одно из решений.

- Подойти к реке.
- Войти в реку.
- Сесть в лодку.
- Плыть, пока не станет мелко.
- Выйти из лодки.
- Идти по дну, пока не выйдешь на другой берег.

Задача 5. Пусть дан отрезок АВ. Определить, для решения какой задачи предназначен следующий алгоритм.

- Поставить ножку циркуля в точку А.
- Установить раствор циркуля равным длине отрезка АВ.
- Провести окружность.

Поставить ножку циркуля в точку В.
Провести окружность.

Провести прямую через точки пересечения окружностей.

Задача 6. Пусть даны действительные числа b , q и натуральное число n . Какая задача решается с помощью следующего алгоритма? Вычислить q^n и обозначить результат буквой С.

Вычислить $1 - C$ и обозначить результат буквой D.

Вычислить $1 - q$ и обозначить результат буквой E.

Разделить D на E и обозначить результат буквой F.

Умножить F на b и обозначить результат буквой S.

Задача 7. Дана фраза на английском языке: 'Mike goes to school'. Составьте алгоритм перехода от утвердительной формы к вопросительной: 'Does Mike go to school?'

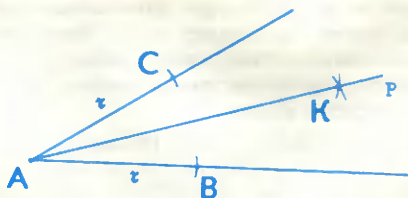
Задача 8. Составьте алгоритм построения биссектрисы угла с помощью циркуля и линейки.

Разбор решения

Вот как может выглядеть «нечеткая» запись того, что должно происходить при построении биссектрисы угла.

Берем циркуль (раствор циркуля произвольный) и отмечаем на двух сторонах угла точки (ножка циркуля стоит в вершине угла). Далее, поставив ножку циркуля в отмеченные точки на сторонах, рисуем дуги во внутренней области угла. Соединяем точку пересечения дуг и вершину угла прямой. Получим биссектрису угла.

Это решение иллюстрирует рисунок.



Запишем наши действия более строго в виде алгоритма:

- Поставить ножку циркуля в точку А.
- Провести окружность α радиуса r .
- Найти точки пересечения В и С окружности α со сторонами угла.
- Поставить ножку циркуля в точку В.
- Провести окружность β с центром в точке В радиуса r .
- Поставить ножку циркуля в точку С.
- Провести окружность γ с центром в точке С радиуса r .

Найти точку К пересечения окружностей α и β , отличную от А.

Провести прямую p через точки А и К.

Задача 9. Даны число x и набор действий: разделить полученное число на 3; умножить x на 2; сообщить результат; прибавить к полученному числу 4; вычесть из полученного числа 7.

Составьте из этих действий два различных алгоритма. Любой ли алгоритм, составленный из этих действий, можно выполнить? Укажите две различные функции от x , значения которых вычисляются с помощью алгоритмов, использующих указанные действия.

Задача 10. Имеются два кувшина емкостью 3 и 8 л. Напишите алгоритм, выполняя который можно набрать из реки 7 л воды (разрешается пользоваться только этими кувшинами).

Задача 11 (старинная). Некий человек должен перевезти в лодке через реку волка, козу и капусту. Каждый раз он может перевезти только либо волка, либо козу, либо капусту. На одном берегу нельзя оставить вместе козу и волка, а также козу и капусту. Составьте алгоритм переправы на другой берег. (Эта задача встречается в рукописях VIII в. Уже тогда интересовались алгоритмами!)

Задача 12. Разведывательный дозор в составе двух человек подошел к реке. Мост был разрушен, а река слишком глубока и широка, чтобы переправиться через нее вброд или вплавь. К счастью, около берега в маленькой лодке проплывали два мальчика. Как переправиться на этой лодке через реку, если она может выдержать либо одного взрослого, либо двух мальчиков?

Задача 13. На полустанке однопутной железной дороги остановился поезд в составе тепловоза и пяти вагонов, доставивший бригаду рабочих для строительства новой ветки. Пока на этом полустанке имеется только небольшой тупик, в котором в случае необходимости может поместиться тепловоз с двумя вагонами или три вагона. Вскоре следом за поездом со строительной бригадой к этому же полустанку подошел пассажирский поезд. Составьте алгоритм, позволяющий пропустить пассажирский поезд.

Задача 1 призвана продемонстрировать, что каждый школьный предмет содержит понятия, строгое определение которых дать невозможно.

Задачи 2—4 относятся к типу задач на нахождение ошибки в алгоритме. Для решения задачи сначала нужно показать, почему предложенный алгоритм неверен, а затем устранить ошибку. Задача 4 служит и для преподавания понятия допустимых дей-

ствий исполнителя алгоритмов: алгоритм, не выполнимый человеком, может быть выполнен, например, роботом-подводником.

Задачи 5 и 6 относятся к типу задач на определение того, что произойдет в результате выполнения алгоритма. Здесь полезны протоколы выполнения алгоритмов.

Задачи 7—13 относятся к типу задач на составление алгоритма. Следует проследить за правильным его написанием. В частности, ученики должны расположить действия в столбик одно под другим.

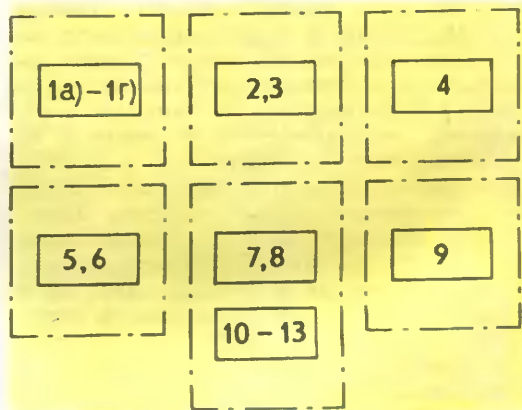
Задачи 7 и 9 просты и нужные алгоритмы записываются «с ходу». Дополнительная цель задачи 9 — продемонстрировать, во-первых, что из одних и тех же действий можно составить разные, в том числе и невыполнимые, алгоритмы, а во-вторых, что один и тот же результат можно получить с помощью разных алгоритмов.

Задачи 8 и 10—13 более сложные. Перед написанием соответствующих алгоритмов необходимо более или менее подробно представить себе то, что должно происходить при их выполнении, и записать это в произвольной, «нечеткой» форме. Для задачи 8 мы приводим возможный вариант такой записи.

Наконец, отметим, что задачи 5—8 показывают учащимся, что алгоритмы естественно возникают в школьных курсах геометрии, алгебры и др.

К первому уровню относятся задачи 2, 3, 4, 5, 6, 7, 9, 11, 12. Ко второму уровню относятся задачи 1а) — 1г), 8, 10, 13.

Разбиение на блоки и группы:



§ 5. Исполнители алгоритмов

Одним из основных вопросов, который активно обсуждается учителями информатики, авторами программ, учебников и учебных пособий по курсу ОИВТ, является вопрос

о роли исполнителей алгоритмов в курсе ОИВТ. Мы являемся сторонниками активного использования исполнителей алгоритмов при обучении информатике. Постараемся обосновать свою точку зрения.

1. Как уже говорилось в [2] и [3], основная цель курса ОИВТ — научить школьников решать задачи с помощью ЭВМ. При этом принципиально важно, что для решения каждого класса задач требуется свой набор действий, свой исполнитель. И для того чтобы «научить» ЭВМ решать задачи данного класса, необходимо симитировать на ней допустимые действия соответствующего исполнителя. Понимание школьниками этого обстоятельства лучше всего достигается при использовании в курсе ОИВТ нескольких существенно различных исполнителей (и соответственно нескольких различных классов задач).

2. Выпускники школ будут встречаться в своей деятельности не только с универсальными ЭВМ, «понимающими» языки программирования, изучаемые в курсе ОИВТ, но и с другими автоматическими программируемыми устройствами, например со станками с ЧПУ. Понятие исполнителя как сочетания «инструмента» и устройства управления (устройство управления воспринимает алгоритм и обменивается информацией с «инструментом») охватывает и ЭВМ, и другие программируемые устройства.

3. Одна из главных задач курса ОИВТ — показать широкое применение ЭВМ. Но в действительности каждое применение ЭВМ базируется на том, что с ее помощью можно имитировать различные наборы допустимых действий, различных исполнителей алгоритмов. Эта точка зрения позволяет с единых позиций подойти и к прикладным программам (редактору текстов, информационно-поисковой системе, электронной таблице и др.), также рассматривая их как имитацию определенных исполнителей с помощью ЭВМ.

4. Исполнители, симитированные на ЭВМ (или другим способом), играют в курсе ОИВТ роль технических средств обучения. Разнообразие исполнителей предоставляет учителю богатый арсенал методических средств. (В [2] указано, для достижения каких целей служит каждый из трех исполнителей, встречающихся в учебнике [1].)

5. Ученики привыкли к алгоритмам, предназначенным для человека, которые можно записывать в произвольной форме, не фиксируя список допустимых действий. А при работе на ЭВМ необходима и строгость в записи конструкций, и понимание ограниченности набора допустимых действий. В то же время указать полный набор допустимых действий ЭВМ практически невозможно. При-

менение исполнителей алгоритмов позволяет обойти эту трудность естественным путем.

6. Простые и наглядные исполнители алгоритмов чрезвычайно важны и при «безмашинном обучении». Можно сказать, что они придают ему «освязаемость», поскольку допускают имитацию не только на ЭВМ, но и, скажем, на классной доске (см., например, [6] и [9]). Кстати, такая имитация полезна и при «машинном варианте».

7. Понятие класса достижимых целей эффективно иллюстрируется только для очень простых исполнителей. Понятие исполнителя позволяет также продемонстрировать, что расширение класса достижимых целей возможно лишь за счет наращивания набора допустимых действий.

8. Только решая задачи на составление алгоритмов, содержащих большое количество действий, учащиеся приобретут навыки (и желание) правильно планировать деятельность (что является одной из главных целей курса ОИВТ и школы вообще). Составляя такие алгоритмы, учащиеся должны думать именно об организации действий, а не над тем, какие сведения из конкретной предметной области привлечь, чтобы решить задачу. И в этом учителю могут помочь исполнители алгоритмов. Если сделать набор допустимых действий достаточно «бедным» и наглядным, то решение даже очень простой задачи потребует большого числа действий и по сути сведется к их правильной организации.

9. Применение на уроках алгоритмизации разнообразных исполнителей позволяет изучать алгоритмические конструкции в общем виде, независимо от конкретного языка программирования. Это разделение обеспечивается тем, что различные исполнители «понимают» одни и те же алгоритмические конструкции, но имеют разные наборы допустимых действий. В частности, появляется возможность изучать алгоритмические конструкции в «чистом виде», используя исполнителей с простейшим набором допустимых действий. Точно так же применение исполнителей позволяет изучать в «чистом виде» табличную форму организации данных.

10. Обычно противники использования исполнителей утверждают, что им хватает лишь одного исполнителя — ЭВМ. Это утверждение не совсем корректно, поскольку неявно подразумевается ограничение возможностями какого-либо языка, скажем Бейсика. Более точным было бы высказывание, что рассматривается исполнитель Бейсик. Однако выбор исполнителя Бейсик (или, скажем, исполнителя Паскаль) в качестве единственного исполнителя вряд ли может считаться методически удачным. Для одних методических целей нужны более простые исполни-

тели, для других — более сложные (см. п. 6, 7, 8).

11. При обучении школьников программированию лишь на одном каком-либо конкретном языке невольно создается стереотип языка общения с ЭВМ, и при переходе к другому языку возникают значительные трудности. Наличие разнообразных исполнителей помогает преодолеть эту трудность.

Напомним, что в СССР традиция активного использования исполнителей алгоритмов в средней школе на уроках программирования и информатики была заложена Г. А. Звенигородским [6]. Параллельно та же идея использовалась А. Г. Кушниренко для обучения программированию студентов МГУ. До некоторой степени, хотя, на наш взгляд, и не вполне последовательно, эта традиция была продолжена пробным школьным учебником [7]. В [8], по сути дела, предлагается сделать понятие исполнителя основным в информатике, что сближает пособия [8] и [6]. Впрочем, и методические цели [8] близки к целям Г. А. Звенигородского: и в [6] и в [8] главная цель — обучение программированию. Общность целей определяет и общность в подборе исполнителей алгоритмов в [6] и [8]. При этом, чтобы продемонстрировать возможности ЭВМ в решении разнообразных классов задач, ими созданы исполнители с достаточно богатыми возможностями. Поэтому между простыми исполнителями (ДЕЖУРИК, МУРАВЕЙ, МАШИНИСТ из [6]), с помощью которых могут решаться лишь весьма несложные задачи, и «богатými» исполнителями (МАЛЫШ из [6], ЧЕРТЕЖНИК из [6] и [8], РОБОТ, ЧЕРЕПАШКА из [8]) существует значительный разрыв. Простые исполнители не могут проверять условия и потому не могут быть использованы для обучения алгоритмическим конструкциям — ветвлениям и циклам. По сути дела, не реализуется возможность изучать алгоритмические конструкции в «чистом виде» (см. п. 9). Отсутствие специального исполнителя для работы с таблицами не позво-

ляет изучать в «чистом виде» и организацию данных.

Выбор исполнителей в нашем учебнике обусловлен тремя основными целями изучения алгоритмизации: *научить организовывать действия, организовывать данные и применять ЭВМ к решению разнообразных классов задач.* Для достижения первой из этих целей служит исполнитель ЧЕРТЕЖНИК, для достижения второй цели — РОБОТ-МАНИПУЛЯТОР, третьей — ВЫЧИСЛИТЕЛЬ. Описание этих исполнителей см. в [2].

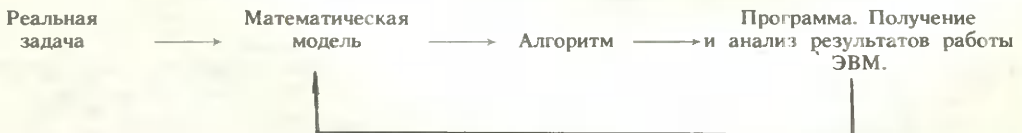
Продолжение следует

Литература

1. Основы информатики и вычислительной техники / А. Г. Гейн, В. Г. Житомирский, Е. В. Линецкий и др. Свердловск: УрГУ, 1989.
2. Гейн А. Г., Житомирский В. Г., Линецкий Е. В., Сапир М. В., Шолохович В. Ф. Программно-методический комплекс по курсу школьной информатики // Информатика и образование. 1988. № 3.
3. Гейн А. Г., Линецкий Е. В., Сапир М. В., Шолохович В. Ф. Информатика: как решать задачи с помощью ЭВМ // Информатика и образование. 1989. № 2.
4. Гейн А. Г., Сапир М. В., Шолохович В. Ф. и др. Дидактические материалы по курсу информатики. Гл. 1—3. Свердловск: СвГПИ, 1988.
5. Изучение основ информатики и вычислительной техники: Методическое пособие для учителей и преподавателей средних учебных заведений. Ч. 1 / Под ред. А. П. Ершова, В. М. Монахова. М.: Просвещение, 1985.
6. Звенигородский Г. А. Первые уроки программирования. М.: Наука, 1985.
7. Основы информатики и вычислительной техники: Пробное учебное пособие. Ч. 1 / Под ред. А. П. Ершова, В. М. Монахова. М.: Просвещение. 1985.
8. Основы информатики и вычислительной техники: Пробный учебник для средних учебных заведений / Под ред. А. П. Ершова. М.: Просвещение, 1988.
9. Гутман Г., Карпилова О. Азбука программирования // Информатика и образование. 1987. № 6; 1988. № 2.

33

От редакции. В № 2 на с. 14 схему следует читать:





В. КАЙМИН, Л. КУГЕЛЬ, Е. КУЗНИЦКИЙ,
И. ТОЧЕННАЯ

Компьютерная грамотность в примерах и задачах

34

Обучение компьютерной грамотности в соответствии с программой курса «Основы информатики и вычислительной техники» по машинному варианту считается задачей-минимум для учащихся IX — X классов. Существует несколько вариантов толкования понятия «компьютерная грамотность», в которых выделяются технические и гуманитарные аспекты. И в то же время в них есть общая минимальная часть, включающая знание основных возможностей современных ЭВМ и умение подготавливать информацию различного рода с помощью персональных компьютеров.

Ключевыми понятиями в этой части курса являются слова «компьютер», «информация», «подготовка информации». Основные виды информации, подготавливаемой с помощью ЭВМ, — тексты, чертежи, рисунки, различного рода таблицы и документы. Важно уметь проводить расчеты и получать справочную информацию с помощью машины. По этой причине машинный вариант обучения информатике должен, на наш взгляд, начинаться с цикла лабораторных работ со следующими учебными программами:

- 1) клавиатурный тренажер,
- 2) редактор текстов,
- 3) графический редактор,

Продолжение. Начало см.: Информатика и образование. 1989. № 2.

- 4) электронные таблицы,
- 5) база данных,
- 6) экспертные системы.

Параллельно с лабораторным практикумом на ЭВМ учащимся сообщаются основные понятия информатики: информация, ее виды и свойства, количество информации. Далее — основные понятия элементарной логики — логические связи: «и», «или», «не», суждение и рассуждение, индукция и дедукция, простейшие законы логики.

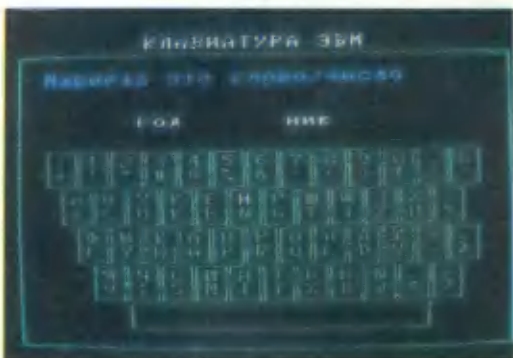
Понятие информации следует давать в развитии. В простейшем толковании «информация» — это некоторые сведения, которые могут передаваться устно, письменно, с помощью технических устройств и т. п. В качестве видов представления информации необходимо указать тексты, рисунки, чертежи, таблицы, документы, справки — все то, что в дальнейшем будет подготавливаться и обрабатываться учащимися с помощью ЭВМ. Среди свойств информации особо следует выделить достоверность и полноту.

Тема, связанная с электронными таблицами, предваряется обсуждением понятий: постановка задачи и анализ ее условий, способы и методы решения задач, проверка правильности результатов. Во всех вариантах проведения уроков должно сохраняться соотношение времени 1:1:1 изучения элементов теории, практикума решения задач и работы на ЭВМ. При этом практикум должен слу-



жить подготовкой к работе на ЭВМ с индивидуальными заданиями — текстами, рисунками, чертежами, таблицами и данными.

Клавиатурный тренажер



Знакомство с персональными ЭВМ и изучение правил работы на компьютерах желательно начать, используя программу клавиатурный тренажер, предварительно ознакомив учащихся с правилами техники безопасности.

Редактор текстов



Работу с учебными редакторами текстов следует предварить кратким изложением их назначения, а также описанием основных команд. При обсуждении возможностей этих программ следует особо обратить внимание на достоинства персональных ЭВМ, в отличие от печатающих машинок, — отсутствие боязни опечаток, возможность хранить тексты и размножать их в любом количестве экземпляров, передавать по телефонной сети и многократно редактировать.

В число основных функций редакторов текстов, как известно, входят:

- 1) подготовка и редактирование текстов,
- 2) сохранение их на магнитных дисках (или на лентах),
- 3) печать текстов (сплошная или постраничная),
- 4) рассылка текстов по сети,
- 5) получение справок о системе команд.

Среди команд редактирования и ввода текстов следует явно указать: команды вставки/удаления строк, команды расщепления/склейки строк, команды записи/чтения текстов.

Необходимо обратить внимание на клавиши перемещения курсора по экрану, клавиши вставки / удаления символов.

Остальные команды сообщаются дополнительно по мере необходимости и их наличия в редакторах текстов. Задания на освоение редактора текстов могут быть самые разнообразные: задания на исправление ошибок в заранее заготовленных текстах, самостоятельная подготовка и ввод текстов, и т. п. Одним из интересных способов освоения умений вводить и редактировать тексты является организация конкурса на правильное введение и редактирование стихотворений, которые самостоятельно выбирают или даже сочиняют ученики. Для определенности можно рекомендовать вспомнить любимые стихи. Результаты конкурса лучше всего оценивать вместе с учениками по отпечатанным текстам.

35

Графический редактор

Работа с графическими редакторами доставляет удовольствие всем учащимся, от мала до велика. Дети в течение 2—3 мин осваивают первые простейшие команды рисования линий, окружностей и т. п. При наличии цветных дисплеев возможно создание цветных рисунков, а в наиболее совершенных редакторах возможна перекomпоновка изображений. Главное — рисунок можно сохранить на магнитных дисках для последующих демонстраций и отпечатать на бумаге.

В число основных средств графических редакторов входят:

- 1) выполнение и редактирование рисунков,
- 2) сохранение рисунков на магнитных носителях,
- 3) вывод рисунков на печатающее устройство,
- 4) рассылка рисунков по сети,
- 5) чтение оглавления,
- 6) получение справок о наборе команд.

Наборы команд и правила работы с графическими редакторами обычно сильно отличаются друг от друга; их следует уточнять по описаниям. В то же время во всех

имеются следующие средства рисования: перемещение курсора по экрану; построение точек, отрезков, дуг, прямоугольников, окружностей и надписей; удаление только что нарисованного элемента; раскраска и стирание фигур.

Каждый ученик должен усвоить, как получить справку о командах и правилах работы данного графического редактора. Кратко изложив назначение графического редактора и его основные возможности, можно сразу же предложить учащимся совместно выполнить несложный рисунок, например изображение кораблика.

Команды редактора следует вводить по мере надобности непосредственно во время работы. Необходимо обратить внимание учащихся на возможность редактора удалять ошибочно построенные элементы. После завершения построений необходимо сохранить полученное изображение, чтобы его можно было использовать вновь, например усложнить задачу (дорисовать море и солнце).



В качестве самостоятельной работы предложите ученикам создать свои рисунки. Лучшие рисунки можно обсудить всем классом и оценить достоинства работ (выдумка, эстетичность). Желательно получить твердые копии этих рисунков.

Лабораторные работы с графическими редакторами можно строить двумя способами. Первый — изложение у доски возможностей графических редакторов с разбором примеров и перечня основных команд, а уже затем рекомендация ученикам самостоятельно воспроизвести эти примеры на ЭВМ. Второй — синхронное выполнение учениками действий на ЭВМ по командам, которые учитель записывает на доске. В этом случае учитель выполняет роль дирижера, последовательно выписывая на доске партитуру — команды, которые ребята воспроизводят на компьютере.

В обоих случаях далее должны следовать самостоятельные импровизации, которые могут быть подготовлены в форме эски-

зов в тетрадах. Важно, что при работе с графическими редакторами развивается самостоятельное творчество учащихся. Чем больше свободы будет предоставлено при выборе тем рисунков, тем большей заинтересованности можно добиться от ученика. Темы, которые могут быть рекомендованы, — «Любимые цветы», «Новое платье», «Космические фантазии», «Дом будущего», «Робот в огороде» и т. п.

База данных

Следующая важная тема — организация баз данных на ЭВМ. Современные компьютеры — это не только удобное средство для подготовки текстов и графической информации, но и эффективный инструмент для накопления и оперативного поиска данных, который в будущем заменит бумажные каталоги, справочники и архивы. Средством замены бумажных носителей станут магнитные и оптические диски, позволяющие размещать большие объемы информации с помощью специальных программ, называемых базами данных.

Главным достоинством баз данных на ЭВМ является быстрота поиска информации — порядка нескольких секунд. Диапазон предоставляемой информации зависит от того, какие архивы хранятся в ЭВМ или доступны в информационной сети, к которой может подключаться компьютер. Даже учебные базы данных позволяют школьникам создавать на ЭВМ различные каталоги: библиотечные, музыкальных записей, имеющихся программ, всевозможных коллекций. Можно заводить телефонные книжки, журналы учета ведомости успеваемости и т. п.

Простейшими считаются табличные базы данных, которые в технической литературе называются базами данных реляционного типа. В них хранимая информация может представляться таблицами. На персональных ЭВМ эти таблицы непосредственно отображаются на экране дисплея. Иллюстрацией может служить база данных с ведомостью успеваемости учеников.

№	1	2	3	4	5
	фамилия	имя	лит-ра	физика	химия
1	Петров	Саша	3	5	3
2	Симонов	Женя	4	3	4
3	Кайкина	Оля	5	4	4

В число функций табличных баз данных входят:

- 1) создание новых таблиц,
- 2) заполнение и редактирование таблиц,

3) упорядочение и поиск информации,
4) сохранение таблиц и печать найденной информации,

5) пересылка запросов и информации по сети.

При создании новых таблиц задается формуляр, в котором указывается число столбцов, их названия и типы значений — числовой или символьный. Заполнение таблиц проводится последовательным вводом строк. Для заполнения или редактирования таблиц в базе данных должны быть следующие команды: ввод новых строк, последовательный просмотр строк, редактирование имеющихся строк, удаление ненужных строк, запись таблицы на магнитный носитель.

Поиск информации в табличных базах данных проводится указанием столбцов и конкретных значений. Например, в приведенной таблице можно найти всех учеников с именем Оля, указав соответствующий столбец, либо найти всех учеников, имеющих оценку «4» по физике. Поиск может быть указан и диапазоном значений, например: найти всех учеников, имеющих оценки от «3» до «5» по литературе. Конкретный набор команд и правил работы опять же зависит от типа используемой базы данных, их можно уточнить по руководству.

Наиболее сложные базы данных позволяют искать информацию по целой комбинации признаков, соответствующие условия связываются логическим «и». К примеру, можно найти всех учеников, имеющих оценки «5» по физике и литературе. В таких базах данных возможно переупорядочение элементов. По числовым столбцам таблицы упорядочиваются в порядке возрастания заданных значений. Пример — упорядочение приведенной ранее таблицы по оценкам по физике. Упорядочение таблиц по столбцу с символьными значениями проводится в алфавитном порядке, который в математической литературе называется также лексикографическим. Пример — упорядочение нашей таблицы по фамилиям.

Выполнение лабораторной работы следует начать с заранее заготовленной базой данных с информацией, имеющей самое прямое отношение к ученикам. Первое знакомство проводится в виде фронтальной работы с диктовкой последовательности команд, сразу проверяемых учениками на ЭВМ, либо с предварительным разбором примера у доски и записью последовательности команд в тетрадь.

Последовательность изучаемых команд может быть, например, такой.

1. Упорядочение учащихся по оценкам по литературе.

2. Лексикографическое упорядочение по фамилиям.

3. Поиск учащихся по имени Оля.

4. Поиск учащихся, имеющих по физике оценку не ниже «4».

5. Поиск тех учащихся, которых зовут Оля и у которых оценка по физике не ниже «4».

6. Поиск всех учеников, имя которых отлично от имени Оля.

Остановимся кратко на связи между знакомством с базами данных и изучением некоторых понятий логики.

Обратим внимание на формы запросов в пунктах 3, 4 и 5. Это хороший материал для иллюстрации понятий простого и составного высказывания, логических связей «и», «или» и «не» и иллюстрации этих понятий с помощью диаграмм. Соотнесение результатов запросов в пунктах 3, 4 и 5 дает возможность рассмотреть связку «или» и выявить различия между ней и связкой «и». Пункт 6 иллюстрирует понятие отрицания. Далее имеет смысл остановиться на законах де Моргана. С помощью системы управления базой данных можно продемонстрировать учащимся практическую значимость умения строить отрицания составных высказываний, дать достаточный материал для упражнений.

Дальнейшая работа может быть связана с легко обозримыми (короткими) базами данных, во-первых, для того чтобы учащиеся могли для закрепления основных понятий давать ответы сами, вместо машины, а во-вторых, чтобы научиться заносить в таблицы новую информацию и редактировать имеющуюся. Завершить изучение темы можно, предложив каждому школьнику создать свою базу данных, например «Телефонную книгу», «Каталог домашней библиотеки», «Каталог музыкальных записей» и т. п.

Одной из тем практикума по компьютерной грамотности является изучение и освоение электронных таблиц. Эту тему следует начинать с разъяснения: что это такое и для чего они нужны? Напомним, что электронная таблица — это бланк, изображаемый на экране ЭВМ, в который можно записывать числа, надписи и формулы. Иллюстрация — бланк раскладки продуктов в поход. Основным достоинством электронных таблиц является мгновенный пересчет числовых значений при изменениях в исходных данных. В приведенной таблице исходными данными являются раскладка количества и цены продуктов, а также количество людей и дней в походе. При изменении числа людей или дней машина сразу же пересчитает итоговую сумму, а при из-

менении цены или количества какого-то продукта пересчитает сумму на этот продукт и итоговую сумму в целом.



38

Запись информации в электронных таблицах проводится непосредственно на экране по клеткам бланка. В каждую клетку может быть вписано число или надпись. Все клетки бланка нумеруются по типу клеток шахматной доски — A1, A2, B1, B2, C1 и т. д. К любой из клеток может быть приписана некоторая расчетная формула. Например, клетке D3 приписана формула $D3=B3*C3$, которая означает, что число в D3 должно быть равно произведению B3 на C3. Это число автоматически пересчитывается, как только изменяется число в клетке B3 или C3.

Заранее заготовленный бланк обычно записывается на магнитный носитель и может быть многократно использован для подготовки различных счетов, калькуляций и смет. После окончательного заполнения и пересчета бланк может быть отпечатан, отправлен по сети или вновь быть записан на магнитный носитель.

Основными функциями электронных таблиц являются:

- 1) заполнение и редактирование таблиц,
- 2) запись и чтение бланков с магнитных носителей,
- 3) вывод бланков на печатающее устройство,
- 4) рассылка бланков по сети,
- 5) чтение оглавления бланков.

Наборы команд, разумеется, зависят от конкретного типа электронных таблиц, и их необходимо уточнять по соответствующим руководствам. Наиболее важными командами для редактирования, заполнения и выполнения расчетов являются перемещение кур-

сора по таблице, запись и редактирование чисел и надписей, запись и редактирование формул, выполнение перерасчетов.

Наиболее существенный момент — уточнение по руководству правил записи формул и перечня элементарных функций, которые могут быть заданы формулами, — извлечение корней, возведение в степень, логарифмы, тригонометрические функции и т. п. Важными дополнительными возможностями являются закрепление символических имен типа x, y и т. д. и использование условных выражений. Примером условного выражения может быть такая запись: $D3=\text{если } B3>0, \text{ то } B3*C3 \text{ иначе } 0$, которая означает, что в клетке D3 должно появляться значение формулы $B3*C3$, если $B3>0$, и значение 0, если $B3\leq 0$. Форма записи условных выражений в значительной мере зависит от типа программы.

Проведение лабораторных работ с электронными таблицами может проводиться либо синхронно, либо с предварительным разбором примера и записью правил работы в тетрадах. Главным результатом в работе должен стать расчет нескольких вариантов решений и определение среди них наилучших по определенным критериям.

Для самостоятельных работ может быть дано задание на подготовку и заполнение бланков на темы «Семейный бюджет», «Бригадный хозрасчет», «Закупка продуктов», а также разбор решения математических задач на темы «Решение простейших уравнений», «Решение квадратных уравнений», «Решение системы двух линейных уравнений с двумя неизвестными».

Подведем итоги. Обучение компьютерной грамотности в минимальном объеме предполагает формирование у учащихся следующих знаний и умений.

Знать основные возможности: персональных ЭВМ, редакторов текстов, графических редакторов, табличных баз данных, электронных таблиц.

Уметь на персональных ЭВМ: работать на клавиатуре, редактировать тексты, выполнять рисунки и чертежи, искать и редактировать информацию в базах данных, проводить вычисления в электронных таблицах.

Продолжение следует.

Литература

1. Клейман М. Школа будущего: компьютеры в процессе обучения. М.: Радио и связь, 1988.
2. Брябрин В. Программное обеспечение персональных ЭВМ. М.: Наука, 1988.

Основные понятия языка Рапира

Процедуры и функции в Рапире

Нами была рассмотрена задача о подсчете числа букв М в произвольном тексте. Как быть, если теперь надо определить число букв Л? букв Б? букв П? Можно еще раз набрать старую программу с необходимыми изменениями. Можно, но... не очень хочется делать одну и ту же работу несколько раз.

Нельзя ли один раз набрать программу, запомнить, исполнить ее, потом внести необходимые изменения, опять исполнить и т. д.? Это можно сделать, если дать программе имя и хранить ее текст в блоке памяти с указанным именем. Такая программа называется процедурой. Запишем в виде процедуры программу для подсчета числа букв М в тексте Т.

```
ПРОЦ ПОДСЧЕТ_М()
Т:="МОЙ МАЛЕНЬКИЙ ГНОМ"
СЧЕТЧИК:=0
для ном от 1 до #Т
цикл
если Т[НОМ]="М" \ *
то СЧЕТЧИК:=СЧЕТЧИК+1
все
кц
вывод: "СИМВОЛ М: ", СЧЕТЧИК \ *
конец
```

Такая запись называется *описанием процедуры*.

Описание процедуры начинается с заголовка и заканчивается словом КОНЕЦ. Между заголовком и словом КОНЕЦ записывается последовательность предписаний, которая называется *телом процедуры*. Заголовок начинается словом ПРОЦ, за которым следует имя процедуры и пара круглых скобок (в них может быть записана некоторая дополнительная информация, об этом будет рассказано позже).

Рассмотрим, какие действия выполняются при записи программы в виде процедуры. После набора заголовка процедуры выделяется блок памяти с указанным именем, и ЭВМ переходит в режим запоминания. В этом режиме надо набрать весь текст процедуры. При этом предписания, входящие в тело процедуры, не исполняются, а запоминаются.

Если окажется, что в памяти машины уже есть процедура с таким именем, то ее текст будет выведен на экран, в него можно внести необходимые изменения. Набрав слово КОНЕЦ, мы сообщаем машине, что закончили описание процедуры. Теперь весь текст процедуры хранится в блоке памяти с заданным именем.

Для того чтобы процедура начала исполняться, необходимо *предписание вызова*. Предписание вызова состоит из имени процедуры и круглых скобок (в них может быть записана дополнительная информация). Для вызова процедуры ПОДСЧЕТ_М надо набрать ПОДСЧЕТ_М().

Вернемся к нашей задаче. Если вычисления нужно выполнить для нескольких букв, то можно поступить так. Набрать текст процедуры ПОДСЧЕТ_М, вызвать ее, записать полученный результат. Затем внести в текст процедуры необходимые изменения для подсчета, например, букв А (в строчки, отмеченные знаком *) и опять вызвать процедуру, записать результат и т. д.

К сожалению, вносить изменения и выполнять вызов процедуры нужно для каждой буквы отдельно! Хорошо бы составить такую программу, которой во время исполнения можно было бы сообщать, для какой буквы требуется выполнять вычисления. Это можно сделать с помощью *предписания ввода*. Для этого надо написать ВВОД: и после двоеточия указать имя, которому надо присвоить вводимое значение. Например, ВВОД: СИМВОЛ.

После двоеточия может быть указано и несколько имен, разделенных запятыми. Во время исполнения этого предписания на экран выдается специальное приглашение (например, знак ?), в ответ на которое человек должен набрать столько значений (чисел, текстов и т. п.), сколько указано имен. При этом значения отделяются друг от друга запятой, а тексты записываются в кавычках. Каждому имени будет присвоено свое значение.

Например, если в ответ на приглашение во время исполнения предписания ВВОД: ПРЕДМЕТ, КЛАСС набрать «ИНФОРМАТИКА», 9 то значением имени ПРЕДМЕТ станет текст ИНФОРМАТИКА, а имени КЛАСС — чис-

ло 9. То есть результат выполнения этого предписания такой же, как и выполнения следующих двух предписаний присваивания: ПРЕДМЕТ: = «ИНФОРМАТИКА» КЛАСС: = 9

Когда значения, которые нужно присвоить данным именам, известны заранее и не должны изменяться при различных вызовах процедуры, используют предписания присваивания, в противном случае — предписания ввода.

Когда требуется ввести ровно один текст, можно воспользоваться специальной формой предписания ввода:

ВВОД ТЕКСТА: имя

В этом случае во время исполнения предписания ввода текст набирается без кавычек.

Запишем описание процедуры для подсчета в тексте количеств каждого из интересующих нас символов, используя предписание ввода.

40

```
ПРОЦ ПОДСЧЕТ ()
  ВЫВОД: "ВВЕДИТЕ ТЕКСТ"
  ВВОД ТЕКСТА: Т
  ВЫВОД: "ВВЕДИТЕ СИМВОЛ"
  ВВОД ТЕКСТА: СИМВОЛ
  СЧЕТЧИК: =0
  ДЛЯ НОМ ОТ 1 ДО #Т
  ЦИКЛ
    ЕСЛИ Т[НОМ]=СИМВОЛ
      ТО СЧЕТЧИК:=СЧЕТЧИК+1
  ВСЕ
КЦ
ВЫВОД: "СИМВОЛ ", СИМВОЛ, ": ", СЧЕТЧИК
КОНЕЦ
```

Наберем предписание вызова ПОДСЧЕТ(). На экране появится запрос ВВЕДИТЕ ТЕКСТ?

В ответ наберем АБРАКАДАБРА, после чего на экране появится запрос ВВЕДИТЕ СИМВОЛ?

Мы ответим А.

После выполнения процедуры на экране останется следующий текст:

```
ВВЕДИТЕ ТЕКСТ
?АБРАКАДАБРА
ВВЕДИТЕ СИМВОЛ
?А
СИМВОЛ А: 5
```

Такой текст называется *протоколом диалога человека с машиной*.

Предписания вывода, предшествующие предписаниям ввода, подсказывают человеку, работающему с программой (пользователю), ввод каких значений он должен выполнить.

В языке предусмотрена специальная форма предписания ввода, объединяющая в себе

вывод и ввод. В этом случае предписание ввода записывается так:

ВВОД ПРИГЛ выражение: имена

или

ВВОД ТЕКСТА ПРИГЛ выражение: имя

Здесь после слова ПРИГЛ записывается выражение, значение которого будет напечатано перед исполнением ввода.

Например, в процедуре ПОДСЧЕТ вместо строк, отмеченных знаком +, можно записать

ВВОД ТЕКСТА ПРИГЛ «ВВЕДИТЕ ТЕКСТ»: Т

а вместо строк, отмеченных знаком * — ВВОД ТЕКСТА ПРИГЛ «ВВЕДИТЕ СИМВОЛ»: СИМВОЛ

Продолжим решение задачи о том, сколько раз заданная буква встречается в тексте.

```
ПРОЦ АЛФАВИТ_1 ()
  ВВОД ТЕКСТА ПРИГЛ "ВВЕДИТЕ ТЕКСТ": Т
  ВВОД ПРИГЛ "ВВЕДИТЕ ЧИСЛО СИМВОЛОВ": ЧИС
  ПОВТОР ЧИС
  ЦИКЛ
    ВВОД ТЕКСТА ПРИГЛ
      "ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ": СИМВОЛ
  СЧЕТЧИК:=0
  ДЛЯ НОМ ОТ 1 ДО #Т
  ЦИКЛ
    ЕСЛИ Т[НОМ]=СИМВОЛ
      ТО СЧЕТЧИК:=СЧЕТЧИК+1
  ВСЕ
  КЦ
  ВЫВОД: "СИМВОЛ ", СИМВОЛ, ": ", СЧЕТЧИК
  КЦ
КОНЕЦ
```

С помощью этой программы можно определить для каждого символа, сколько раз он входит в текст.

Как правило, ввод-вывод и действия, связанные с вычислениями, записывают в виде разных процедур. В этом случае становится легче вносить изменения в диалог (например, изменить текст приглашений) или в вычисления (например, искать число символов не во всем тексте, а в некоторой его части).

Попробуем выделить в отдельную процедуру все вычисления (это строки из процедуры АЛФАВИТ_1, отмеченные знаком *). Назовем эту процедуру СЧЕТ_НЕПРАВИЛЬНЫЙ. Тогда процедуру АЛФАВИТ_1 можно записать так:

```
ПРОЦ АЛФАВИТ_1 ()
  ВВОД ТЕКСТА ПРИГЛ "ВВЕДИТЕ ТЕКСТ": Т
  ВВОД ПРИГЛ "ВВЕДИТЕ ЧИСЛО СИМВОЛОВ": ЧИС
  ПОВТОР ЧИС
  ЦИКЛ
    ВВОД ТЕКСТА ПРИГЛ
      "ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ": СИМВОЛ
```

```

СЧЕТ НЕПРАВИЛЬНЫЙ()
Вывод: "СИМВОЛ ", СИМВОЛ, ": ", СЧЕТЧИК
КЦ
КОНЕЦ

```

Мы предполагаем, что теперь процедура АЛФАВИТ_1 выполнит ввод-вывод, а также вызовет процедуру СЧЕТ_НЕПРАВИЛЬНЫЙ, которая произведет необходимые вычисления.

Попробуем исполнить процедуру АЛФАВИТ_1. Для этого наберем АЛФАВИТ_1(). На экране появится запрос: ВВЕДИТЕ ТЕКСТ

?
Набираем:
СТРАНА

Следующий запрос:
ВВЕДИТЕ ЧИСЛО СИМВОЛОВ

?
Набираем:
3

После этого будет предложено:
ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ
?

Набираем:
А
Мы ждем результат, а получаем... сообщение об ошибке
НАДО ТЕКСТ (У ВАС — ПУСТО)
Что случилось? Мы ведь ожидали увидеть на экране следующее:
СИМВОЛ А: 2

Почему же не получился ожидаемый результат? Причина в следующем. Имена Т и СИМВОЛ известны только внутри процедуры АЛФАВИТ_1, а в процедуре СЧЕТ_НЕПРАВИЛЬНЫЙ их значения неизвестны (в РАПИРЕ все имена, которым не были присвоены значения, имеют значение ПУСТО)*.

Как же сделать, чтобы процедуры могли «передавать» друг другу значения? Этого можно добиться двумя способами. В одном из них роль таких «передатчиков» играют *параметры*. Рассмотрим, к примеру, процедуру СЧЕТ_ПОЧТИ_ПРАВИЛЬНЫЙ.

```

ПРОЦ СЧЕТ_ПОЧТИ_ПРАВИЛЬНЫЙ(СИМВОЛ, Т)
СЧЕТЧИК:=0
для ном от 1 до #Т
цикл
если т[ном]=СИМВОЛ
то СЧЕТЧИК:=СЧЕТЧИК+1
все
кц
конец

```

После имени процедуры в круглых скобках записаны имена. Их называют *формальными*

параметрами. Записывая вызов процедуры, надо указать их значения (*фактические параметры*). Например, можно вызвать

СЧЕТ_ПОЧТИ_ПРАВИЛЬНЫЙ («К», «КРОКОДИЛ»)

Фактическими параметрами могут быть имена, значения которых должны быть заданы до вызова процедуры, например:
СИМ:= «А»; Т:= «ЯМАХА»

СЧЕТ_ПОЧТИ_ПРАВИЛЬНЫЙ (СИМ, Т)

Обратите внимание, что имена формальных и фактических параметров могут совпадать.

Итак, мы передали информацию внутрь процедуры. Параметры, предназначенные для передачи информации внутрь процедуры, называются *входными*.

Исправим процедуру АЛФАВИТ_1:

```

ПРОЦ АЛФАВИТ_2()
ввод текста пригл "ВВЕДИТЕ ТЕКСТ":Т
ввод пригл "ВВЕДИТЕ ЧИСЛО СИМВОЛОВ":числ 41
повтор чис
цикл
ввод текста пригл
"ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ": символ
счет_почти_правильный(символ, т)
вывод: "СИМВОЛ ", символ, ": ", счетчик
кц
конец

```

Попробуем исполнить нашу модифицированную процедуру. Начало будет таким же, как и при исполнении процедуры АЛФАВИТ_1. А вот результат — не намного лучше:

СИМВОЛ А: ПУСТО

Счет выполнен, но результат не напечатан. Почему? Значение имени СЧЕТЧИК осталось «известным» только внутри процедуры СЧЕТ_ПОЧТИ_ПРАВИЛЬНЫЙ. Теперь нам надо передать информацию (значение имени СЧЕТЧИК) из процедуры СЧЕТ_ПОЧТИ_ПРАВИЛЬНЫЙ в процедуру АЛФАВИТ_2. В таких случаях используют параметры, которые называются *возвратными* (их значения передаются в процедуру при ее вызове и «возвращаются» после исполнения процедуры). В описании процедуры и в предписании ее вызова перед именами таких параметров ставится знак <=. В предписании вызова процедуры возвратным параметром всегда является имя. После исполнения процедуры это имя получит новое значение.

Рассмотрим процедуру СЧЕТ.

```

ПРОЦ СЧЕТ(СИМВОЛ, Т, <= СЧЕТЧИК)
СЧЕТЧИК:=0
для ном от 1 до #Т
цикл
если т[ном]=СИМВОЛ
то СЧЕТЧИК:=СЧЕТЧИК+1

```

* В других языках программирования возможны другие ситуации.

ВСЕ
КЦ
КОНЕЦ

Параметры СИМВОЛ и Т — входные, а параметр СЧЕТЧИК — возвратный. После исполнения предписания СЧЕТ («К», «КУКАРЕКУ», <=КОЛИЧЕСТВО) значением имени КОЛИЧЕСТВО станет число 3.

После исполнения программы Т := «ПРОГРАММИРОВАНИЕ»

СЧЕТ («Р», Т, <=С1)

СЧЕТ («О», Т, <=С2)

СЧЕТ («Я», Т, <=С3)

таблица значений имен имеет вид:

Имя	Значение
Т	ПРОГРАММИРОВАНИЕ
С1	3
С2	2
С3	0

Теперь мы можем описать процедуру, которая будет определять, сколько раз в тексте встречается каждый из заданных символов.

ПРОЦ АЛФАВИТ_3()

ВВОД ТЕКСТА ПРИГЛ "ВВЕДИТЕ ТЕКСТ":ТЕКСТ
ВВОД ПРИГЛ "ВВЕДИТЕ ЧИСЛО СИМВОЛОВ":ЧИС
ПОВТОР ЧИС

ЦИКЛ

ВВОД ТЕКСТА ПРИГЛ

"ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ": ЗНАК

СЧЕТ(ЗНАК, ТЕКСТ, <=СЧ)

ВЫВОД: "СИМВОЛ ", ЗНАК, ":", СЧ

КЦ

КОНЕЦ

С помощью этой процедуры удобно определять число вхождений букв в один и тот же текст. Если необходимо выполнять вычисления для различных текстов, то ввод текста лучше выполнять внутри цикла.

Можно не заставлять человека считать число различных символов, частоты которых нужно определить. Вместо этого можно договориться, какой символ будет обозначать окончание подсчета. Возьмем в качестве такого символа *.

ПРОЦ ЧАСТОТЫ_1(ТЕКСТ)

ВВОД ТЕКСТА ПРИГЛ

"ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ": ЗНАК

ПОКА ЗНАК /= "*"

ЦИКЛ

СЧЕТ(ЗНАК, ТЕКСТ, <=ЧИСЛО)

ВЫВОД: "СИМВОЛ ", ЗНАК, ":", ЧИСЛО

ВВОД ТЕКСТА ПРИГЛ

"ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ": ЗНАК

КЦ

КОНЕЦ

При вызове этой процедуры необходимо вводить по одному символу, для которых должны быть выполнены вычисления. Как только будет введен символ *, работа процедуры будет закончена. А как быть в случае, когда * тоже нужно подсчитывать? В этом случае в качестве последнего символа надо взять любой символ, для которого не надо определять частоту.

Запишем решение этой задачи, используя еще один вариант предписания цикла (цикл с условием окончания).

ПРОЦ ЧАСТОТЫ_2(ТЕКСТ)

ЦИКЛ

ВВОД ТЕКСТА ПРИГЛ

"ВВЕДИТЕ ОЧЕРЕДНОЙ СИМВОЛ": ЗНАК

СЧЕТ(ЗНАК, ТЕКСТ, <=ЧИСЛО)

ВЫВОД: "СИМВОЛ ", ЗНАК, ":", ЧИСЛО

КЦ ПО ЗНАК="*"

КОНЕЦ

Как выполняется такое предписание цикла? Здесь, в отличие от цикла ПОКА, сначала выполняется тело цикла, а потом проверяется условие, записанное после слова ПО. Выполнение цикла прекращается, когда условие истинно.

Если результатом процедуры является один объект, то вместо процедуры можно описать функцию. Перепишем процедуру СЧЕТ в виде функции с именем СКОЛЬКО.

ФУНК СКОЛЬКО(СИМВОЛ, Т)

СЧЕТЧИК:=0

ДЛЯ НОМ ОТ 1 ДО #Т

ЦИКЛ

ЕСЛИ Т[НОМ]=СИМВОЛ

ТО СЧЕТЧИК:=СЧЕТЧИК+1

ВСЕ

КЦ

ВОЗВРАТ СЧЕТЧИК

КОНЕЦ

Заголовок функции имеет вид ФУНК имя (параметры)

В функциях можно использовать только входные параметры. Роль возвратного параметра играет имя функции. В теле функции записываются предписания возврата, имеющие вид

ВОЗВРАТ выражение

Как только будет исполнено одно из предписаний возврата, выполнение функции прекращается, значением имени функции становится значение выражения из этого предписания возврата.

Вызов функции, в отличие от вызова процедуры, не является самостоятельным предписанием. Поэтому функции можно использовать только в выражениях.

Процедура ГЛАС определяет общее число гласных букв в тексте.

ПРОЦ ГЛАС (Т)

```
ГЛ_БУК:= СКОЛЬКО("А", Т) +
СКОЛЬКО("Е", Т) + СКОЛЬКО("И", Т) +
СКОЛЬКО("О", Т) + СКОЛЬКО("У", Т) +
СКОЛЬКО("Ы", Т) + СКОЛЬКО("З", Т) +
СКОЛЬКО("Ю", Т) + СКОЛЬКО("Я", Т)
ВЫВОД:"В ТЕКСТЕ", ГЛ_БУК, " ГЛАСНЫХ БУКВ"
КОНЕЦ
```

Задача 15. Описать функцию ПРОЦЕНТ_БУКВ(СИМВОЛ, ТЕКСТ), значением которой является процент числа вхождений данного символа в текст.

Задача 16. Описать процедуру ЗАМЕНА(СИМ_1, СИМ_2, <=ТЕКСТ, ЧИСЛО), выполняющую в тексте замену одного символа на другой и определяющую число выполненных замен.

Задача 17. Описать функцию ЧИСЛО_ЦЕП(ЦЕП, Т), значением которой является количество заданных цепочек символов в тексте. При выделении цепочек символ один и тот же символ исходного текста не должен рассматриваться более одного раза. Например:
ЧИСЛО_ЦЕП(«КУ», «КУКАРЕКУ») = 2
ЧИСЛО_ЦЕП(«КУКУ», «КУКУКУКУ») = 2

Задача 18. Описать процедуру ЗАМЕНА_ЦЕП(ЦЕП_1, ЦЕП_2, <=ТЕКСТ, <=ЧИСЛО), выполняющую в тексте замену одной цепочки символов на другую и определяющую число выполненных замен (цепочки символов могут быть разной длины!). Считаем, что после выполнения очередной замены ЦЕП_1 на ЦЕП_2 цепочка ЦЕП_2 не участвует в дальнейшем рассмотрении текста.

Указание. Значением выражения Т (НОМ:НОМ — 1) является пустой текст. Здесь НОМ принимает значения от 1 до (#Т+1) (в частности, значениями выражения Т[1:0] и Т(#Т+1:#Т) является пустой текст).

В Рапире предусмотрено большое число стандартных процедур и функций, которые можно вызывать в любой программе. Примерами стандартных функций являются:
ABS (X) — абсолютная величина X;
КОРЕНЬ (X) — корень квадратный из X;
ЦСЧ (X) — Целое Случайное Число из промежутка [1, X] при положительных X и из промежутка [X, — 1] при отрицательных X.

Здесь X — выражение, значением которого является число. Для функции ЦСЧ значением X должно быть целое число, отличное от нуля.

В заключение приведем программу для игры «Отгадайте число» для случая, когда машина «загадывает» число, а человек его отгадывает.

ПРОЦ УГАДАЙКА (X)

```
ЗАДУМКА:=ЦСЧ(X)
ВЫВОД:"Я ЗАДУМАЛА ЧИСЛО ОТ 1 ДО", X
ВЫВОД:"      ОТГАДАЙТЕ ЕГО"
СЧЕТЧИК:=0
ЦИКЛ
ВВОД ПРИГЛ "ВВЕДИТЕ ЧИСЛО":ЧИС
СЧЕТЧИК:=СЧЕТЧИК+1
ВЫБОР
  ПРИ ЗАДУМКА<ЧИС:
    ВЫВОД:"МОЕ ЧИСЛО МЕНЬШЕ"
  ПРИ ЗАДУМКА>ЧИС:
    ВЫВОД:"МОЕ ЧИСЛО БОЛЬШЕ"
ВСЕ
КЦ ПО ЗАДУМКА=ЧИС
ВЫВОД:"МОЛОДЕЦ!"
ВЫВОД:"ВЫ УГАДАЛИ С", СЧЕТЧИК, " ПОПЫТКИ"
КОНЕЦ
```

Решения задач.

Задача 15.

Решение 1. (с использованием функции СКОЛЬКО).

43

```
ФУНК ПРОЦЕНТ_БУКВ (СИМВОЛ, ТЕКСТ)
ЧИСЛО_БУКВ:=СКОЛЬКО(СИМВОЛ, ТЕКСТ)
ЕСЛИ #ТЕКСТ=0
  ТО ВОЗВРАТ "НЕ ОПРЕДЕЛЕНО"
ИНАЧЕ ВОЗВРАТ ЧИСЛО_БУКВ*100/#ТЕКСТ
ВСЕ
КОНЕЦ
```

Обратите внимание на то, каков будет результат функции, если рассматривается пустой текст.

Решение 2.

```
ФУНК ПРОЦЕНТ_БУКВ (СИМВОЛ, ТЕКСТ)
СЧЕТЧИК:=0
ДЛЯ НОМ ОТ 1 ДО #ТЕКСТ
ЦИКЛ
  ЕСЛИ ТЕКСТ[НОМ]=СИМВОЛ
    ТО СЧЕТЧИК:=СЧЕТЧИК+1
ВСЕ
КЦ
ЕСЛИ #ТЕКСТ=0
  ТО ВОЗВРАТ "НЕ ОПРЕДЕЛЕНО"
ИНАЧЕ ВОЗВРАТ СЧЕТЧИК*100/#ТЕКСТ
ВСЕ
КОНЕЦ
```

Задача 16.

```
ПРОЦ ЗАМЕНА(СИМ_1, СИМ_2, <=ТЕКСТ, <=ЧИСЛО)
ЧИСЛО:=0
ДЛЯ НОМ ОТ 1 ДО #ТЕКСТ
ЦИКЛ
  ЕСЛИ ТЕКСТ[НОМ]=СИМ_1
    ТО ТЕКСТ[НОМ]:=СИМ_2
    ЧИСЛО:=ЧИСЛО + 1
ВСЕ
КЦ
КОНЕЦ
```

Задача 17.

```

ФУНК ЧИСЛО_ЦЕП (ЦЕП,Т)
ЧИСЛО:=0
НОМ:=1
ПОКА НОМ <= #Т - #ЦЕП + 1
ЦИКЛ
ЕСЛИ Т[НОМ:НОМ + #ЦЕП - 1]=ЦЕП
ТО ЧИСЛО:=ЧИСЛО + 1
НОМ:=НОМ + #ЦЕП
ИНАЧЕ НОМ:=НОМ + 1
ВСЕ
КЦ
ВОЗВРАТ ЧИСЛО
КОНЕЦ
    
```

Задача 18.

```

ПРОЦ ЗАМЕНА_ЦЕП
(ЦЕП_1, ЦЕП_2, <=ТЕКСТ, <=ЧИСЛО)
ЧИСЛО:=0
НОМ:=1
ПОКА НОМ <= #ТЕКСТ - #ЦЕП_1 + 1
ЦИКЛ
ЕСЛИ ТЕКСТ[НОМ:НОМ + #ЦЕП_1 - 1]=ЦЕП_1
ТО ТЕКСТ[НОМ:НОМ + #ЦЕП_1 - 1]=ЦЕП_2
ЧИСЛО:=ЧИСЛО + 1
НОМ:=НОМ + #ЦЕП_2
ИНАЧЕ НОМ:=НОМ+1
ВСЕ
КЦ
КОНЕЦ
    
```

Задачи 17 и 18 лучше всего проверять с помощью наборов тестов, примеры которых приведены ниже.

Тексты	Цепочки	Примечание
КУКАРЕКУ	ПЕТУХ	заданных цепочек нет
ПРЕДМЕТ	ИНФОРМАТИКА	длина цепочки больше длины текста
КУКАРЕКУ	КУ	цепочка стоит в начале/конце текста
КРОКОДИЛ	КОД	цепочка стоит в середине текста

44

Текст	ЦЕП_1	ЦЕП_2	Примечание
РОДИТЕЛИ	МАМА	ПАПА	в тексте нет ЦЕП_1 ЦЕП_1 в начале текста длины цепочек одинаковы длины цепочек различны
МАЙОР ИВАНОВ	МАЙОР	КАПИТАН	
Я, ТЫ, ОН, ОНА	ТЫ	ВЫ	
МЫ —ОТРЯД	ОТРЯД	ДРУЖИНА	
АБАБАБАБАБ	А	АА	

Окончание следует

И. ПОЛИВАНЬИ,

Павлодарский педагогический институт

Исполнение Бейсик-программ

В статье, основанной на опыте проведения практических занятий со студентами, учителями школ, преподавателями техникумов и СПТУ, показано, как можно добиться хороших результатов в изучении основ программирования. В популярной книге Л. Пула «Работа на персональном компьютере» говорится, что «лучший способ научиться всему этому (программированию на Бейсике) — самостоятельно выполнить рассматриваемые операции». Действительно, результаты практических занятий свидетельствуют о том, что наиболее полно и глубоко язык познается после «ручного» исполнения программ.

Самостоятельно исполняя программу, ученик начинает осознавать «внутренние» процессы, происходящие в ЭВМ: этапы выполнения команд, обращение к ячейкам памяти, отображение информации на экран дисплея и т. д. Одновременно с этим вырабатываются очень полезные навыки алгоритмического мышления. Предлагаемый подход к изучению Бейсика позволяет:

- превратить процесс изучения языка в интересное и увлекательное занятие; доходчиво и достаточно наглядно продемонстрировать работу основных операторов; существенно облегчить учителю процесс проверки знаний и понимания учащимися основных идей программирования;
- приучить к тщательному рассмотрению программ с целью выявления ошибок, допущенных при программировании;

эффективно проводить занятия по программированию даже в условиях острого дефицита машинного времени.

Чтобы не ограничиваться только рамками школьного курса, излагаемый материал не увязан с алгоритмическим языком пробного учебного пособия; данная связь при необходимости может быть проведена очевидным образом.

Порядок исполнения программ

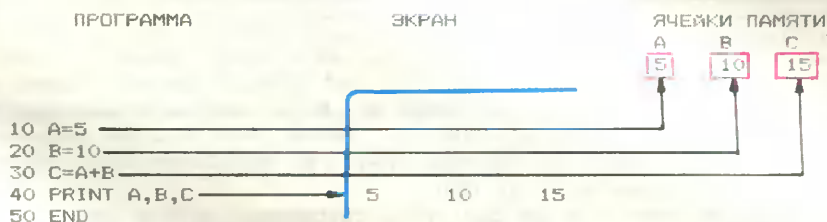
Предлагаемая методика исполнения Бейсик-программ отличается от уже известных тем, что, во-первых, она не требует постоянного вычерчивания громоздких таблиц, во-вторых, позволяет наглядно отображать все основные процессы, происходящие в ЭВМ (обращение к памяти, отображение информации на экран дисплея, передача управления и т. д.), и, в-третьих, максимально приближает протокол исполнения программы к тексту самой программы.

Поскольку описываемый подход к изучению Бейсика ориентирован в основном на неподготовленного учащегося, в ряде случаев пришлось поступиться строгостью определений. Как показывает опыт, строгое описание основных конструкций программирования труднодоступно для начинающего пользователя, поэтому изложение в статье опирается на терминологию, с одной стороны, понятную новичку, а с другой стороны, достаточно правильно отражающую реальные объекты и процессы, происходящие в ЭВМ.

Поэтому сразу условимся вместо терминов «переменная величина» и «идентификатор» (имя переменной величины) использовать понятия «ячейка памяти» и «имя ячейки памяти». Такие упрощенные формулировки преследуют исключительно практическую цель — облегчить понимание основных операторов и конструкций языка.

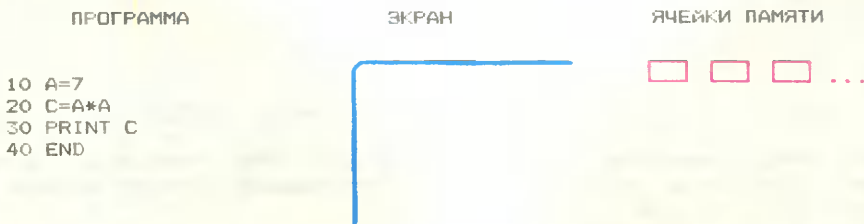
45

Процесс исполнения программ будет проиллюстрирован диаграммами, в основе которых текст исполняемой программы и объекты действия операторов Бейсика: фрагмент экрана дисплея и используемые ячейки памяти машины. Пример диаграммы — на рис. 1. Как видно из него, диаграмма состоит из трех самостоятельных частей: текста программы, экрана и набора ячеек памяти. С помощью стрелок мы будем указывать на объекты действия исполняемого оператора либо осуществлять передачу управления другому оператору программы.

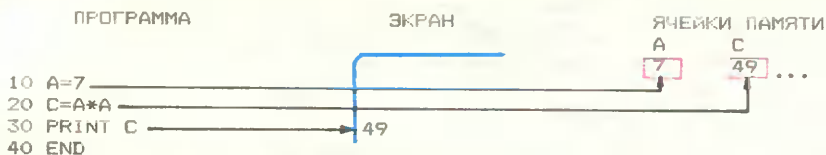


Правомерен вопрос: каким образом появляются в памяти ЭВМ необходимые нам ячейки?

Не вдаваясь в подробности, отметим, что в памяти вычислительной машины имеется участок, специально отведенный под хранение данных. Состоит этот участок из достаточно обширной последовательности ячеек памяти. Перед запуском программы каждая такая ячейка не имеет определенного имени и не содержит в себе конкретных значений. Этот факт можно отразить при предварительном оформлении диаграммы, включая в ее заголовок последовательность пустых, безымянных ячеек (рис. 2). Только после запуска программы, в процессе ее выполнения, происходит обращение к ячейкам, в результате чего



они получают соответствующие имена и значения. Таким образом, исполнив предыдущую программу, мы получим диаграмму на рис. 3.



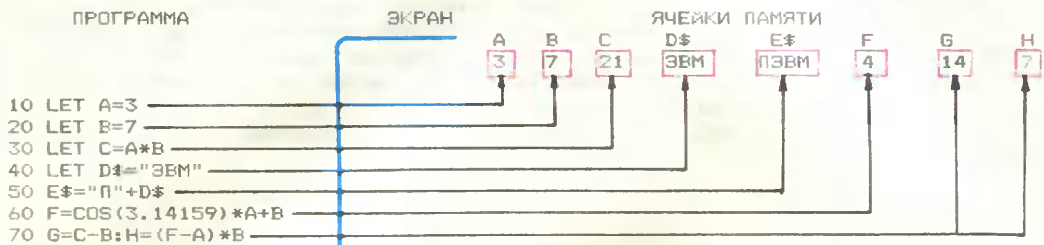
Данный пример хорошо иллюстрирует как предварительную подготовку к исполнению (оформление заголовка диаграммы и запись текста исходной программы), так и порядок исполнения (последовательное исполнение операторов, входящих в программу).

Прежде чем приступить к рассмотрению особенностей исполнения основных операторов, хотелось бы отметить, что предлагаемая статья предназначена для учителей, уже обладающих определенным опытом программирования на Бейсике. Излагаемый материал не претендует на исчерпывающую полноту в описании языка, поэтому при рассмотрении каждого оператора дается лишь краткая характеристика, раскрывающая особенности его реализации на ЭВМ и определяющая порядок исполнения.

Исполнение основных операторов

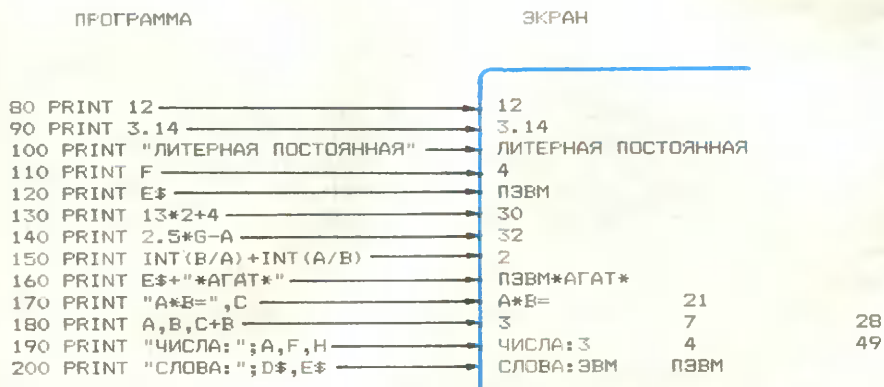
46

Оператор присваивания LET. Выполняя его, ЭВМ вычисляет значение выражения, стоящего справа от знака «=», а полученный результат заносит в ячейку памяти, имя которой указано слева от знака «=» (рис. 4).



Оператор печати PRINT. Выполняя его, ЭВМ может печатать на экране (рис. 5):

- постоянное значение (операторы 80, 100);
- содержимое указанной ячейки памяти (операторы 110, 120);
- значение выражения (операторы 130, 160);
- совокупность перечисленных выше элементов, отделенных друг от друга запятой или точкой с запятой (операторы 170, 200).



Оператор конца программы END. Встретив его, ЭВМ прекращает выполнение программы, а для человека, исполняющего программу, данный оператор является признаком завершения работы (рис. 6).

ПРОГРАММА

ЭКРАН

```

10 PRINT "БЕЙСИК - ЭТО"
20 PRINT "НЕСЛОЖНО"
30 END
  
```

БЕЙСИК - ЭТО
НЕСЛОЖНО

Оператор комментария REM является неисполняемым. Встретив в программе оператор REM, ЭВМ автоматически переходит к выполнению следующего по порядку оператора (рис. 7).

ПРОГРАММА

ЭКРАН

```

10 REM *****
20 REM * ВЫЧИСЛЕНИЕ ПЛОЩАДИ *
30 REM * КРУГА, R=3 CM *
40 REM *****
50 REM
60 PRINT "ПЛОЩАДЬ КРУГА=";3.1415*3^2
70 END
  
```

ПЛОЩАДЬ КРУГА=28.2735

Операторы ввода списка данных READ, DATA. Выполняя оператор READ, ЭВМ записывает постоянные значения, собранные в операторе DATA, в ячейки памяти с указанными в операторе READ именами (рис. 8). 47

ПРОГРАММА

ЭКРАН

```

10 DATA 7,3,2,4,6,КОНЕЦ
20 READ A,B,C,D
30 READ E,F$
40 PRINT C;B;D;E;A;F$
50 END
  
```

2 3 4 6 7 КОНЕЦ

ЯЧЕЙКИ ПАМЯТИ
A 7 B 3 C 2 D 4 E 6 F\$ КОНЕЦ

Оператор ввода значений с клавиатуры INPUT. Выполняя его, ЭВМ в качестве приглашения для ввода информации выдает на экран знак «?». В ответ человек должен с помощью клавиатуры набрать необходимые значения и нажать клавишу «ввод», после чего набранная информация записывается в ячейки памяти с указанным в операторе INPUT именами (рис. 9).

ПРОГРАММА

ЭКРАН

```

10 PRINT "ВВЕДИ ДЛИНЫ КАТЕТОВ"
20 INPUT A,B
30 C=SQR(A^2+B^2)
40 PRINT "ГИПОТЕНУЗА=";C
50 END
  
```

ВВЕДИ ДЛИНЫ КАТЕТОВ
? 3,4
ГИПОТЕНУЗА= 5

ЯЧЕЙКИ ПАМЯТИ
A 3 B 4 C 5

Оператор определения функции DEF. Выполняя его, ЭВМ на время работы программы пополняет набор стандартных функций (SIN, COS, EXP,...) новой функцией, определенной пользователем (рис. 10).

ПРОГРАММА

ЭКРАН

```

10 DEF FNF(X)=X^2+2*X+7
20 PRINT FNF(3)
30 PRINT FNF(1)
40 PRINT FNF(0)
50 END
  
```

22
10
7

ПРОГРАММА

ЭКРАН

ЯЧЕЙКИ ПАМЯТИ

```
10 DIM A(1),B(3)
...
```



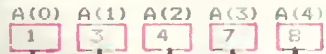
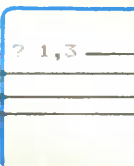
Оператор определения массива DIM. Выполняя его, ЭВМ отводит в памяти машины участок (или массив), состоящий из указанного количества ячеек. Каждая такая ячейка получает имя, зависящее от места ее расположения в массиве. Нумерация ячеек массива начинается с нуля. С помощью одного оператора DIM можно определять несколько массивов (рис. 11). Заполнение ячеек массива производится аналогично заполнению обычных ячеек памяти, т. е. с помощью операторов LET, READ, INPUT (рис. 12).

ПРОГРАММА

ЭКРАН

ЯЧЕЙКИ ПАМЯТИ

```
10 DIM A(4)
20 INPUT A(0),A(1)
30 A(2)=A(0)+A(1)
40 A(3)=A(1)+A(2)
50 READ A(4)
60 DATA B
70 END
```



48

Оператор безусловного перехода GOTO. Выполняя его, ЭВМ передает управление оператору программы, номер строки которого указан после ключевого слова GOTO (рис. 13).

ПРОГРАММА

ЭКРАН

```
10 PRINT "ПРИГОВОР:"
20 GOTO 50
30 PRINT "КАЗНИТЬ,"
40 PRINT "НЕЛЬЗЯ"
50 PRINT "ПОМИЛОВАТЬ."
60 PRINT "*****"
70 END
```

```
ПРИГОВОР:
ПОМИЛОВАТЬ.
*****
```

Оператор условного перехода IF—THEN (—ELSE). Выполняя оператор IF—THEN, ЭВМ в первую очередь проверяет, является ли верным условие, записанное после ключевого слова IF (рис. 14). Если условие является верным, то управление передается оператору, расположенному после ключевого слова THEN.

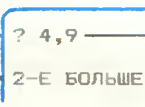
Если условие не является верным, то управление передается следующему по порядку оператору программы.

ПРОГРАММА

ЭКРАН

ЯЧЕЙКИ ПАМЯТИ

```
10 INPUT A,B
20 IF A>B THEN PRINT "1-Е БОЛЬШЕ"
30 IF A<B THEN PRINT "2-Е БОЛЬШЕ"
40 END
```



При исполнении программы результат проверки условия будем отмечать стрелками:
 «→» условие является верным;
 «←» условие не является верным.

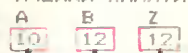
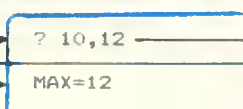
Выполняя второй вариант, оператор IF—THEN—ELSE, ЭВМ поступает аналогично предыдущему случаю с той разницей, что если проверяемое условие не является верным, то управление передается оператору, расположенному после ключевого слова ELSE (рис. 15).

ПРОГРАММА

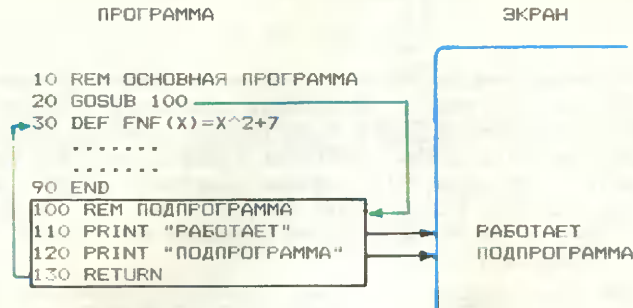
ЭКРАН

ЯЧЕЙКИ ПАМЯТИ

```
10 INPUT A,B
20 IF A>B THEN Z=A ELSE Z=B
30 PRINT "MAX=";Z
40 END
```

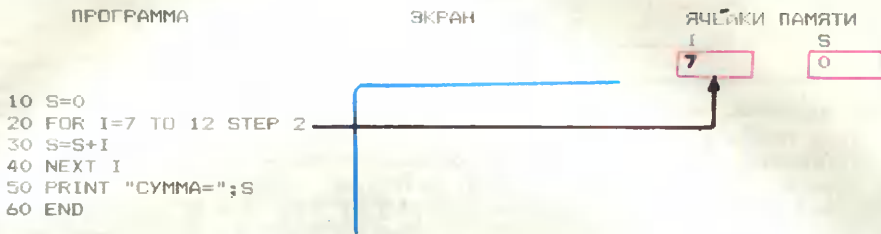


Оператор вызова подпрограммы GOSUB, оператор возврата RETURN. Выполняя оператор GOSUB, ЭВМ (как и в случае оператора GOTO) передает управление оператору, номер строки которого указан после ключевого слова GOSUB (т. е. осуществляет переход на подпрограмму). Выполняя же оператор RETURN, ЭВМ возвращает управление из подпрограммы в основную программу, причем именно в то ее место, откуда был осуществлен вызов (рис. 16). «Запоминанием» этого места оператор GOSUB отличается от оператора GOTO.

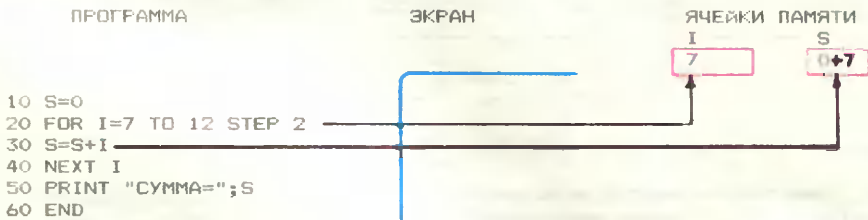


Операторы цикла FOR—TO, NEXT. Выполняя оператор заголовка цикла FOR—TO при первом проходе (рис. 17), ЭВМ резервирует ячейку памяти с указанным после ключевого слова FOR именем в качестве так называемого счетчика (в примере ячейка I) и заносит в нее первоначальное значение (в примере 7) (для большей наглядности несущественная стрелка, отражающая присваивание $S = 0$, опущена).

49



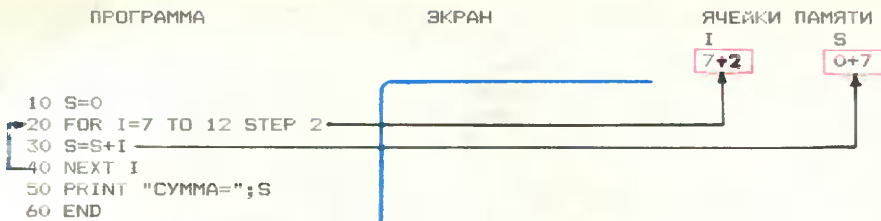
Оператор FOR указывает на начало циклического участка, а оператор NEXT — на его конец. Последовательность операторов, расположенных между ними, принято называть телом цикла (в нашем примере это единственный оператор 30 $S = S + I$). После первого выполнения оператора FOR ЭВМ приступает к выполнению операторов, образующих тело цикла (рис. 18).



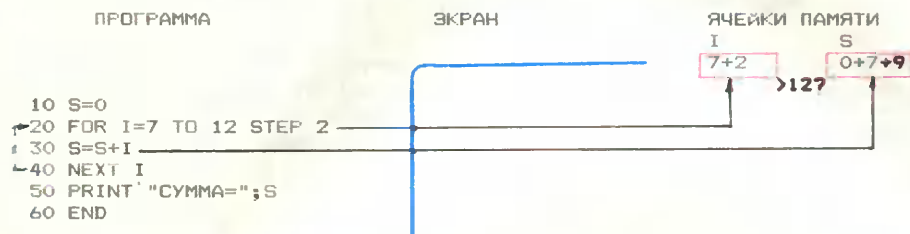
Выполняя оператор NEXT, ЭВМ возвращает управление на соответствующий оператор FOR—TO (в примере оператор 20), увеличивая при этом содержимое ячейки счетчика на величину шага (в примере шаг равен 2). Если ключевое слово STEP опущено, величина шага полагается равной 1. Таким образом, после исполнения оператора NEXT получим рис. 19.

Запись $0+7$ означает, что в ячейке S теперь содержится значение 7; первоначально было выполнено присваивание $S = 0$, затем $S = S + I$ (к содержимому ячейки S добавили содержимое ячейки I и полученный результат опять занесли в ячейку S). Аналогично запись $7+2$ означает, что в ячейке I теперь содержится значение 9.

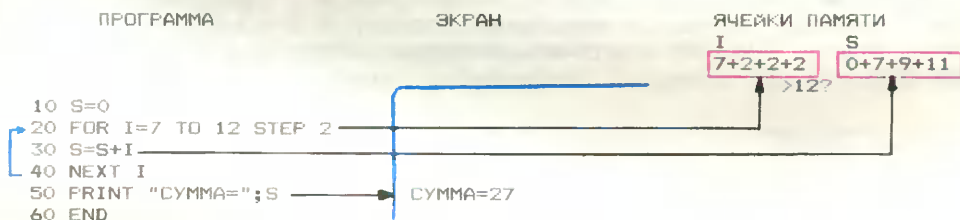
Выполняя оператор FOR—TO при каждом новом проходе, ЭВМ сравнивает текущее



содержимое ячейки счетчика с максимально возможным значением (в нашем примере оно равно 12). Если текущее значение больше, управление передается за пределы цикла, т. е. оператору, следующему за оператором NEXT. Если текущее значение меньше, выполняются операторы, входящие в тело цикла. Таким образом, в рассматриваемом нами примере перед повторным исполнением оператора FOR текущее содержимое ячейки счетчика 9, а его максимально возможное значение 12. Исполняя оператор, мы сравниваем эти значения. Результат сравнения говорит о том, что мы должны перейти к исполнению тела цикла, т. е. к оператору 30 $S = S + I$ (рис. 20).

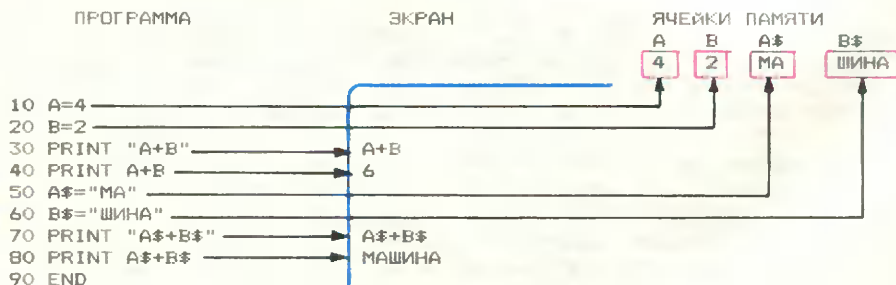


Далее, по аналогии, опять увеличиваем значение счетчика, возвращаемся к проверке, исполняем тело цикла и т. д. до тех пор, пока текущее содержимое ячейки счетчика не превысит максимально возможного значения. В нашем примере очередное увеличение счетчика на величину шага даст значение 11. В результате проверки ($11 > 12?$) мы опять вернемся в тело цикла и выполним оператор 30 $S = S + I$ ($16 + 11 = 27$). Нарастив счетчик в очередной раз, мы получим значение 13. После проверки ($13 > 12?$) управление теперь будет передано за пределы цикла (в примере оператору 50 PRINT...) (рис. 21).



Примеры исполнения программ

В качестве примеров рассмотрим исполнение программ, взятых из методического пособия по преподаванию курса «Основы информатики и вычислительной техники» (рис. 22—24).



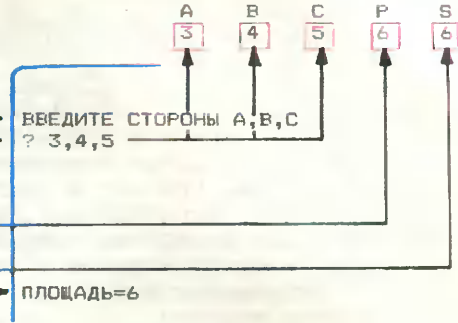
ПРОГРАММА

ЭКРАН

ЯЧЕЙКИ ПАМЯТИ

```

10 REM *ПЛОЩ. ТРЕУГ.*
20 PRINT "ВВЕДИТЕ СТОРОНЫ А,В,С"
30 INPUT A,В,С
40 REM *А,В,С - СТОРОНЫ ТРЕУГ.*
50 IF (А>В+С) OR (В>А+С) OR (С>А+В) THEN 20
60 REM *Р - ПОЛУПЕРИМЕТР*
70 Р=(А+В+С)/2
80 REM *ФОРМУЛА ГЕРОНА*
90 S=SQR(Р*(Р-А)*(Р-В)*(Р-С))
100 PRINT "ПЛОЩАДЬ=";S
110 END
    
```



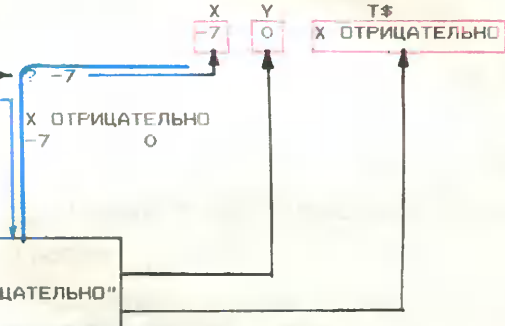
ПРОГРАММА

ЭКРАН

ЯЧЕЙКИ ПАМЯТИ

```

90 INPUT X
100 IF X>0 THEN GOSUB 200 ELSE GOSUB 300
110 PRINT T$
120 PRINT X,Y
130 END
200 T$="X ПОЛОЖИТЕЛЬНО"
210 Y=X*X
220 RETURN
300 REM
310 Y=0
320 IF X=0 THEN T4="X=0" ELSE T$="X ОТРИЦАТЕЛЬНО"
330 RETURN
    
```



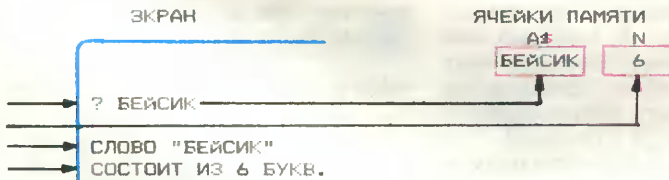
Методические рекомендации

Наилучших результатов в изучении языка программирования с помощью излагаемой методики можно достичь путем медленного, но основательного и последовательного освоения основных операторов и конструкций языка.

На практических занятиях при рассмотрении очередного оператора автор придерживался следующего порядка работы.

Задание 1. Исполнить предлагаемую программу. (Программа заранее подготавливается преподавателем, при этом в текст программы включаются различные варианты использования рассматриваемого оператора.)

Задание 2. По фрагменту диаграммы восстановить текст исходной программы (рис. 25).



Задание 3. Написать и исполнить программу для решения конкретной задачи.

Такое чередование заданий имеет преимущества психологического и методического порядка, так как позволяет внести разнообразие в подачу материала и использовать разные виды работ для его закрепления.

В заключение — ряд замечаний, которые могут оказаться полезными при организации занятий:

для удобства исполнения желательно каждый оператор программы записывать с новой строки;

очень важным является сам процесс исполнения программы — наглядное и простое прослеживание последовательности действий, а не «плакатная» демонстрация уже исполненных программ;

необходимо обратить внимание учащихся на тот факт, что программа в ЭВМ начинает выполняться не сама по себе, а только по команде человека (поэтому целесообразно каждый раз в конце исходной программы добавлять команду RUN и только после этого приступить к исполнению программы);

при подборе и составлении программ, рассчитанных на исполнение, не стоит стремиться к сложным и запутанным алгоритмам (предпочтение необходимо отдавать наиболее интересным задачам, взятым из реальной жизни);

для упрощения исполнения программы несущественные стрелки можно опускать, оставляя лишь те, которые отвечают за передачу управления;

опыт показывает, что описанный подход к изучению языка позволяет обучающимся очень быстро и легко «схватить» основные положения и идеи программирования, после чего им становится скучно исполнять то, что «и так понятно». Поэтому стоит не слишком затягивать с исполнением, а переходить к обычному составлению программ, как только такая потребность возникает.

Б. БЕЛОВ, А. БЕЛОВА,
г. Ульяновск

Активизация работы учащихся при изучении табличных величин и массивов

Рекомендуемые программы ОИВТ предусматривают изучение понятий табличных величин и массивов, умение работать с ними при составлении алгоритмов.

Материал, предлагаемый в данной статье, апробирован на занятиях в Ульяновском электромеханическом техникуме при изучении понятий «таблица» и «массив»; при выдаче заданий на программирование по предмету «Программирование и алгоритмические языки».

Мнение психологов о том, что усвоение знаний и формирование умений происходят более успешно в процессе активной практической деятельности учащихся, подтверждается нашим опытом работы с описываемой ниже системой задач. К этим задачам разработана серия таблиц. Этот комплект способствует формированию глубоких и прочных знаний и отвечает следующим методическим требованиям.

1. Содержание таблиц знакомо учащимся из повседневной жизни, в частности в него могут быть включены игры.

2. Содержание задач носит реальный характер, т. е. учащимся приходится решать такие задачи в жизни.

Плакаты должны быть красочно исполнены, с необходимыми пояснениями.

Комплект задач и таблиц должен постепенно дополняться и расширяться преподавателем, а использование его должно производиться с учетом уровня подготовки учащихся.

Предлагаемые задачи могут быть решены и другими способами с записью решений на алгоритмическом языке, но с применением других его конструкций. Все таблицы описаны строкой со служебным словом АЛГ. Решения задач приведены в виде фрагментов алгоритмов.

Таблица «Расписание занятий» состоит из четырех столбцов и шести строк. Первая строка (номера пар занятий) и первый столбец (обозначения дней недели) в массив не входят и в дальнейшем не используются. Элементы массива имеют значения сокращенных наименований предметов, состоящих из трех символов. Например: ФИЗ — физика, Ф. В — физвоспитание, ОХТ — охрана труда и т. д. (рис. 1).

	1	2	3	4
П	ИНФ	ГЕО	Ф. В	
В	ХИМ	ИСТ	ОХТ	
С	ФИЗ	ТОЗ	БИО	
Ч	ИНЯ	ЛИТ	ХИМ	
П	Ф. В	МАТ	РУС	
С	НВП	ФИЗ	ИСТ	

Таблица «Турнир шахматистов» состоит из пяти строк и шести столбцов. Первый столбец (номера шахматистов) и первая строка (номера столбцов) в массив не вхо-

дят и в дальнейшем не используются. Шестой столбец называется «ОЧКИ». Элементы массива могут принимать значения 0, 1 или 2 (рис. 2).

	1	2	3	4	5	6
1	0	0	1	2	2	
2	2	0	1	1	1	
3	1	1	0	2	0	
4	0	1	0	0	2	
5	0	1	2	0	0	

Таблица «Морской бой» состоит из десяти строк и десяти столбцов. Первый столбец (номера строк) и первая строка (номера столбцов) в массив не входят и в дальнейшем не используются. Элементы массива могут принимать значения 0 и 1. Запись 1 в клетку означает нахождение в ней корабля или его части (рис. 3).

	1	2	3	4	5	6	7	8	9	10
1										1
2		1	1							
3										1
4		1	1	1						1
5										
6				1	1	1				1
7										
8		1	1			1	1	1		1
9										
10										1

1. Задачи, использующие таблицу «Расписание занятий».

Таблица описана следующей строкой:
АЛГ РАБОТА С РАСП
(ЛИТ ТАБ РАСП [1:6, 1:4])

Задача 1.1. Несколькими командами перенести первую пару занятий на четвертую в понедельник.

Решение 1:
РАСП[1,4]:=«ИНФ»
РАСП[1,1]:=« »

Решение 2:
РАСП[1,4]:=РАСП[1,1]
РАСП[1,1]:=« »

Второй ответ предпочтительнее, так как он не зависит от конкретного предмета.

Задача 1.2. Отменить одним циклом занятия в субботу в связи с проведением всесоюзного субботника.

Решение:
M:=6; K:=1
ПОКА K<4

НЦ
РАСП[M,K]:=« »

K:=K+1

КЦ

Задача 1.3. Одним циклом исправить расписание так, чтобы в течение всей недели третьими парами были занятия по информатике.

Решение:
M:=1; K:=3
ПОКА M<=6

НЦ
РАСП[M,K]:=«ИНФ»

M:=M+1

КЦ

2. Задачи, использующие таблицу «Турнир шахматистов».

Таблица описана следующей строкой:
АЛГ ТУРНИР (ЦЕЛ ТАБ ТУР[1:5,1:6])
Выигрыш — 2, ничья — 1, проигрыш — 0 очков.

Задача 2.1. Оператор ЭВМ допустил ошибку и неправильно занес на табло (у нас таблица) результат игры четвертого и пятого участников. Выиграл пятый участник, а не четвертый. Как оператору исправить свою ошибку?

Решение 1:
ТУР[5,4]:=2
ТУР[4,5]:=0

Решение 2:
ТУР[5,4]:=ТУР[4,5]
ТУР[4,5]:=0

Задача 2.2. Подсчитать количество очков у второго участника и записать эту сумму в шестой столбец второму участнику. Здесь столбец № 6 можно назвать «очки».

Решение:
M:=2; K:=1; S:=0
ПОКА K<=5

НЦ
S:=S+ТУР[M,K]
K:=K+1

КЦ
ТУР[2,6]:=S

Задача 2.3. Подсчитать сумму очков у каждого участника и записать их в столбец № 6.

Решение:
M:=1; K:=1; S:=0
ПОКА M<=5

НЦ
ПОКА K<=5

НЦ
S:=S+ТУР[M,K]
K:=K+1

КЦ
ТУР [M,6] := S; S := 0; K := 1; M := M + 1

КЦ

Работу с данной таблицей можно продолжить, если ставить задачи посложнее. Например: ввести седьмой столбец в таблице под названием «МЕСТО». (Имеется в виду место, занятое участником в турнире). Заполнить седьмой столбец при помощи одного фрагмента алгоритма.

3. Задачи, использующие таблицу «Морской бой».

Таблица описана следующей строкой:
АЛГ ИГРА (ЦЕЛ ТАБ МБ[1:10,1:10])

Попадание — запись 0 вместо 1.

Сначала преподавателю необходимо договориться с учащимися об условиях игры: попадание в корабль — запись числа 0 в клетку, которую занимает корабль или его часть; считать корабль потопленным хотя бы при одном попадании.

Задача 3.1. Потопить все «синие» корабли.

Решение:

M := 1; K := 10

ПОКА M ≤ 10

НЦ

МБ[M,K] := 0

M := M + 1

КЦ

Задача 3.2. Потопить все «красные» корабли.

Решение 1:

M := 1; K := 1

ПОКА M ≤ 10

НЦ

МБ[M,K] := 0

M := M + 1; K := K + 1

КЦ

Решение 2:

M := 2; K := 2

ПОКА M ≤ 8

НЦ

МБ[M,K] := 0

M := M + 2

K := K + 2

КЦ

Второе решение предпочтительнее, так как уменьшает количество «выстрелов» до четырех вместо десяти в решении 1.

Задача 3.3. При проведении учений всегда имеется посредник. Ввести в театр военных действий корабль «зеленых» — посредника.

Решение:

МБ[8,2] := 1

МБ[8,3] := 1

Задачи к последним двум плакатам успешно использовались для программирования на языках Фокал и Бейсик и выполнялись на БК-0010 и ПЭВМ «Агат».

Вниманию организаций-пользователей персональных компьютеров класса IBM PC

Вы можете приобрести по договорным ценам (не выше государственных)

обучающие системы:

обучение языкам BASIC, TURBOPROLOG, C;
ознакомление с системами MSDOS, DBASE, PARADOX.

Возможна специализированная разработка программ по заказу.

Наш адрес: 101000, Москва, Центр,
ул. Богдана Хмельницкого, 6/8, МОС ВОИР.
Тел. 923-30-58 (с 9.00 до 18.00).

Экспериментальный центр

«А Л Г О»

Мы готовы приступить к сотрудничеству с Вами немедленно!

А. ГРИЦЕНКО,
зав. кабинетом ОИВТ Алтайского КИУУ

Система подготовки текстов для КУВТ-86

Непременной принадлежностью любой современной ЭВМ является экранный редактор текстов. Это программа, предназначенная для набора на экране дисплея текстов и записи их на внешний носитель (диск или магнитную ленту).

Примеры экранных редакторов текстов для ДВК-2М — K52, SCREEN, EDIK, NED, ED.

С помощью набираемых с клавиатуры команд можно редактировать уже набранный текст: изменять в нем отдельные буквы, слова, строки, переставлять абзацы, удалять или заменять часть текста и т. д. На обычной пишущей машинке такие операции либо сильно затруднены (исправление букв), либо просто невозможны и требуют полной перепечатки текста.

Набранный таким образом текст можно записать на диск или ленту и хранить там длительное время. Такая запись называется текстовым файлом.

Так можно создавать библиотеки документов на дисках. На один стандартный диск входит, как правило, не менее 200 страниц машинописного текста.

Записанный на диск документ можно опять ввести в память ЭВМ. Если требуется, то в него можно внести необходимые изменения и поправки и записать на диск в измененном (т. е. отредактированном) виде. Можно напечатать его на печатающем устройстве ЭВМ в любом количестве экземпляров.

Именно такова современная технология подготовки, хранения и размножения документов. Особенно широко она используется в газетном и книгоиздательском деле, т. е. там, где требуется большая оперативность в подготовке и редактировании текстов.

К сожалению, в нашей стране эта техно-

логия пока не получила широкого распространения. Мы сильно отстали от мирового уровня, и этот разрыв необходимо как можно скорее ликвидировать. Необходимо обучить основам этой технологии широкие слои населения, что является одной из прямых задач курса ОИВТ в средней школе.

К величайшему сожалению, даже в учебных заведениях, имеющих КУВТы, экранные редакторы практически не изучаются. Причина тому — отсутствие соответствующего программного обеспечения, методическая неразработанность темы, отсутствие методической литературы.

Перечисленные обстоятельства заставили автора попытаться создать систему подготовки текстов для самой распространенной модели учебного дисплейного класса КУВТ-86. Работа над ней была завершена в марте 1988 г.

Экранный редактор текстов представляет собой программу на языке машинных команд, записанную на диск. Эта программа может пересылаться и работать в ЭВМ БК-0010 на рабочих местах учеников.

После того как экранный редактор текстов начнет работу на БК-0010, можно считать в его память любой текстовый файл, хранящийся на диске. Можно также создать новый текстовый файл.

После завершения работы с текстовым файлом можно записать созданный или отредактированный файл из памяти БК-0010 на диск.

Учитель имеет возможность распечатать любой из созданных таким образом текстовых файлов на печатающем устройстве.

В состав системы входят:
EDASPM.BIN — экранный редактор текстов для БК-0010;

CONLST.SAV — программа распечатки текстовых файлов на печатающем устройстве;

EDASP.LST — текстовый файл с описанием редактора EDASP;

набор учебных текстовых файлов, предназначенных для обучения работе с текстовым редактором.

Экранный редактор текстов EDASP8 был создан в 1987 г. программистами из Зеленограда С. А. Фоменко, А. В. Гречишкиным и П. В. Успенским, которые подписывают свои программы групповым псевдонимом ASP согр. Ими же создано описание экранного редактора EDASP8, содержащееся в текстовом файле EDASP8.LST.

Созданная ими версия экранного редактора была рассчитана лишь на работу в автономном БК-0010 и предусматривала ввод и вывод текстовых файлов только на магнитофон. Поэтому ее пришлось переделать так, чтобы заменить ввод-вывод на магнитофон вводом-выводом на диск терминального класса. Эти работы были выполнены О. О. Войцеховичем при участии автора статьи.

Программа CONLST была создана О. О. Войцеховичем.

Учебные файлы были подготовлены автором статьи.

56 Основы работы с экранным редактором EDASP

Программа EDASPM обычным порядком пересылается с диска ДВК-2М в БК-0010 и автоматически запускается. После запуска программа очищает экран и выдает вопрос:

Считать файл? (1)

Если считать файл не надо, т. е. работа с экранным редактором будет представлять собой создание какого-либо нового текстового файла, то надо ответить Н (нет) и нажать клавишу ВВОД.

Если же надо считать с диска какой-либо ранее созданный текстовый файл, то нужно ответить Д (да) и нажать клавишу ВВОД (далее нажатие клавиши ВВОД особо оговариваться не будет). После этого на экране появится вопрос:

Имя файла? (2)

В ответ следует ввести имя нужного файла. Произойдет считывание файла с дисковода ДВК-2М. Если файл успешно считан, то на экране появятся первые 24 строки содержащегося в нем текста. Если же при чтении произошла ошибка или файла с этим именем на диске не оказалось, то на экране появится вопрос:

Повторить чтение?

Если ответить Д, то будет опять выдан запрос (1) и т. д.

После чтения файла с диска или отказа от такого чтения редактор текстов готов к работе. На экране появится курсор. Положение курсора на экране можно менять с помощью клавиш со стрелками. При попытке сместить курсор вверх или вниз за пределы экрана произойдет сдвиг всего текста на экране на строку вниз или вверх.

Если нажать на какую-либо символьную клавишу, то соответствующий символ будет занесен в позицию, в которой стоял курсор, а курсор сдвинется на одну позицию вправо. Используя это, можно набирать на экране любой текст. Перевод курсора на другую строку осуществляется нажатием клавиши ВВОД. При этом автоматически происходит создание пустой строки под предыдущей. Попытка сдвинуть курсор вверх или вниз за пределы существующего текста вызовет звуковой сигнал и игнорирование данной команды.

Если курсор стоит на уже набранном тексте, то с помощью клавиш с широкими стрелками (влево и вправо) можно сдвигать соответственно влево и вправо текст, находящийся в данной строке правее курсора.

У EDASP есть два режима работы — вставки и замены. При запуске редактора автоматически устанавливается режим вставки. В нем набор на клавиатуре любого символа автоматически приводит к сдвигу части строки справа от курсора на одну позицию вправо. В режиме замены набор на клавиатуре любого символа автоматически приводит к замене им символа, который ранее стоял в позиции курсора; сдвига текста не происходит.

Кроме того, EDASP имеет достаточно широкий набор команд, предназначенных для облегчения выполнения часто встречающихся операций. Рассмотрим их.

Команды экранного редактора

На БК-0010 последних выпусков устанавливается кнопочная клавиатура, которая несколько беднее, чем пленочная, устанавливавшаяся ранее. Поэтому некоторые команды экранного редактора на этих клавиатурах набираются по-разному, хотя в результате в БК вводятся одни и те же коды.

Везде в дальнейшем запись типа СУ/А означает: «нажать на клавишу СУ и, не отпуская ее, нажать клавишу А».

Команды экранного редактора, набираемые с помощью клавиши СУ.

СУ/Е — перенос курсора в начало следующего справа слова текста.

СУ/Д — перенос курсора в начало следующего слева слова текста.

СУ/К — преобразование всех букв, стоя-

щих справа от курсора до ближайшего пробела, из строчных в заглавные.

СУ/И — то же, но из заглавных в строчные.

СУ/] (на пленочной клавиатуре также клавиша ↗) — удаление строки, в которой стоит курсор. Эта команда используется для удаления из текста ненужных строк.

СУ/Ч (на пленочной клавиатуре также клавиша ↘) — раздвижка строк, т. е. сдвиг на строку вниз текста, находящегося ниже строки, где стоит курсор. В строку экрана, где стоит курсор, вставляется пустая строка. Эта команда используется, если нужно вставить в текст дополнительные строки.

СУ/Ъ (на пленочной клавиатуре также клавиша ↙) — установка метки; редактор запоминает номер строки, где стоит курсор. С помощью особой команды (см. ниже) можно автоматически вернуть курсор в эту строку. На экране метка никак не индицируется. Установка новой метки автоматически отменяет установленную ранее. При начале работы редактора метка автоматически устанавливается на первую строку текста.

СУ/Э (на пленочной клавиатуре также клавиша ↘) — поиск заданного контекста (см. ниже).

СУ/R (на пленочной клавиатуре также клавиша ↵) — команда «Префикс»*. Используется для задания более сложных команд.

Везде в дальнейшем запись типа ПР СУ/А означает последовательную выдачу команды «Префикс» и команды СУ/А (см. выше).

Команды экранного редактора, набираемые с помощью команды «Префикс».

ПР↑ — сдвиг курсора на 20 строк вверх.

ПР↓ — сдвиг курсора на 20 строк вниз.

ПР→ — сдвиг курсора в конец текущей строки.

ПР← — сдвиг курсора в начало текущей строки.

ПР Н — сдвиг курсора в начало текста. Курсор автоматически помещается в первую строку редактируемого текста.

ПР К — сдвиг курсора в конец текста. Курсор автоматически помещается в последнюю строку редактируемого текста.

ПР У — удаление части текста. Удаляется текст начиная от строки, где в последний раз устанавливалась метка, и кончая строкой, где стоит курсор. После этой команды редактор задает вопрос: «Удалить, Вы уверены?». Удаление будет произведено, только если на этот вопрос будет дан ответ Д. В противном случае удаление

произведено не будет. Эта команда используется для удаления из редактируемого текста больших его частей. Того же можно добиться, если последовательно удалять по одной строке, но это требует больше времени.

ПР R — переключение режима — со вставки на замену и обратно. После этой команды редактор задает вопрос: «Режим замены?» Если ответить на этот вопрос Д, то редактор начнет работать в режиме замены. Любой другой ответ приведет к установлению режима вставки.

ПР Д — задание длины строки. После этой команды редактор спрашивает длину строки. В ответ на это надо задать желаемую длину строки и нажать на клавишу ВВОД. После выполнения этой команды любая попытка набирать текст за пределами заданной таким образом длины строки будет приводить к автоматическому переводу курсора на следующую строку.

ПР З — запись текстового файла на дискетку ДВК-2М. После получения этой команды редактор задает вопрос: «Запись?» Если ответить Д, то редактор задает вопрос: «Имя файла:?» В ответ на это следует ввести имя, состоящее из не более чем шести заглавных латинских букв, и нажать клавишу ВВОД. После этого редактор запишет на диск текстовый файл с указанным именем, содержащий текст, который находился до этого в памяти редактора.

ПР Ч — чтение файла с дискетки. После получения этой команды редактор задает вопросы (1) и (2). Отвечать на них следует аналогично тому, как это делается при начале работы с редактором. Если до того, как была дана эта команда, в памяти редактора уже находился какой-то текст, то считанный по ней с дискетки текст будет присоединен к концу текста, который ранее находился в памяти.

С помощью этой команды можно компоновать текстовые файлы из имеющихся на диске файлов-фрагментов.

Если случится так, что для размещения полученного в результате выполнения этой команды текста не хватает места в памяти, то файл с диска будет читаться лишь до тех пор, пока памяти хватает. После этого чтение будет прекращено, на экране появится сообщение «Не хватает памяти», и при дальнейшей работе редактор будет игнорировать любые команды, связанные с добавлением новых строк в имеющийся в памяти текст.

ПР П — повторение последующей команды. Получив эту команду, редактор выдает запрос «Повтор:». В ответ на это следует ввести желаемое число повторений после-

* В более поздних версиях команда «Префикс» задается клавишей КТ.

дующей команды, затем набрать команду, которую следует повторять. После этого редактор выполнит избранную команду указанное число раз.

PR СУ/Ъ (на пленочной клавиатуре также PR ↖) — переход к метке. Установка курсора в строку, в которой в последний раз ставилась метка.

PR СУ/Ј (на пленочной клавиатуре также PR ↗) — запись текста в буфер (т. е. промежуточную память редактора). Получив эту команду, редактор удалит текст от строки, где последний раз ставилась метка, до строки, где стоит курсор, скопировав этот текст в буфер. Емкость буфера — приблизительно 20 строк. Эта команда вместе с последующей используется для перемещения части текста с одного места на другое.

PR СУ/Ч (на пленочной клавиатуре также PR ↘) — копирование текста из буфера. Получив эту команду, редактор вставляет в текст, начиная со строки, где стоит курсор, текст, который был ранее записан в буфер. Эта команда удобна, если требуется вставить один и тот же фрагмент в несколько мест текста.

PR СУ / \ (на пленочной клавиатуре также PR ↙) — установка модели контекстного поиска. Эта команда используется для автоматического нахождения в тексте нужного слова или иной последовательности символов, что избавляет от утомительной необходимости просматривать весь текст в поисках нужной строки. Получив эту команду, редактор выдает запрос: «Модель». В ответ надо ввести комбинацию символов (например, слово или его часть), которую в дальнейшем придется искать в тексте, и нажать клавишу ВВОД. Теперь модель контекстного поиска задана. Если в ходе дальнейшей работы с редактором дать команду СУ/Э, то редактор автоматически произведет поиск заданной последовательности символов в тексте, лежащем ниже курсора. Если эта последовательность будет найдена, то курсор будет установлен на ее начало, в противном случае будет выдано сообщение «Контекст не найден».

PR М — задание макрокоманды. Макрокоманда — это произвольная последовательность команд экранного редактора, объединенных в одну большую команду. Получив эту команду, редактор задает вопрос: «Макро?» Если ответить Д, то все последующие команды, которые будут принимать редактор, будут запоминаться как команды, составляющие макрокоманду. EDASP может хранить в памяти только одну макрокоманду, задание новой макрокоманды отменяет предыдущую.

PR О — окончание макрокоманды. Полу-

чив эту команду, редактор завершает формирование макрокоманды и переходит к обычному выполнению команд.

PR И — исполнение макрокоманды. Получив эту команду, редактор автоматически выполняет заданную ранее макрокоманду.

PR W — выход из редактора без сохранения текста. Получив эту команду, редактор задает вопрос: «Выход, Вы уверены?» Если ответить Н, то выхода не произойдет. При любом другом ответе произойдет выход из редактора в стартовый монитор БК-0010. Дав ему команду F, можно перейти в FOCAL или BASIC, в зависимости от того, что записано в ПЗУ БК-0010.

Эта команда используется для завершения работы с экраным редактором EDASP.

Дополнительные сведения

Оперативная память EDASP — приблизительно 10000 символов, что соответствует примерно 5 страницам машинописного текста. В верхней (контрольной) строке экрана дисплея постоянно высвечивается объем оставшейся свободной памяти редактора (в знаках). Если этот объем будет близок к исчерпанию, на экране появится сообщение: «Буфер полон. Запись?» Если ответить Д, то произойдет запись текста на дисковод аналогично команде «Запись». Если ответить Н, то можно будет продолжать редактирование текста, но добавить к нему разрешается лишь не более 10 строк. После этого рекомендуется записать текст на дисковод с помощью команды «Запись».

Для облегчения работы с командами «Префикс», «Макро», «Повтор» при их выполнении появляются соответствующие индикаторы в верхней (контрольной) строке.

После того как с помощью экранного редактора текстовый файл создан и записан на диск, его можно там хранить практически неограниченное время. В случае необходимости этот файл можно распечатать на печатающем устройстве КУВТа. Для этого следует использовать программу CONLST.SAV, которую рекомендуется иметь на системном диске ДВК-2.

Запускается она (при загруженной операционной системе) с пульта ДВК-2 командой R CONLST

Начав работу, она выдает на экран ДВК-2 список режимов работы. Следует выбрать режим Edasp→LST. После этого программа спросит имя входного файла. В ответ следует ввести имя текстового файла, который нужно распечатать. После этого программа спросит имя выходного файла. Если в ответ на это ввести имя, состоящее не

более чем из шести заглавных латинских букв, то программа создаст на диске текстовый файл в формате ДВК-2 с этим именем, содержащий текст, аналогичный входному файлу. Этот файл можно распечатать с помощью команды PRINT или же известной программы постраничной печати PAGER.

Если перед именем выходного файла будут стоять символы TT:, то программа не будет создавать текстовый файл на диске, а сразу напечатает выходной файл на печатающем устройстве. После этого программа снова запросит имя входного файла. В ответ на это следует либо дать ей на вход имя другого текстового файла, либо прекратить работу, дав команду СУ/Ц.

Рекомендации по методике

На изучение темы «Система подготовки текстов» рекомендуется отвести не менее 5 ч.

Урок 1. Лекционное занятие в обычном классе. Понятие о системе подготовки текстов. Понятие об экранном редакторе. Сведения об экранном редакторе EDASP. Порядок вызова, чтения файла, редактирования текста, основные команды.

Урок 2. Практическое занятие. Редактирование текста. Текстовый файл рекомендуется подготовить заранее. Это может быть известный стихотворный текст, содержащий большое число грамматических ошибок и

смысловых искажений. Учащимся ставится задача устранить все ошибки в тексте и записать его на диск.

Урок 3. Практическое занятие. Набор текста с помощью редактора EDASP (тексты заранее печатаются на бумаге и раздаются учащимся). Порядок распечатки текстовых файлов на печатающем устройстве. Распечатка созданных учащимися текстовых файлов.

Урок 4. Практическое занятие. Команды редактора, набираемые с помощью команды «Префикс». Выполнение заданий с применением этих команд на заранее подготовленном учебном текстовом файле.

Урок 5. Контрольная работа. Создание текста путем его переделки из существующего текстового файла, содержащего текст, близкий по содержанию к тому, который требуется (например, автобиография, заявление о приеме на работу и т. д.). Распечатка данного текста.

Возможно также применение описанной выше системы для подготовки карточек-заданий по другим предметам. Это можно поручать учащимся в порядке общественного полезного труда.

Данную систему можно скопировать (на диски заказчика) по адресу: 656099, Алтайский край, г. Барнаул, Социалистический проспект, 60, каб. № 15.

Автор желает успеха всем настоящим и будущим пользователям системы.

В. ДЕМЕНТЕЕВ,
г. Биробиджан

«Агат» рисует

«Лучше один раз увидеть, чем сто раз услышать». В справедливости этой поговорки легко убедиться на уроках информатики при изучении тем, связанных с получением графических изображений. На экране дисплея ПЭВМ «Агат» можно получить любые зрительные эффекты: объекты могут менять цвета, форму, возникать и исчезать, двигаться и совершать другие завлекательные действия.

Программирование изображений — занятие не простое, но очень увлекательное. Наглядность результата, возможность влиять на ситуацию стимулируют у учащихся желание составлять «графические программы». Как правило, к этому стремятся даже слабые ученики.

Создание статических изображений достаточно хорошо описано в руководстве по программированию на «Агате», поэтому остановимся на программировании динамических изображений. Описываемые упражнения выполнялись учениками IX—X классов средних школ г. Биробиджана.

Программировать динамические объекты, пользуясь оператором PLOT, очень сложно и возможно лишь в случае простых фигур (блин, тарелка, книга, змея, треугольник...). Рисунок может быть цветным, но скорость движения объектов мала, программы громоздки, неудобны для чтения.

Для управления изображениями объектов со сложной конфигурацией и большой динамичностью используются операторы

DRAW, XDRAW, ROT, SCALE, работающие в режиме ГВР (графика высокого разрешения). Объем ОЗУ «Агата» не обеспечивает цветной графики с высокой разрешающей способностью, поэтому предложенные ниже примеры одинаково выглядят на цветном и черно-белом дисплеях.

Изображение строим по принципу «черепашки», которая перемещается по экрану и, когда необходимо что-либо начертить, опускает хвост; при движении без черчения хвост поднят. Для управления действиями черепашки в пределах одного шага используются три бита, которые обозначим РДД.

Бит Р управляет хвостом:

0 — хвост поднят (движение без черчения);

1 — хвост опущен (движение с черчением).

Биты ДД указывают направление движения.

00 — вверх по экрану;

01 — вправо;

10 — вниз;

11 — влево.

Последовательность шагов записывается в последовательность ячеек оперативной памяти, называемую описателем. Один или несколько таких описателей вместе со служебной информацией составляют таблицу образов.

Каждый байт описателя разделен на три выполняемые последовательно секции А, В, С, каждая из которых описывает один шаг. Структура байта описателя приведена в табл. 1.

Таблица 1

	Секция		
	С	В	А
Номер бита	7 6	5 4 3	2 1 0
Содержание битов	Д Д	Р Д Д	Р Д Д

Секция С не содержит бита Р, в нее записывается только движение вниз, вправо и влево без черчения.

Для получения изображения на экране нужный описатель считывается из памяти ПЭВМ операторами XDRAW I AT X, Y

или
DRAW I AT X, Y

где XDRAW — воспроизведение объекта белым цветом, DRAW — черным; I — номер описателя; X и Y — координаты начала движения. Команды XDRAW и DRAW просматривают описатель секцию за секцией. Если секции, оставшиеся до конца байта, содержат одни нули, они игнорируются.

Например, байт описателя не может заканчиваться секцией С, равной 00 (движение вверх без черчения), так как она будет проигнорирована. Просмотр описателя заканчивается, когда встречается байт, все биты которого равны нулю.

Для примера изобразим на экране прямоугольник с отсеченными углами (рис. 1), который увеличивается в размерах и вращается.

Выбираем начальную точку 0 и строим путь движения «черепашки» (рис. 2).

Учитываем, что повороты возможны только на 90 градусов.



I
→ шаг без черчения;
→ шаг с черчением.

Последовательность шагов рис. 2, начиная с точки 0, такова:

↓ ↓ ↓ ← ← ↑ ↑ ↑ → → → ↓ ↓ ↓ ← ←

Закодируем ее и запишем в табл. 2.

Если код шага невозможно поместить в данную секцию, то она заполняется нулями, а код переносится в следующий байт.

Переводим значения байтов из двоичной системы счисления в шестнадцатиричную, используя табл. 3.

Записываем описатель в шестнадцатиричной системе счисления:

12 1F 20 44 6D 32 D6 07 00

Таблица 2

Байты	Условное содержимое секций			Реальное содержимое секций		
	С	В	А	С	В	А
0		↓	↓	00	010	010
1		←	←	00	011	111
2		↑	↑	00	100	000
3	→	↑	↑	01	000	100
4	→	↓	↓	01	101	101
5		→	→	00	110	010
6	←	↓	↓	11	010	110
7		←	←	00	000	111
8	Конец описателя			00	000	000

Тетрады	Шестнадцатиричные цифры
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Эту последовательность байтов необходимо записать в память «Агата».

Для завершения создания таблицы образов необходимо закодировать еще некоторую служебную информацию. Она предшествует описателю и показана в табл. 4.

К — адрес, с которого таблица образов записывается в оперативную память. Все адреса хранятся в порядке «младший байт, старший байт».

Таблица 4

Адрес ячейки	Содержание ячейки	Формат
К	Количество образцов в таблице	Байт
К+1	Не используется	Байт
К+2	Относительный адрес начала первого описателя в таблице образцов	Адрес
К+2*I	Относительный адрес начала I-го описателя в таблице образцов	Адрес

В нашем примере таблица образов содержит один описатель, следовательно, в шестнадцатиричном формате значения байтов будут следующие: [K]=01; [K+1]=00; [K+2]=04; [K+3]=00. Абсолютный адрес первого байта описателя равен K+4, а относительный равен 4. Описатель закончится байтом 00.

Запись таблицы образов производим в ОЗУ пользователя начиная с адреса K=1DFC. Выбирая адрес, нужно быть очень внимательным, чтобы при работе программы ваши исходные данные не были испорчены. Если же такая неприятность произойдет — измените начальный адрес размещения таблицы образов.

Командой CALL — 151 входим в системный монитор. Набираем на клавиатуре 1DFC:01 00 04 00 12 1F 20 44 6D 32 D6 07 00 (ввод). По этой команде таблица образов будет помещена в ОЗУ начиная с адреса 1DFC. Далее необходимо сообщить интерпретатору адрес начала таблицы. Для этого служат ячейки E8 и E9, причем в E8 записываются младшие разряды адреса, а в E9 — старшие. В системный монитор адрес помещаем следующим образом:

E8:FC 1D (ввод)

Затем необходимо вернуться в интерпретатор, либо набрав E003G (ввод), либо одновременно нажав клавиши СБР и УПР.

При наличии пригласительного знака набираем следующую программу.

```

10 HGR=2: X1=0
20 FOR X=0 TO 15
30 SCALE=X: ROT=2*X
40 XDRAW 1 AT 120,120
50 SCALE=X1: ROT=2*X1
60 DRAW 1 AT 120,120
70 X1=X
80 NEXT X
90 GOTO 10
    
```

Здесь строка 10 задает режим ГВР, начальное значение X1;

20 — заголовок цикла;

30 — масштаб и угол поворота;

40 — воспроизведение объекта белым цветом;

50 — изменение масштаба и угла поворота;

60 — воспроизведение объекта черным цветом;

70 — запоминание параметра;

80 — конец цикла;

90 — безусловный переход.

Эта программа обеспечивает отсутствие мигания при вычерчивании объектов. При исполнении программы таблица образов и ее адрес должны находиться в памяти машины. Чтобы при повторном запуске программы каждый раз не вводить с клавиатуры таблицу образов, ее необходимо либо сохранить на магнитном диске, либо ввести в текст программы. На ГМД таблице записывается командой

BSAVE (имя), A \times 1DFC,L \times D

где имя — идентификатор, под которым таблица будет записана в библиотеку B;

A \times 1DFC — начальный адрес таблицы;

L \times D — количество байтов в таблице.

По команде BLOAD (имя) таблица образов вызывается с ГМД и загружается в память по адресу, с которого она была прочитана (1DFC).

Не вызывая системный монитор, непосред-

ственно в тексте программы командами POKE\$E8, \$FC:POKE\$E9, \$1D можно поместить адрес таблицы образов в ячейки E8 и E9 (\$ — признак шестнадцатичного формата записи чисел и адресов). Кроме того, таблицу образов можно задать в тексте программы с помощью последовательности команд POKE.

Рассмотрим фрагмент программы «Гонки на парусных лодках», созданной членами кружка информатики. В программе используются генераторы случайных чисел для изменения скорости движения лодок, задается длина дистанции (сколько раз лодки должны «проплыть» по экрану), выдается сообщение о том, чья лодка пришла к финишу первой, второй, третьей и за какое время.

Таблица образов для рисования парусной лодки и стирания ее следа:

```
02 00 06 00 12 00 2D D5 3F 3F 03 28
2D 24 24 37 05 00 24 04 18 FE 92
35 35 3F 24 00
```

Во фрагменте приведенной ниже программы таблица с ГМД вводится в ОЗУ, и на экране

появляется изображение трех движущихся лодок.

```
1 PRINT CNR$(4) «BLOADLODKA»
5 POKE$E8; $FC:POKE $E9, $1D
10 HGR=2
20 SCALE=3
30 ROT=0
40 FOR X=0 TO 255
50 XDRAW 1 AT X, 128
53 XDRAW 1 AT X+3,108
55 XDRAW 1 AT X+7,148
60 DRAW 2 AT X1,128
65 DRAW 2 AT X1+3,108
67 DRAW 2 AT X1+7,148
70 X1=X
80 NEXT X
90 GOTO 40
```

Для учеников графика является одной из любимых областей применения компьютера, а для учителя должна играть не последнюю роль при создании обучающих программ по алгебре, геометрии, физике, черчению и другим предметам. Создавать такие программы нужно как можно активнее.

А. КАРАВАЕВ

Электронный коммутатор видеосигнала для дисплейного класса

Одной из основных проблем преподавателя информатики является оперативный контроль за работой учеников. Сегодняшняя реализация КУВТ-86, а также КУВТ на базе УКНЦ не позволяет этого сделать. Передача информации по сети УКНЦ происходит очень медленно, при этом, как правило, занимается рабочее место преподавателя. А. Караваевым предложено простое и технически грамотное решение этой проблемы.

А. АРХАНГЕЛЬСКИЙ

Первый опыт работы по проведению практических занятий с учащимися в дисплейном классе показал, что необходимо облегчить труд учителя информатики. В процессе общения с ЭВМ у учащихся возникает масса вопросов. Прежде чем ответить, учитель должен проанализировать сообщение на экране монитора учащегося. Бесконечная ходьба по классу в течение нескольких часов утомительна. Как от нее избавиться?

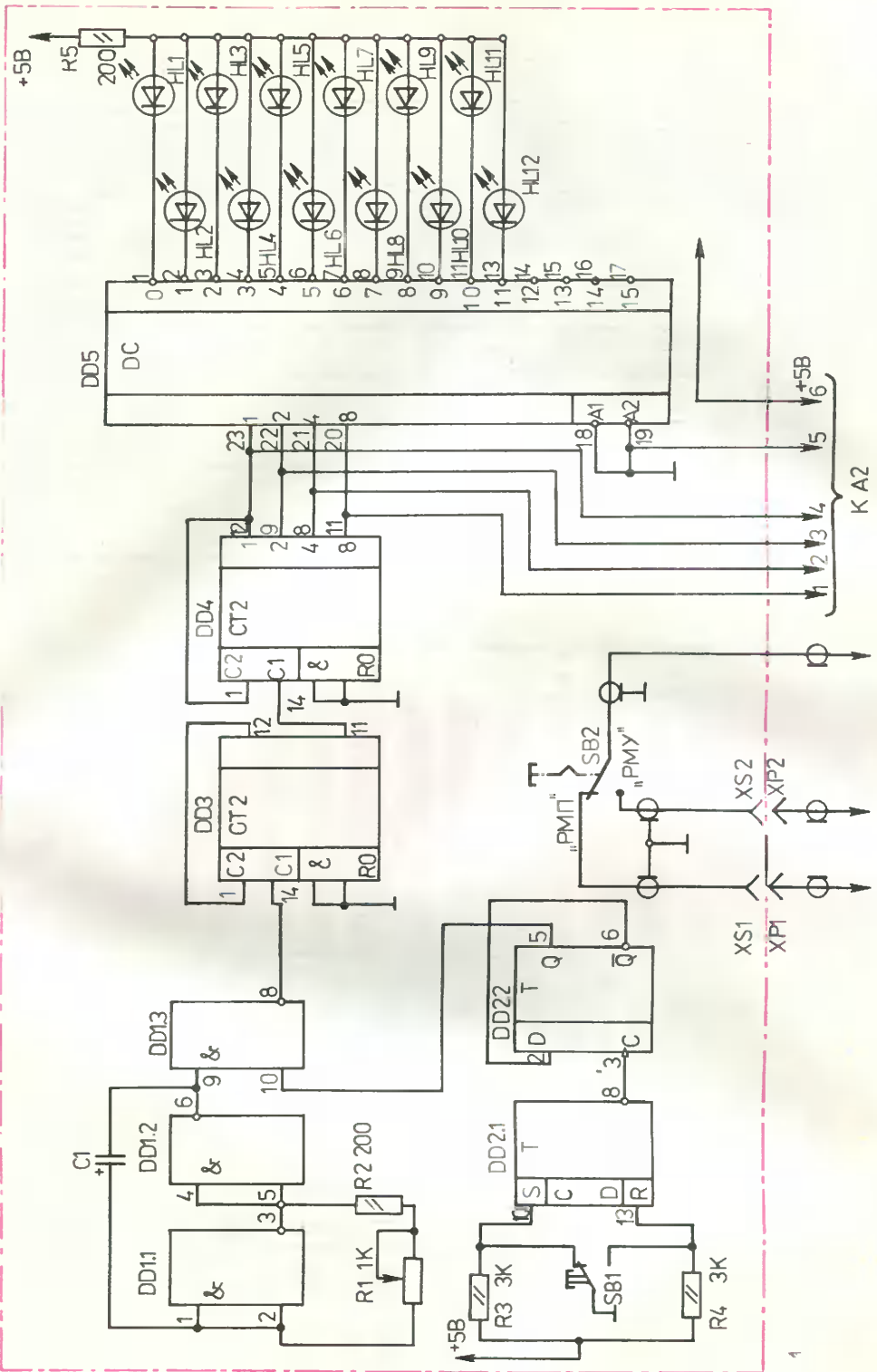
А что если привлечь программно-аппаратные средства? Для нового комплекса УКНЦ, который мы начали эксплуатировать, таких сервисных программ пока нет. Но даже при их наличии возникнет ряд неудобств, так как машины преподавателя и учащихся объединены в замкнутую локальную сеть. При передаче информации по витой паре про-

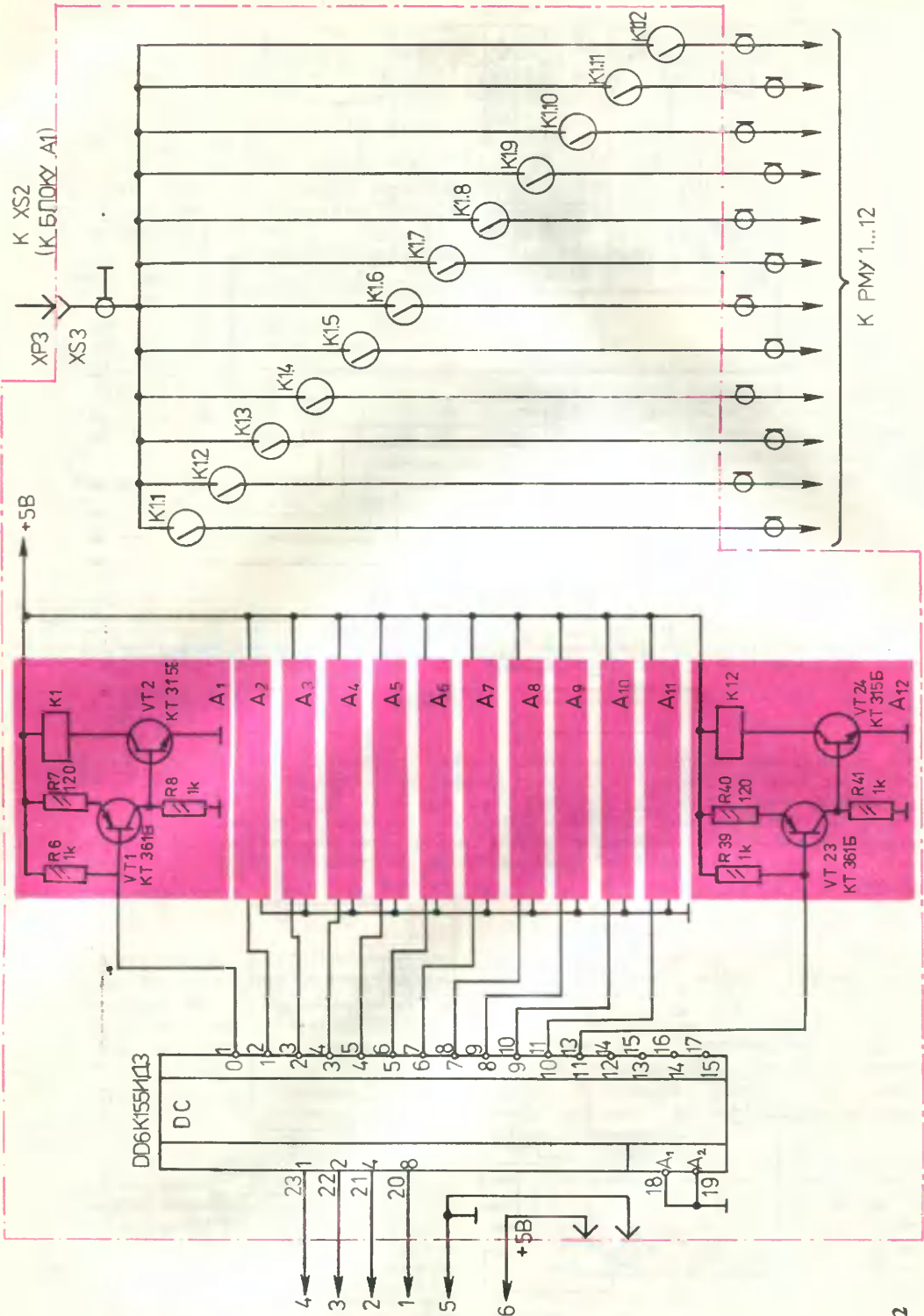
дов оперативность комплекса в процессе исполнения команд по запросам недостаточна.

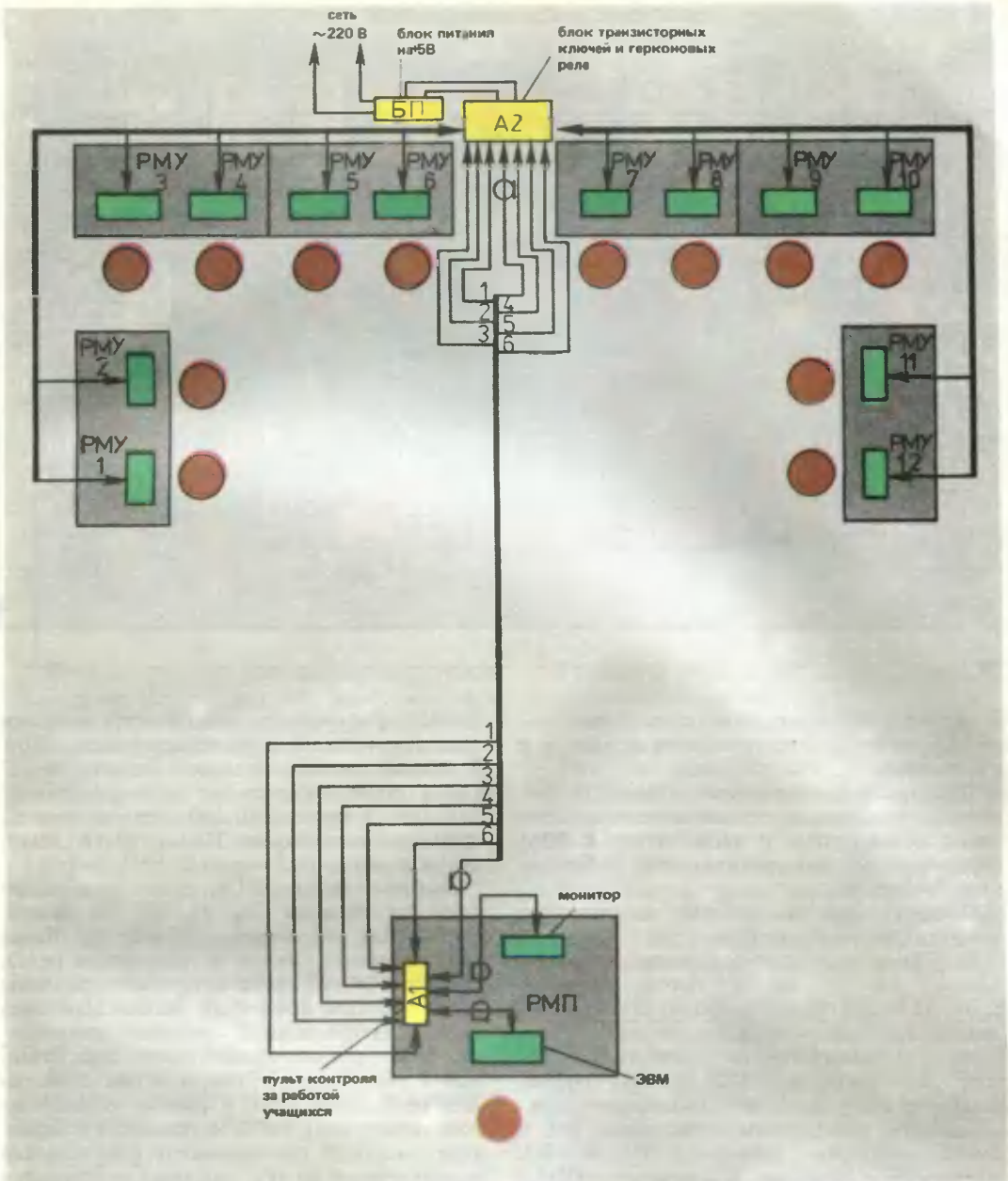
Мы с членами школьного кружка электронного конструирования решили проблему оперативного «копирования» изображения с мониторов учащихся на монитор преподавателя с помощью внешних электронных устройств.

В чем достоинства предлагаемого варианта? Во-первых, отпадает необходимость вмешиваться в работу локальной сети. Во-вторых, привлекает простота и удобство управления устройством (всего две кнопки на пульте преподавателя). Кроме того, не нужны дефицитные детали или предусмотрена их замена.

Устройство позволяет вести работу в двух







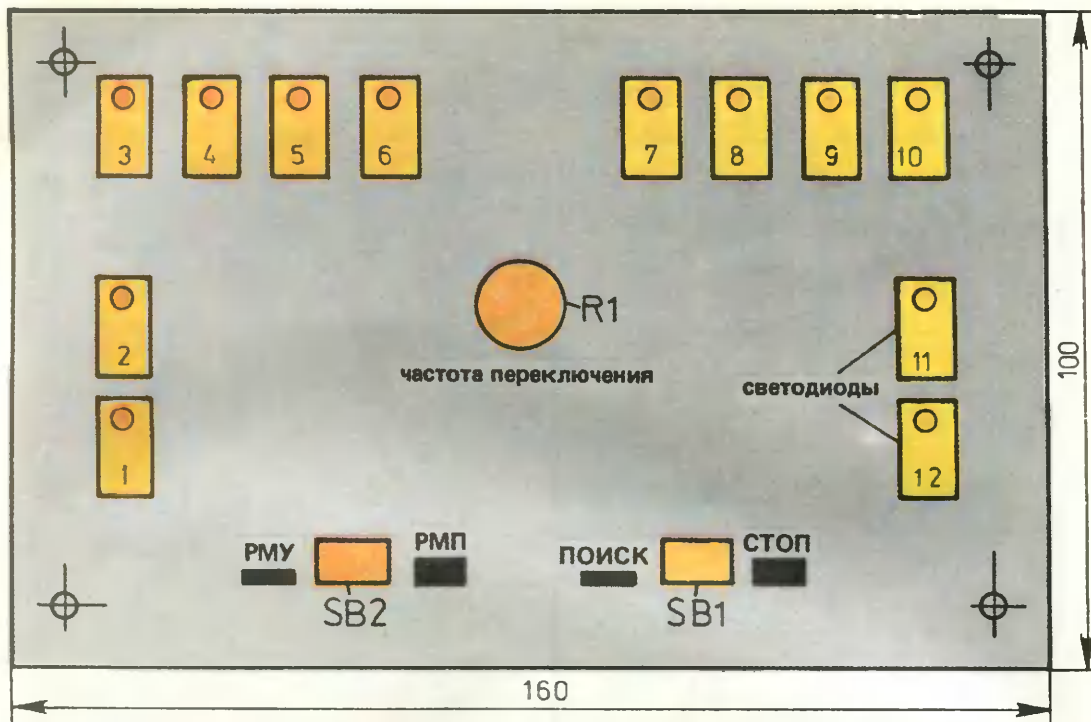
режимах — автоматического или индивидуального контроля.

В режиме автоматического контроля на монитор преподавателя последовательно выводится изображение с мониторов учащихся. Необходимая частота коммутации видеосигнала устанавливается ручкой переменного резистора.

В режим индивидуального контроля можно перейти нажатием кнопки на пульте. При

этом «зафиксируется» изображение от той ЭВМ учащегося, которая в данный момент была подключена к видеовходу монитора преподавателя. Переход в автоматический режим осуществляется вторичным нажатием на ту же кнопку.

Как преподаватель узнает, с какого монитора в данный момент «копируется» изображение? Для этого служат индикаторы на светодиодах. Расположение светодиодов и их



нумерация на пульте аналогичны расположению рабочих мест учащихся в классе и их нумерации.

Преподаватель при желании может выйти из режима контроля, отключив источник питания коммутатора. В этом случае к ЭВМ преподавателя подключится его собственный монитор.

Опишем принцип работы электронного коммутатора видеосигнала.

Устройство состоит из двух блоков — А1 и А2, а также блока питания. Блок А1 — это блок контроля, его принципиальная схема изображена на рис. 1. Этот блок устанавливается на столе преподавателя. На элементах DD1.1, DD1.2 собран генератор импульсов, частоту которого можно менять переменным резистором R1, а также подбором элементов C1 и R2. Импульсы с выхода 8 элемента DD1.3 поступают на вход 14 микросхемы DD3 только в том случае, если на входе 10 элемента DD1.3 присутствует уровень 1. Если же на этом входе будет уровень 0, то прохождение импульсов блокируется. Эти уровни будут сменять друг друга при периодическом нажатии кнопки SB1 (эта кнопка — без фиксации). Триггер DD2.1 устраняет дребезг контактов, а триггер DD2.2 работает в счетном режиме. Микросхемы DD3 и DD4 делят частоту импульсов генератора. На выходах 12, 9, 8, 11 счетчи-

ка DD4 формируется код. К этим выходам подключен дешифратор на микросхеме DD5. В режиме автоматического контроля на выходах этой микросхемы последовательно появляется уровень 0, при этом загорается один из светодиодов HL1... HL12, индицируя номер подключенной ЭВМ.

Выходы счетчика DD4, кроме того, соединены со входами 20, 21, 22, 23 микросхемы DD6, находящейся в блоке А2 (блоке транзисторных ключей и герконовых реле), принципиальная схема которого изображена на рис. 2. При появлении логического нуля на соответствующем выходе микросхемы DD6 транзисторный ключ, подключенный к этому выходу, открывается, срабатывает герконовое реле¹ и своими контактами подключает одну из ЭВМ учащихся к входу монитора преподавателя. Она остается подключенной до тех пор, пока сохраняется неизменным код на входах микросхем DD5, DD6.

Коротко об особенностях конструкции устройства. Видеосигнал от ЭВМ учащихся к монитору преподавателя передается по обычному телевизионному коаксиальному кабелю с волновым сопротивлением 75 Ом. Эти кабе-

¹ Герконовые реле можно заменить на малогабаритные электромагнитные (например, РЭС-10), срабатывающие от источника напряжения 5В.

ли проложены по полу вдоль плинтуса от каждой машины к блоку А2, установленному в одном из учебных столов. Место установки блока выбирается с учетом наименьшего расхода кабеля и расположения сетевой розетки для подключения блока питания. Схема блока питания со стабилизацией на 5 В не приводится. Подобные схемы часто печатаются в журнале «Радио». Можно также блок питания заменить двумя параллельно соединенными батарейками типа «Планета» (4, 5В). Конечно, для удобства монтажа желательно использовать тонкий коаксиальный кабель. Соединение блоков между собой показано на рис. 3. В качестве линии из проводов 1, 2, 3, 4, 5, 6 использован 5-ти парный телефонный ка-

бель. Неиспользованные жилы его остаются в качестве резерва.

Вариант оформления панели пульта контроля — на рис. 4.

Если дисплейный класс расположен в помещении с небольшой площадью, то можно исключить микросхему DD6, а все устройство собрать в одном корпусе. При этом входы транзисторных ключей подсоединяются к выходам микросхемы DD5. При отсутствии светодиодов можно вместо них включить через транзисторные ключи лампочки для карманного фонарика.

Элементы схемы монтируются на печатных платах из одностороннего фольгированного стеклотекстолита.

В. ЗИНЮК, Э. БАРАНАУСКАС
г. Мурманск

67

ПМК на уроке труда

Последние два года в слесарной мастерской школы № 42 г. Мурманска с успехом используются микрокалькуляторы при выполнении различных практических работ. Первоначально использовались простейшие МК для вычислений при работе на токарных, фрезерных станках, разметке.

Для настройки токарных станков ТВ-4, ТВ-6 на точение заданного размера по диаметру используется формула

$$n = \frac{D-d}{2 \cdot 0,025} = 20(D-d),$$

где n — число делений лимба поперечной подачи, D — первоначальный диаметр заготовки, d — заданный диаметр детали, 0,025 — цена деления лимба.

При точении конических поверхностей тангенс угла поворота верхней части суппорта вычисляется по формуле

$$\operatorname{tg} \alpha = \frac{D-d}{2l},$$

где D — большой диаметр конуса, d — малый диаметр конуса, l — длина конуса. Затем по таблице тангенсов определяется величина угла.

На первом этапе использования МК учащиеся производят соответствующие вычисления, приобретают при этом навыки использования микропроцессорной техники для выполнения практических работ. Это сокращает время вычислений, повышает точность, качество выполняемых работ.

Расчеты упрощаются, если использовать программируемые МК.

Программа вычислений числа делений лимба для ПМК «Электроника-54» и «Электроника-56» такова: 00.— 01.2 02.0 03*04.С/П 05.БП 06.00. Конкретные данные для вычислений вводятся в следующем порядке: D ↑ V d V/O C/П

При незначительном количестве ПМК в мастерской и разнообразии выполняемых учащимися работ целесообразно составлять сводные программы по наиболее часто используемым формулам. Вот одна из них.

	00	01	02	03	04	05	06	07	08	09
00	↑	1	—	X=0	12	С/П	—	2	0	*
10	БП	05	1	—	X=0	25	С/П	—	←	÷
20	2	÷	tg-1	БП	16		—	X=0	36	С/П
30	—	4	0	*	БП	29	1	—	X=0	51
40	С/П	*	π	*	1	0	0	0	÷	БП
50	40	С/П	↑	1	8	0	←	÷	sin	*
60	БП	51								

Программа вводится в ПМК перед занятием и позволяет производить расчеты по одной из пяти формул.

- $n = 20(D-d)$.
- $\alpha = \arctg((D-d)/2l)$.
- $n = 40(L-l)$: число делений лимба продольной подачи суппорта при подрезании детали до заданной длины, L — первоначальная длина заготовки, l — длина детали по чертежу.
- $V = \frac{\pi D n}{1000}$ м/мин: скорость резания при

цилиндрическом точении, D — диаметр заготовки, m , n — число оборотов шпинделя, об/мин.

5. $H = D \sin \frac{\pi}{n}$: длина хорды n -угольника, вписанного в окружность диаметра D .

Для выбора любой из приведенных формул необходимо набрать ее номер:
№ формулы В/О С/П

После этого можно вводить данные в соответствии с образцами:

1. $D \uparrow C / \text{П}$
2. $l \uparrow D \uparrow d \text{ С/П}$
3. $L \uparrow l \text{ С/П}$
4. $D \uparrow n \text{ С/П}$
5. $D \uparrow n \text{ С/П}$

После получения ответа можно снова ввести данные, и калькулятор выполнит вычисления по выбранной вами формуле.

Картинка — на принтер

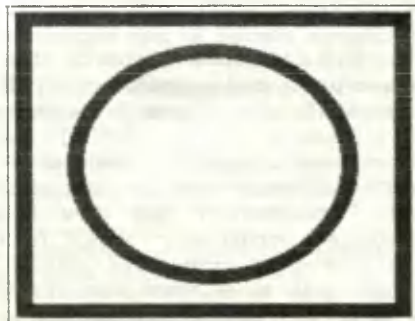
В секции прикладной математики и вычислительной техники электростальского филиала Московского института стали и сплавов найдена возможность вывода на принтер преподавательской ЭВМ машинной графики компьютеров «Электроника БК-0010Ш», работающих на Бейсике.

В журнале «Информатика и образование» (1988, № 5, с. 58) уже сообщалось о возможности печати листингов программ и алфавитно-цифровых результатов счета ЭВМ БК-0010Ш без обращения к накопителю на гибких магнитных дисках ЭВМ ДВК-2М*.

Идея печати машинной графики БК-0010Ш заключается в копировании на принтер картинки с экрана дисплея. Принтер подачей управляющих кодов может быть переведен в растровый (графический) режим работы. Определенное число поступающих затем кодов обрабатывается им следующим образом: срабатывают иглы печатающей головки принтера, соответствующие «установленным» битам в двоичном представлении анализируемого кода. Машинной графике Бейсик-системы доступна область экранного ОЗУ с адреса 17408 по 32767 размером 240 на 512 точек. Следовательно, при копировании построчно картинки с экрана дисплея на принтер надо обработать в растровом режиме 512 кодов, составленных так, чтобы иглы принтера срабатывали в необходимых местах. Подача с БК-0010Ш на принтер нулевого кода затруднительна, но его можно заменить кодом 128, настроив принтер на обработку только семибитной информации. Количество строк при построчном копировании картинки может быть равно $240:6=40$, но тогда картин-

ка получится мелкой. Крупная картинка получается при числе строк, равном 80, где каждой копируемой точке картинки соответствуют два соседних бита в двоичном представлении обрабатываемого принтером кода.

Приведем Бейсик-программу (авторы — И. А. Буров и А. С. Разумов), позволяющую получить на принтере ROBOTRON CM 6329.02-М крупную копию картинки с экрана дисплея. В ней строки 110—190 — построение картинки (в данном случае — окружности), строки 200—400 — вывод картинки из экранного ОЗУ на принтер. На принтер с БК-0010Ш необходимо передать $519 \times 80 + 5 = 41\,525$ кодов, включая управляющие коды установки интервала подачи бумаги, перевода принтера в растровый режим, а также его инициализации. Для промежуточного хранения такого объема информации не хватает буфера ДВК-2М. Поэтому вся информация делится пополам и передается частями. Поскольку программа просматривает адресное пространство экранного ОЗУ последовательно, то его первый бит должен соответствовать точке экрана в левом верхнем углу. Режим экранного «ролика» нарушает это соответствие, но его можно восстановить «смаргиванием» экрана, дважды подав управляющий код 140. Эта процедура обязательна перед построением копируемой на принтер картинки. Общее время копирова-

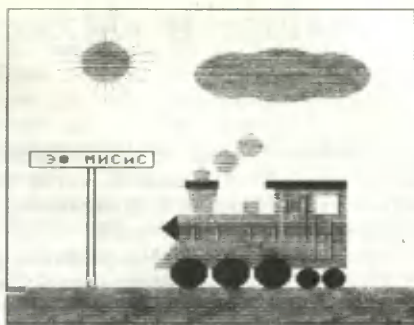


* К сожалению, в эту заметку вкрались опечатки: в приведенной программе аргумент оператора OPEN — не TT:RL, а TT:LP; вместо «занятые и встроенные функции» следует читать «занятые и встроенные функции». — Примеч. ред.

ния составляет около 390 с, из них только 56 с тратится на формирование требуемых кодов, остальное время уходит на передачу их принтеру.

Кратко опишем назначение используемых в программе переменных: OO % — код, передаваемый на принтер для срабатывания в необходимых местах иглол печатающей головки; OA % — адрес ОЗУ, на единицу меньший начального адреса очередной копируемой строки картинке; OB % — адрес машинного слова экранного ОЗУ, содержимое которого определяет старшие биты передаваемого кода; OC %, OD %, OE % — содержимое машинных слов экранного ОЗУ соответственно с адресами OB %, OB % + 64, OB % + 128; OG % — номер анализируемого бита машинного слова; OF % — маска, у которой «установлен» единственный бит с номером OG %. Алгоритм копирования картинке несложен и, надеемся, будет понятен из листинга программы.

Драйвер LP.SYS должен быть настроен на прием графической информации командой SET LP:CTRL,WIDTH=525 монитора RT11SJ.SYS, а затем загружен командой LOAD LP в ОЗУ ДВК-2М. В нашем случае использовался драйвер LP.SYS версии от 29.10.84. Драйвер другой версии может потребовать изменения команды настройки его на прием графической информации. Программа NET.SAV связи ДВК — БК должна



работать в режиме разрешения записи. Принтер необходимо настроить на обработку семибитной информации. Только тогда возможно получение на принтере копии картинке с экрана дисплея без обращения к накопителю на гибких магнитных дисках ЭВМ ДВК-2М.

Желающие получить консультацию могут обращаться по адресу: 144000, Московская обл., г. Электросталь, ул. Первомайская, 7, электростальский филиал Московского института стали и сплавов, секция прикладной математики и вычислительной техники; тел. 4-37-26, 4-40-42. Здесь можно также получить программы в машинных кодах для печати на принтере крупной и мелкой копий картинке с экрана дисплея БК-0010Ш. Время их работы вдвое меньше, чем программ на Бейсике.

69

```

100 * ПЕЧАТЬ МАШИНОЙ ГРАФИКИ КУВТ-86. ( ЭО МИСИС, 14.11.88 )
110 ? CHR$(140%);CHR$(140%) * ОБЯЗАТЕЛЬНО "СМОГНИТЕ" !!!
120 * ДЛЯ ПРИМЕРА РИСУЕМ КАРТИНКУ В РЕЖИМЕ 32 СИМВОЛА В СТРОКЕ:
130 LINE (0%,0%)-(255%,239%),1%,B
140 LINE (5%,5%)-(250%,234%),1%,B
150 LINE (15%,15%)-(240%,224%),1%,B
160 PAINT (6%,6%)
170 CIRCLE (127%,119%),90%,,,,1%
180 CIRCLE (127%,119%),80%,,,,1%
190 PAINT (127%,30%)
200 * КОПИРУЕМ КАРТИНКУ НА ПРИНТЕР ROBOTRON CM 6329.02-М:
210 OPEN "TT:LP:"
220 ? # CHR$(27%);CHR$(65%);CHR$(6%); * ИНТЕРВАЛ ПОДАЧИ 6/72"
230 FOR OAX=17407% TO 32575% STEP 192%
240 IF OAX=25087% THEN OPEN "TT:LP:"
250 ? # CHR$(27%);CHR$(42%);CHR$(5%);CHR$(1%);CHR$(2%); * РАСТРОВЫЙ РЕЖИМ
260 FOR OBX=OAX+1% TO OAX+63% STEP 2%
270 OCX=PEEK(OBX)
280 ODX=PEEK(OBX+64%)
290 OEX=PEEK(OBX+128%)
300 OFX=1%
310 FOR OBX=0% TO 15%
320 DOX=3%*ABS(SGN(OFX AND OEX))+12%*ABS(SGN(OFX AND ODX))+48%*ABS(SGN(OFX AND OCX))
330 ? # CHR$(DOX+128%);
340 IF OBX<14% THEN OFX=OFX+OFX ELSE OFX=-32768
350 NEXT OBX,OBX
360 ? # CHR$(128%)
370 IF OAX=24895% THEN CLOSE
380 NEXT OAX
390 ? # CHR$(27%);CHR$(64%) * ИНИЦИАЛИЗАЦИЯ ПРИНТЕРА
400 CLOSE

```

Подробнее о «Микроше»

Публикация сообщения об этом компьютере (ИНФО, 1988, № 5) вызвала поток писем. Редакция передала их В. В. Дворцовому, и он отвечает на вопросы читателей.

— Приобрести ПЭВМ «Микроша» можно в тех городах, где есть фирменные магазины «Радиотехника»; к их числу относятся Москва, Ленинград, Новгород, Минск, Казань, Баку, Ереван, Алма-Ата. Однако следует учесть, что компьютеров не хватает, поступают они в магазины нерегулярно и не обязательно удастся купить ПЭВМ с «первого захода».

В этих же магазинах бывает ПЭВМ «Кристы», отличающаяся от «Микроши» наличием «светового пера» и несколько другой конструкцией клавиатуры (с использованием герконов)*.

Добавочные ОЗУ и ПЗУ пока не продаются; сделать их самостоятельно тоже нелегко, так как соответствующие микросхемы дефицитны.

Вопрос о покупке «Микроши» по безналичному расчету решается непосредственно в магазине.

О совместимости с другими моделями ПЭВМ. Написанные на Бейсике программы для других компьютеров зачастую идут на «Микроше» без всяких переделок, если используется подходящий интерпретатор, а их создано уже немало. В других случаях необходимые переделки программ невелики. Вводят программы с клавиатуры.

Значительное число программ на Бейсике приведено в книге В. П. Дьяконова «Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ», изданной в 1987 г. тиражом 300 000 экз. Большинство программ из нее подходят (может быть, с минимальными переделками) для «Микроши». Книга содержит информацию об основных характеристиках и возможностях ПЭВМ, описание Бейсика (в частности, рассмотрены вопросы перевода программ с одной версии языка на другую), алгоритмы и программы элементарных вычислений, реализации основных численных методов, спектрального, статистического, корреляционного и регрессионного анализа, вычисления специальных

функций, прикладные программы технических и экономических расчетов.

Среди разработанных для «Микроши» интерпретаторов Бейсика лучшие — «Бейсик +», совместимый с Бейсиком MSX, и «Бейсик best», позволяющий использовать программы, написанные для IBM PC.

Конечно, программа для IBM PC не будет работать на «Микроше», если ей, например, требуется 500К байт памяти, но это уже проблема другого плана.

Наиболее существенное ограничение для переноса программ — отсутствие у «Микроши» полноценной графики; на нем реализована только псевдографика.

Непосредственно с магнитной ленты в «Микрошу» можно ввести программы компьютера РК-86, используя программу «Автосвод» в формате РК-86. Эта программа изменяет константы ввода и ведет подсчет контрольной суммы в формате РК-86. Если в программах РК-86 нет обращений к монитору, то они будут работать на «Микроше»; в противном случае адреса обращений к монитору в программах нужно заменить на «микрошинские»: F81B — на FE0A, F821 — на FE01 (фактически в программе эти адреса записаны иначе, и менять нужно запись 1BF8 на EAFE, 21F8 — на 01FE).

Нельзя использовать те программы РК-86, в которых есть обращения к подпрограммам монитора, отсутствующим у «Микроши».

Впрочем, особой необходимости в переделке программ РК-86 для «Микроши» нет, так как практически все программы, реализованные на РК-86, реализованы и на «Микроше».

Паспортные характеристики «Микроши». Система команд микропроцессора КР-580, количество команд — 78, тактовая частота системы — 1,8МГц, разрядность шины данных — 8 бит, разрядность шины адреса — 16 бит. Время выполнения операций типа регистр-регистр и арифметических операций — 3 мкс.

Объем ОЗУ — 32К байт, объем свободного дополнительного адресного пространства — 16К байт.

Имеются технические средства для связи по параллельному интерфейсу, последовательному интерфейсу (локальная сеть), по интерфейсу кассетного магнитофона.

Формат изображения при выводе на экран телевизионного приемника информации: алфавитно-цифровой — 64×25 знакомест; псевдографической — 128×50 точек.

* Перед самой сдачей статьи в печать выяснилось, что у «Кристы» есть также отличия и в мониторе: область видимой части экрана сдвинута в экранной памяти на 5 позиций влево относительно «микрошиной», так что некоторые программы «Микроши» на ней не идут.

Объем свободного дополнительного адресного пространства — 16К байт — можно использовать под добавочное ОЗУ или ПЗУ. Монитор «Микроши» размещен в ПЗУ объемом 2К байта.

Размещение памяти: ОЗУ 32К байт — с адреса 0000 по 7FFF; ПЗУ с монитором (2К байт) — с F800 по FFFF; ячейки ввода — вывода (8К байт) — с C000 по DFFF; свободное адресное пространство — с 8000 по F7FF.

Многие владельцы «Микроши» жалуются на треск во внутреннем динамике «Микроши» во время записи программы на ленту. Избавиться от него очень просто — достаточно занести в ячейку D803 код B0; из Бейсика, например, это делается так: POKE 10237, 176. При этом динамик ПЭВМ отключается, контролировать процесс записи можно через динамику магнитофона. Для включения динамика ПЭВМ достаточно нажать клавишу «сброс».

В следующих номерах журнала о про-

граммах для «Микроши» будет рассказано подробнее. Пока же упомянем о новой разработке — файловой системе FILMON, позволяющей при использовании в качестве устройства внешней памяти незначительно модифицированного магнитофона не только хранить на кассете МК-60 2М байта информации, но и осуществлять ее быстрый автоматический поиск (среднее время доступа — 1 мин).

Среди поступивших в редакцию откликов на заметку о «Микроше» было письмо Дениса Свинына. Он дает владельцам этой ПЭВМ несколько советов.

1. Чтобы обеззвучить клавиатуру, следует нажать клавишу УС, затем любую буквенную или цифровую клавишу (до тех пор, пока она не начнет отображаться на дисплее), а затем клавишу РУС — ЛАТ.

2. В «Микроше» есть два типа алфавита. Для перехода от одного к другому следует нажать F1. Программно то же самое делает команда POKE — 14335,255.

71

Новые возможности БК

Группа разработчиков программного обеспечения персонального компьютера «Электроника БК-0010», объединяемая кооперативом «Информ», в течение уже трех лет работает над системными программами, которые превращают этот бытовой компьютер в профессиональный. Каждый разработчик решал свою частную задачу, преследуя собственную цель. Эти разработки сформировались в некую систему, апробированную в многочисленных применениях. Конечно же, объективной целью была всеобщей, и очень скоро обнаружилось, что многие пользователи микрокомпьютеров БК-0010 ищут решения тех же проблем.

Кооператив «Информ» поставляет ряд программ, которые представляют собой набор «деталей», «узлов» и «инструментов». Этот набор дает возможность создавать прикладные системы — новые программные продукты — для различных областей применения (расчеты и математическое моделирование, автоматизация исследований и технологических процессов, образование и культурный досуг). Пользователи БК-0010 смогут эффективно заниматься этой работой, когда получат наши программы. С ними легко работать — все они снабжены исчерпывающими руководствами по использованию.

Ниже перечислены и кратко описаны предлагаемые в настоящий момент программные продукты.

Интерфейс Внешних Функций (ИВФ) интерпретатора Фокал-БК-0010 позволяет расширить язык Фокал новыми функциями (процедурами), приспособивая компьютер к какому-либо классу прикладных задач. Внешние функции — это программы в кодах. Они работают быстро и экономят память компьютера. За счет этого значительно повышается эффективность машины. Пользователь может сам создавать нужные ему системы внешних функций, имея только ИВФ, или наращивать готовые системы (системы-заготовки), добавляя необходимые процедуры.

Система Внешних Функций «МАССИВЫ 2» позволяет определять и использовать числовые массивы данных трех типов: байт, целое и действительное (1, 2 и 4 байта на одно число). Фокал (без системы «МАССИВЫ 2») разрешает только действительные числа (в формате с плавающей точкой), выделяя для каждого данного 8 байт памяти. Ко всему еще и время обращения к таким данным сильно зависит от их количества. Система «МАССИВЫ 2» снимает эти проблемы и в любом приложении оказывается эффективной.

Система Внешних Функций «Музыка-Цвет-Джойстик» (СВФМЦД) является набором процедур для организации программного интерфейса пользователя. Эта система обеспечивает прикладные программы средствами эффективного диалога с пользователем: окнами, музыкальными сообщениями, интерактивными операциями через указатель информации джойстик. СВФМЦД версии 01 позволяет создавать графические редакторы.

Система машинной графики PLOT (Фокал) реализует простой графический язык, создает удобства при программировании графического вывода. Она превосходно подходит к системе «Музыка-Цвет-Джойстик», при этом пользователь приобретает инструмент для полноценной оконной графики.

На основе двухлетнего опыта создания и эксплуатации разнообразных систем внешних функций разработчики пришли к новой версии ИВФ. Она получила название «Интерфейс Внешних Функций с режимом Расширения Памяти» (ИВФ-РП). ИВФ-РП поддерживает режим РП и является перемещаемым. Системы функций генерируются из библиотеки файлов на магнитной ленте с помощью диалоговой программы-компоновщика. Скомпонованные системы могут в свою очередь включаться в библиотеку файлов. Автоматическая генерация упрощает адаптацию машины к прикладным задачам пользователя.

На основе ИВФ-РП работает библиотечная Система Внешних Функций «МАРКА 2». Она позволяет создавать и использовать разнообразные шрифты и спрайты для выразительного оформления прикладных программ, обеспечивает мультипликацию, генерирует музыкальные сигналы. Для подготовки музыкального сопровождения к системе «МАРКА 2» придается музыкальный редактор «МЕЛОМАН-М».

Библиотечная Система Внешних Функций «ГРАФ 2» обеспечивает пользователя быстрой оконной графикой. Это удобный инструмент для представления результатов вычислений и обработки данных.

Система Внешних Функций «ПОИСК» служит для управления базами данных. «ПОИСК» — набор процедур-функций в кодах, обеспечивающих эффективное использование ОЗУ и высокую скорость работы. Прикладное содержание базы данных, ее параметры и весь диалог с оператором могут быть выполнены пользователем самостоятельно (на Фокале), исходя из конкретной задачи. «ПОИСК» позволяет перераспределять ОЗУ между Фокал-программой и данными системы, задавать длину информационной карты в базе данных, записывать и считывать информацию в выбранном формате,

проводить поиск информации по полной или сокращенной модели, записывать на магнитную ленту и считывать данные.

В библиотеку ИВФ-РП также включена система «МАССИВЫ 2», о которой говорилось выше.

Система Внешних Функций «FEDASP» позволяет преобразовывать тексты программ на Фокале в формат распространенного редактора текстов EDASP.

Имеется также библиотечная функция — драйвер принтера для распечатки программ на Фокале и данных из этих программ.

Разрабатывается функция создания на принтере «твердых» копий изображений с экрана дисплея.

ИВФ и ИВФ-РП являются мощными средствами расширения интерпретатора Фокал-БК-0010. Однако, если не используются готовые системы внешних функций (ИВФ) или библиотечные функции (ИВФ-РП), для подготовки новых функций нужны инструментальные программы разработки кодовых процедур-функций. Кооператив «Информ» может поставить два хорошо апробированных инструмента: программу-отладчик «ГРОТ» и систему ассемблера «МИКРО». Оба продукта снабжаются подробными руководствами пользователя.

Программа «ГРОТ» (версии 2.x) предназначена для отладки программ пользователя в машинных кодах. Коды могут быть получены с помощью какого-либо транслятора. «ГРОТ» также может создавать коды, транслируя вводимые построчно мнемкоды (команды языка ассемблера) или упаковывая строки текстовых сообщений. «ГРОТ» позволяет читать кодовую программу в мнемкодах (обратная трансляция), обеспечивает просмотр всех ячеек памяти, регистров общего назначения и флагов состояния процессора, пошаговое исполнение программы и автоматический прогон (разрешена одна точка останова). Обеспечивается работа с магнитофоном и принтером, поддерживается механизм мнемонического кодирования псевдокоманд арифметики чисел с плавающей точкой (запятой). «ГРОТ» может работать вместе с отлаживаемыми внешними функциями Фокала; внешние функции нормально работают одновременно с программой «ГРОТ».

Имеется также версия «ГРОТ» для ДВК.

Система ассемблера «МИКРО» включает одновременно присутствующие в памяти редактор текста, ассемблер и компоновщик объектных кодовых модулей (формат «МИКРО»). «МИКРО» нужен для подготовки исходных программ на языке ассемблера (разрешены символическое кодирование адресов, арифметические выражения, комментарии).

«МИКРО» также выполняет трансляцию исходных программ в объектные модули, компоновку загрузочных модулей из объектных. Таким образом можно легко создавать программы в кодах, внешние процедуры-функции интерпретатора. Фокал-БК-0010.

Для многих целей полезен отдельный пакет подпрограмм арифметики с плавающей точкой. «Информ» может поставлять и этот программный продукт. Программа-отладчик «ГРОТ» работает с пакетом так же, как и с арифметикой интерпретатора Фокал-БК-0010. Пакет снабжается руководством программиста, в котором детально описаны внутренние подпрограммы Фокала для арифметики с плавающей точкой.

«FOGROT», простой интерфейс кодовых программ для интерпретатора Фокал-БК-0010, позволяет осуществить связь кодовой программы пользователя с программой на Фокале, а также обеспечивает связь с программой-отладчиком «ГРОТ». В руководстве пользователя программы «FOGROT» содержится подробная информация о внутренних подпрограммах интерпретатора Фокала.

Если пользователь имеет в своем распоря-

жении принтер типа ROBOTRON K6311M или K6312M (СМ6329.01М, СМ6329.02М) или «Электроника МС6304» (УВВПЧ-30-04), то для распечатки исходных программ или других текстов ему будет безусловно необходима программа-форматер «UNIFORM». Она позволяет создавать печатные документы на много страниц (без ограничения по количеству, с нумерацией или без). Печатаются тексты, содержащие одновременно русские и латинские буквы, строчные и прописные.

Кооператив «Информ» может выполнить заказы на разработку любых программ в кодах, внешних функций интерпретатора Фокал-БК-0010 и систем внешних функций для получения наилучших решений задач создания эффективного программного обеспечения микрокомпьютера «Электроника БК-0010».

Адрес для переписки: 142292, Московская обл., г. Пущино, микрорайон В, д. 29, кв. 32, «Информ».

Телефоны (московские, с 19.00 до 21.00): 240-08-16, 497-86-52, 367-37-56; 923-74-67 (доб. 2-93), 923-96-68 (доб. 2-93).

Модем для ПЭВМ — реальность

Откликаясь на заметки во втором номере журнала за 1988 г., сообщаем, что специалистами нашего кооператива разработан модем для ПЭВМ. Он обеспечивает подключение ПЭВМ типа «Электроника БК-0010», «Микроша», РК-86 и других к коммутируемым телефонным каналам.

Скорость передачи — 300 бит/с.

Осуществляется защита от ошибок.

В состав программного обеспечения входят базовый драйвер модема, обеспечивающий работу с линией связи, и программа-меню RETRAN, обеспечивающая передачу имеющихся типов файлов, диалог с оператором и чтение-запись на магнитную ленту.

Испытания в пределах Ленинграда на дистанции 35 км показали устойчивую работу. Эксперименты на большем расстоянии не проводились.

Модем акустический, т. е. не требует электрического подключения к телефонной линии,

передача информации осуществляется через телефонную трубку. Санкции Министерства связи СССР на его использование не нужно.

Отметим, что в упомянутой заметке А. Иглицкого технический аспект проблемы выглядит упрощенным в оценке как необходимой квалификации разработчика, так и габаритов и сложности устройства.

Кооператив принимает заказы от организаций и индивидуальных пользователей на изготовление и установку модемов, адаптацию программного обеспечения для конкретных применений и для ПЭВМ других типов, предложения от предприятий по организации крупносерийного производства, предложения по организации банков данных и сетей микро-ЭВМ.

Наш адрес: 191014, Ленинград, а/я 129, кооператив «Компьютер».

Телефон: 274-71-60, с 20.00 до 22.00.

ЧТО? МОЖЕТ ЭВМ

Поговорите с микроскопом

Новый операционный микроскоп западногерманской фирмы «Оптон» обладает многими превосходными качествами. Лучшее тому подтверждение — внимание к нему специалистов МНТК «Микрохирургия глаза». Оставив на этот раз в стороне достоинства оптики и механики, поговорим об электронной начинке.

Еще совсем недавно компьютеры научились разговаривать, распознавание речи до сих пор остается серьезной проблемой, но новый микроскоп уже сегодня освобождает руки медика от управления приборами. Почти все управление ведется устными командами. Микрофон, установленный на тубусе бинокулярного микроскопа, воспринимает речь врача, аналогово-цифровой преобразователь переводит электрические колебания в цифровой вид. Оцифрованная речь запоминается в оперативной памяти. В дело вступает блок распознавания речи: как только команда понята, компьютер подтверждает ее получение, повторяя эту команду своим голосом.

Из управляющих приспособлений осталась только педаль, которая выполняет две функции — включает микрофон и управляет движением, выбранным текущей командой. Микроскоп понимает практически все необходимые команды, такие, как фокусировка, увеличение, смещение поля зре-

ния и еще 10 других команд. Блок распознавания речи умеет адаптироваться к внешним условиям. Например, в экстремальной ситуации в операционном помещении может быть довольно шумно, поэтому каждый сеанс работы начинается с установки рабочего уровня. Стоит повернуть выключатель, как микрокомпьютер попросит пользователя проговорить несколько фраз, чтобы он смог выбрать оптимальный уровень усиления. Как только компьютер справится с задачей, он сообщит об этом врачу, сказав «Спасибо».

Предусмотрено, что с установкой могут работать до 15 человек, например при сменной работе. Чтобы машина уверенно распознавала команды всех врачей, требуется небольшая подготовка — каждый должен наговорить в микрофон все команды, которые умеет выполнять прибор, повторяя команду по три раза. Этого достаточно, чтобы машина сохранила слепки с голоса каждого врача в своей памяти, даже если электронный блок отключат от

на будет общаться с внешним миром, пользователь может заложить сам. Исключение составляют тексты команд, хранящиеся в постоянной памяти. И еще одна деталь — высокая надежность электроники известна, но установка медицинская — любая ошибка опасна, видимо, поэтому на всякий случай имеется дополнительный пульт с управлением джойстиком или педалью.

Найти и обезвредить

Художники, музыканты, программисты и, конечно, писатели во время работы на ЭВМ обязательно сталкиваются с редакторами: графическими, музыкальными, языковыми, текстовыми. Все эти программы облегчают работу профессионалов, вовремя предупреждая их от ошибок.

Но, к сожалению, не создан еще такой редактор, который мог бы следить за смыслом написанного. Для облегчения поиска в готовых программах логических ошибок были созданы отладчики, позволяющие в процессе выполнения программ следить за правильностью ее работы, контролировать содержимое указанных ячеек памяти, значения некоторых переменных и регистров. Кроме того, отладчик позволяет приостанавливать работу программ, изменять величины переменных, используемых ею, исполнять отлаживаемую программу в пошаговом режиме.

Но все эти процедуры естественно замедляют выполнение программ и не дают возможности проверить ее временные характеристики. Особенно это заметно для управляющих программ, от которых требуется мгновенная реакция на поведение управляемого компьютером объекта. Для отладки подобных программ, написанных на языке С без замедления их работы (в реальном масштабе времени), и был разработан отладчик ЕСНО.

Отладчик удобен в работе построен на основе системы «меню» и очень прост для обучения. И хотя ЕСНО не может самостоятельно найти коварные ошибки, он сделает все, чтобы их смог найти даже не искусственный в таких делах начинающий программист.

А рассказал об отладчике журнал «Компьютер дизайн» на английском языке в девятом номере за 1987 г.



сети. Блок синтеза речи знает не только верное произношение команд, он способен по запросу врача вежливо сообщить параметры работы. Немаловажная деталь — фирма снабжает микроскоп блоками, рассчитанными на работу не только с немецким, английским, американским, французским, испанским и японским языками, но и с русским. При необходимости последовательность слов, с помощью которых маши-

Л. ШТЕРНБЕРГ,

канд. физ.-мат. наук

Куйбышевский авиационный институт

Уроки с программируемыми микрокалькуляторами

Перевод алгоритмов на язык микрокалькулятора

После проведения уроков, на которых ученики знакомятся с возможностями программируемого калькулятора, у них возникает естественный вопрос: как связаны программирование в командах калькулятора и запись алгоритмов на алгоритмическом языке? Ответу на него и посвящаются уроки, темы которых можно сформулировать следующим образом:

1. Перевод линейных алгоритмов на язык калькулятора.
2. Перевод ветвлений.
3. Перевод циклов.
4. Отладка программ на калькуляторе.

Первые 3 урока проводятся без калькуляторов, а 4-й является фактически лабораторной работой на калькуляторах.

При изложении этого материала учитель должен подчеркнуть, что любая ЭВМ понимает весьма своеобразный язык машинных команд, несколько похожий на коды калькулятора. А программа на языке программирования после ее ввода в ЭВМ переводится на язык машинных команд и только затем выполняется. Процесс перевода называется трансляцией. Так как калькулятор — это очень маленькая ЭВМ, то перевод с алгоритмического языка приходится выполнять вручную. Правила этого перевода и составляют содержание трех уроков.

Отметим, что наличие любой вычислительной техники (калькуляторов в том числе) заставляет учителя отойти от плана, рекомендованного для безмашинного варианта курса.

Ряд уроков необходимо отвести для работы с калькуляторами (или ЭВМ), поэтому часть материала IX класса откладывается до X класса, где имеется большой резерв времени, так как материал учебника X класса рассчитан на 34 часа, а на курс отведено 68 часов.

Урок «Перевод линейных алгоритмов» лучше всего провести сразу после изучения структуры калькулятора, урок «Перевод ветвлений» — после того, как рассмотрены разветвляющиеся алгоритмы обработки величин, т. е. где-то в конце 2-й четверти, уроки «Перевод циклов» и «Отладка программ» — в середине 3-й четверти, после рассмотрения циклических алгоритмов, когда на калькуляторе можно будет просчитать практически полезные программы, такие, как вычисление площади криволинейной трапеции, вычисление многочлена, определение корня методом деления пополам и т. д.

Перед тем как начать изложение правил перевода, следует рассмотреть, какие элементы калькулятора могут служить аналогами элементов алгоритма. Переменная величина может принимать разные значения, на калькуляторе могут принимать разные значения регистры, значит, переменным алгоритма должны соответствовать регистры. Все регистры в арифметических операциях работают одинаково, значит, любой из них может быть взят для переменной типа цел и вещ, а переменные типа лит обеспечить регистром невозможно: литерные величины на калькуляторе не обрабатываются. Команде (предписанию) присваивания из алгоритма соответствует серия команд калькулятора, так как исполнитель Калькулятор понимает более мелкие команды, чем исполнитель, понимающий алгоритмический язык.

Правила перевода алгоритмов удобно рассказывать, параллельно иллюстрируя их на каком-либо алгоритме, а затем сформулировав в явном виде. Они сводятся к правилам перевода отдельных команд алгоритмического языка (чтобы не путаться, мы будем говорить «предписание» алгоритмического языка и «команда» калькулятора). Проиллюстрируем их на примере алгоритма вычисления косинуса угла треугольника по заданным длинам трех его сторон:

алг косинус угла (вещ a, b, c , кос),

арг a, b, c ; рез кос;

нач кос: = $(b^2 + c^2 - a^2) / (2 \cdot b \cdot c)$;

кон;

Правила перевода:

1. Заголовок алгоритма, предписания арг, рез, нач не переводятся.

2. По описаниям типов переменных (вещ, цел и т. д.), стоящим как в заголовке, так и после нач, составляется таблица, в которой каждой переменной ставится в соответствие адресуемый регистр. Регистры можно распределять произвольно, рекомендуется брать подряд.

Для нашего алгоритма: $a - P0$; $b - P1$, $c - P2$; кос — $P3$.

3. Присваивания переводятся очевидным образом. Составленная таблица распределения регистров показывает, откуда брать значение и куда помещать результат. Если понадобятся регистры для запоминания промежуточных результатов, то их берут из числа свободных.

4. кон переводится командой «С/П».

Программы рекомендуется всегда составлять с нулевого адреса (даже не объясняя ученикам возможность их написания с других адресов) и записывать в виде таблицы «адрес — команда — код — алг. язык — комментарий», как это сделано ниже. Столбец «код» заполняется после написания программы.

Для выполнения программы надо занести значение аргумента в $P0, P1, P2$ соответственно и нажать клавиши «В/О» и «С/П». После остановки считать результаты из регистров, отведенных под результаты. В нашем случае, так как результат один и он вычисляется в самом конце, он при остановке окажется на индикаторе, но так будет не всегда.

Однако калькулятор может помочь нам в занесении аргументов и выдаче результатов: он может сам считать значения в память или выдать их на индикатор. Это одно из основных действий калькулятора, которое нельзя выразить через другое, поэтому для описания

Адрес	Код	Команда	Алгоритмический язык
		$a - P0$	
		$b - P1$	
		$c - P2$	<u>алг</u> косинус угла
		кос — $P3$	(<u>вещ</u> a, b, c , кос);
00	61	$P - x 1$	кос: =
01	22	$F x^2$	b^2
02	62	$P - x 2$	+
03	22	$F x^2$	c^2
08	44	$x - P 4$	
09	02	2	/ (2
10	61	$P - x 1$	$\cdot b$
14	41	$P - x 4$.
15	13	—	.
16	23	$F 1/x$.
17	43	$x - P 3$.
18	50	С/П	<u>кон</u>

таких действий в язык надо ввести специальные средства: предписания ввод и вывод.

Предписание «ввод список переменных» — это запрос в процессе выполнения программы значений указанных переменных, а предписание «вывод список выражений» — это запрос в процессе выполнения программы. Ввод и вывод можно сочетать с аргументами и результатами.

Правила их перевода таковы:

1. Конструкция ввод программируется командами « $x - P$ » с номерами регистров, соответствующими перечисленным после ввод переменным, перед каждой из которых, кроме первой, ставится «С/П»;

2. Вывод программируется командами вычисления соответствующих выражений, после каждого выражения, кроме последнего, ставится «С/П».

С применением этих конструкций алгоритм принимает вид:

алг косинус; (аргументов, результатов нет, все переменные — вспомогательные) (запрашиваем значения a, b, c)
нач вещ a, b, c ;
ввод a, b, c ;
вывод $(b^2 + c^2 - a^2) / (2 \cdot b \cdot c)$
кон

Программа выглядит так (распределение регистров прежнее):

У школьников может возникнуть вопрос: почему, нарушая правила формального перевода, можно получить более короткую или быструю программу? Ответ на этот вопрос таков: мы пользуемся очень простым алгорит-

Адрес	Код	Команда	Алгоритмический язык	Комментарий
00	40	х—П 0	<u>ввод</u> a	Запоминаем набранное перед пуском значение a .
01	50	С/П	b	При этом останове набирается b , после пуска оно запоминается.
02	41	х—П 1		
03	50	С/П	c	Этот останов нужен для набора c , значение c запоминается, далее начинается вычисление.
04	42	х—П 2		
05	61	П—х 1		
06	22	F x^2		
...	...			
23	50	С/П		Когда выражение вычислено, команда С/П позволяет увидеть его значение на индикаторе.

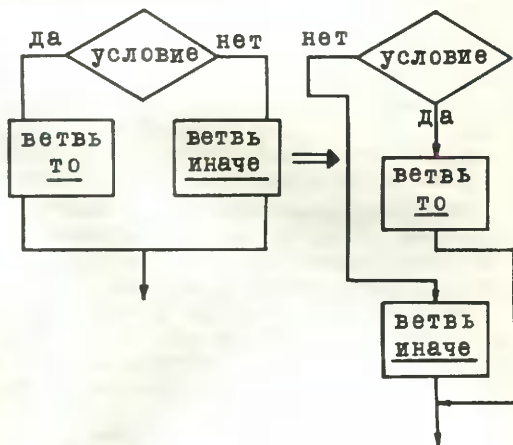
мическим языком, в котором нет ряда полезных возможностей. Если усложнить язык, то на нем можно будет описать все нюансы, дающие при переводе оптимальные программы.

Следующий «калькуляторный» урок проводится после изучения конструкции «если-то-иначе-все» и решения ряда задач с использованием этой конструкции применительно к алгоритмам работы с величинами.

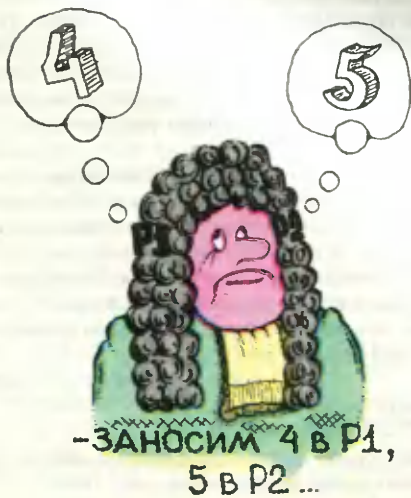
Объяснение команд условного и безусловного переходов должно занимать около 10 мин. Здесь следует объяснить, что эти команды засылают в счетчик адреса значение, хранящееся во втором слове команды, подменяя то значение, которое было в счетчике адреса ранее. Тем самым происходит переход к выполнению команды с указанным адресом. Следует упомянуть, что команды условных переходов выполняются только в программном режиме и что в командах переходов нельзя исправить только адрес: надо повторно занести и команду.

Далее, перерисовав схему ветвления так, как показано на рис. 1, легко объяснить, что ромбу, из которого имеется выход либо даль-

ше, либо в обход части программы, соответствует команда условного перехода (позволяющая идти либо дальше, либо в обход части программы), а ломаной стрелке, обходящей



часть «иначе», соответствует безусловный переход. Это объяснение позволяет сформулировать правила перевода ветвлений.



1. Программируется условие, приведенное к имеющемуся на калькуляторе сравнению с нулем (т. е., например, вместо $a \leq b$ рассматривается условие $b - a \geq 0$).

2. Пишется нужная команда условного перехода, за которой оставляется пустая ячейка для адреса, который выяснится позже.

3. Программируется ветвь «то».

4. Пишется команда «ВП», и оставляется свободная ячейка для адреса, который также выяснится позже.

5. Программируется ветвь «иначе».

6. Когда встречается «все», пропущенный на шаге 2 адрес заполняется адресом начала ветви «иначе», а пропущенный на шаге 4 адрес — адресом очередной свободной ячейки.

Эта часть урока требует также около 10 мин. Затем правила перевода иллюстрируются на конкретном примере, например на классическом: выбор максимума из двух чисел. Пусть a находится в $P0$, b — в $P1$, максимум M размещается в $P2$. Тогда тело алгоритма и программы примет вид (фрагмент программы совершенно произвольно размещен с адреса 05):

«К НОП», заносятся исходные данные, и команда выполняется в пошаговом режиме 2 раза: при $a < b$ и при $a \geq b$. После выполнения команды 07 нажатием «F ПРГ» получаем возможность взглянуть на счетчик адреса: на индикаторе «11 61 60 08» — последние 3 команды и номер команды, которая будет выполняться. Возвращаемся в автономный режим («F АВТ»), выполняем одну команду («ПП»)

Адрес	Код	Команда	Алгоритмический язык	Комментарий
05	60	П—x 0	<u>если</u>	Программируем условие, приведенное к сравнению с нулем.
06	61	П—x 1	$a-b$	
07	11	—		Адрес выяснится позже.
08	59	$Fx \geq 0$	≥ 0	
09	□	□	<u>то</u>	
10	60	П—x0	$M:=a$	
11	42	x—П2		И этот адрес выяснится позже.
12	51	БП	<u>иначе</u>	
13	□	□		И этот адрес выяснится позже.
14	61	П—x 1	$M:=b$	
15	42	x—П 2		Вот сейчас ячейку 09 можно заполнить адресом 14, а ячейку 13 — адресом 16.
16	<u>все</u>	

Обратите внимание учеников на строгое соответствие элементов алгоритмического языка и отчеркнутых частей программы. Полезно также рисовать стрелки переходов.

Если останется время (или же на следующем уроке), можно поработать с этой программой на калькуляторе. Для этого программа дополняется в ячейках 00—04 командами

и снова смотрим на счетчик адреса («F ПРГ»), где при $a < b$ видим «17 51 42 15», т. е. выполнен переход, а при $a \geq b$ видим «15 59 11 10», т. е. перехода не произошло.

Отметим, что этот фрагмент можно написать существенно короче, если пользоваться более развитым алгоритмическим языком.

В качестве домашнего задания можно предложить написать программу вычисления и максимума и минимума или значения функции, определяемой по разным формулам при аргументе больше нуля и меньше нуля соответственно.

На этих уроках не следует затрагивать вложенные ветвления, а также сложные условия с союзами и и или — этот материал более тяжел для понимания, а потому должен быть отложен на более позднее время, когда учащиеся лучше освоятся с калькуляторами.

Работа с циклами будет рассмотрена в следующей статье.

В завершение отметим, что статьи данного цикла предполагают, что учитель знаком с работой на ПМК, и потому в них рассматриваются только методические вопросы. Подробное изложение программирования на ПМК, техники перевода с алгоритмического языка на примерах задач школьного учебника дано в [1].

Продолжение следует.

Литература

1. Штернберг Л. Ф. Программирование на микрокалькуляторе. М.: Просвещение, 1988.



Школьная информатика глазами первокурсников

Начиная с 1987/88 учебного года преподавание вузовских курсов, связанных с программированием для ЭВМ, должно было начаться не с нуля, как в предшествующие годы, а с учетом знаний, полученных школьниками в результате изучения предмета «Основы информатики и вычислительной техники». Мы поставили перед собой задачу определить уровень знаний школьников, решение которой разбили на два этапа: подготовительный и основной. Подготовительный состоял в анкетировании, которое было проведено на одном из первых занятий с первокурсниками факультета прикладной математики и механики (ПММ) Воронежского государственного университета. Анкета содержала общие вопросы, касающиеся организации преподавания курса информатики в школе и отношения к нему первокурсников.

В анкетировании принимали участие 142 человека, окончивших среднюю школу в 1987 г.: 56,34 % выпускников школ г. Воронежа, 4,22 % — других городов, 39,44 % опрошенных окончили школы Воронежской или близлежащих областей. Анонимная форма анкетирования позволила, по мнению самих первокурсников, быть более раскованными, отвечать честно, «без оглядки».

В одном из вопросов анкеты предлагалось определить место информатики в системе школьных дисциплин. Диапазон ответов был огромен: от первого до последнего. Большинство (52,11 %) ответили информатике 4—5-е место.

Все без исключения отмечали, что хорошему усвоению информатики мешает отсутствие свободного доступа к компьютерной технике. Здесь студенты безусловно правы.

В анкете спрашивалось о связи информатики с преподаванием других школьных дисциплин. Как самостоятельную дисциплину, не связанную с другими предметами, информатику называли 7,74 % первокурсников. Подавляющее большинство студентов (67,6 %) указали на связь информатики с математикой и физикой. Полезность использования информатики во всех школьных предметах подчеркнули 6,33 % первокурсников, а 1,4 % определили ее как «царицу всех наук».

Комментируя эти цифры, можно отметить, что они зависят как от подбора задач и упражнений, которые решались на уроках информатики, так и от отношения учителей

других школьных дисциплин к особому подходу при изучении явлений и формировании прочных знаний, умений и навыков, который называется *алгоритмическим*. На наш взгляд, возможности реализации алгоритмического подхода имеются в большинстве предметов средней школы.

Школьную информатику назвали своим любимым предметом 14,08 % первокурсников (математику — 52,11 %, физику — 9,86 %). Однако после первого года обучения в университете курс «ЭВМ и программирование» отметили как любимый 61,97 % студентов.

В этой связи интересно следующее. На собеседованиях с абитуриентами у будущих студентов, как правило, спрашивают, чем обоснован их выбор специальности. Большинство из поступающих на факультет ПММ связывают выбор с бурным развитием компьютерной техники, без которой немислим научнотехнический прогресс, с безграничными возможностями применения ЭВМ, с будущей интересной работой, с советами родителей, родственников, друзей и знакомых. Несмотря на то что все абитуриенты факультета имеют в аттестате по информатике оценки «4» и «5», лишь около 10 % из них объясняют свое желание обучаться на факультете ПММ интересом к этому предмету в школе.

Ниже приведен процент первокурсников, по мнению которых указанные разделы изучались в школьном курсе информатики:

1 Алгоритм и его свойства	97,18
2 Промежуточные величины	
Присваивание значений	95,77
3 Команда ветвления	95,77
4 Команда выбора	64,79
5 Алгоритмы работы с литературными величинами	49,29
6 Команды повторения	46,48
7 Линейные таблицы	56,34
8 Прямоугольные таблицы	49,29
9 Вспомогательные алгоритмы	46,48
10 Вспомогательные алгоритмы-функции	28,87
11 Алгоритмы работы с графической информацией	36,62
12 Работа с калькулятором	70,42
13 Устройство ЭВМ	52,11
14 Язык программирования Бейсик	64,79
15 Роль ЭВМ в современном обществе. Перспективы развития вычислительной техники	42,25

По результатам ответов студентов только 9,86 % из них смогли получить доступ к ЭВМ и отладить программу, а 7,74 % сделали это на различного рода факультативных и кружковых занятиях.

В качестве причины, из-за которой курс информатики изучен не полностью, немалая часть студентов отмечали замену информатики, по которой нет экзамена, уроками математики или физики (в зависимости от того, какой учитель вел занятия по информатике — математик или физик).

В большинстве школ практическая часть курса ОИВТ свелась к работе с микрокалькулятором.

Основной этап работы по определению уровня знаний по информатике включал ответы студентов на несложные вопросы по разделу курса ОИВТ «Алгоритмический язык». Нас интересовал именно этот раздел, так как составление программы на любом языке программирования начинается с раз-

работки алгоритма ее решения на алгоритмическом языке, элементы которого изучались в школе.

Дать точное определение алгоритма смогли лишь 6,33 % первокурсников, написать его общую структуру — 19,01 %, вспомнить схему работы команды повторения — 28,87 %, условного оператора — 31,69 %, команды выбора — 11,27 %, привести пример вспомогательного алгоритма — 7,04 %; как работать с таблицей исполнения алгоритма, знают только около 10 %.

Разумеется, необходимо сделать скидку на то, что студенты заранее не готовились к ответам на вопросы по школьному курсу информатики и из-за почти полугодового перерыва в учебе многое могли забыть. Но тем не менее, как показывают приведенные цифры, вузовский курс «ЭВМ и программирование» пока не имеет прочной школьной основы и его изучение приходится начинать практически заново.

А. АБДУКАДЫРОВ, М. ИСТОЧНИКОВА, Е. ВАЙНШТЕЙН

Компьютерный кроссворд в учебных целях

Проверка знаний — важная составная часть процесса обучения. Ее основной целью является определение качества усвоения учащимися программного материала.

Мы предлагаем проверять знания учащихся в игровой форме — с помощью кроссворда.

Составленный кроссворд относится к школьному курсу «Основы информатики и вычислительной техники» и составлен по теме «Принципы устройства ЭВМ». В него вошли все основные вопросы.

1. Центральное устройство ЭВМ.
2. Устройство для ввода информации.
3. Устройство для записи и чтения информации с внешнего носителя.
4. Минимальная единица информации.
5. Метка на экране, указывающая место для ввода символа.
6. Номер ячейки памяти.
7. Машина для автоматической обработки информации.
8. Печатающее устройство.
9. Кратковременное запоминающее устройство.
10. 8 битов.
11. Внешний носитель информации.
12. Устройство вывода информации для восприятия человеком.

Т.е. рассматриваются основные узлы машины.

В начале работы с программой дается инструкция. После нее на экран дисплея выводится оглавление. После можно перейти непосредственно к разгадыванию кроссворда или справочнику, а при необходимости можно снова обратиться к инструкции.

Наша программа построена так, что можно просмотреть справочник, в котором объясняются все термины, используемые в кроссворде. При составлении справочника были частично использованы фрагменты программ Новосибирского и Свердловского педагогических институтов.

После просмотра справочника на экране дисплея появится кроссвордная сетка, а в верхней части экрана — вопрос.

Ответ на данный вопрос фиксируется, проверяется, а затем при правильном ответе печатается в сетке кроссворда, иначе представляются еще дополнительные попытки ответа. Всего может быть сделано три попытки ответа на один вопрос. Если же правильный ответ так и не был набран, то осуществляется переход к следующему вопросу. После первой же неудачной попытки предлагается воспользоваться справочником.

Все это хорошо отражает логическая схема



1

тематической проверки (рис. 1).

Хотелось бы отметить, что если при наборе с клавиатуры ответа учеником была допущена ошибка, то ее устранить, воспользовавшись специальной клавишей. Если учеником использованы все попытки ответа, а правильный так и не был набран, то он автоматически печатается в сетке кроссворда. Это очень удобно, так как ученик сразу получает правильную информацию и в следующий раз при ответе на такой же вопрос не допустит ошибки.

После того как ученик ответит (правильно или нет) на все предложенные вопросы (на рис. 2 показан уже заполненный кроссворд), на дисплей выводится протокол, в котором указываются количество вопросов, количество правильных ответов, количество неправильных ответов, количество ответов с первой попытки.

После этого машина выставляет ученику оценку.

Таким образом, учащийся видит благодаря протоколу, в соответствии с какими критериями ему была выставлена оценка.

Объективная оценка может поддерживать,

подбодрить ученика. Это все осуществимо при машинном контроле.

Кроссворды можно составлять и для проверки знаний основных терминов по другим курсам (алгебре, геометрии, физики, химии, географии, анатомии, биологии и т. д.).

Методика составления таких кроссвордов очень проста.

Эту программу можно использовать не только при контроле знаний учащихся, но и при объяснении темы «Устройство и принципы работы ЭВМ», можно воспользоваться ею и при закреплении материала темы, для чего нужно обратиться только к справочнику, в котором доступно и наглядно отображены все основные узлы ЭВМ.

Данная программа реализована на «Ямахе», но такие программы можно составлять и для других машин, имеющих графику.

Такой вид проверки знаний учащихся удобен и интересен: самостоятельная работа учащихся по разгадыванию кроссворда оживляет учебный процесс, а также дает возможность преподавателям проверять одновременно знания у всего класса или у отдельной группы учащихся по его усмотрению.

81



2

Эта программа была апробирована на уроках информатики в X классе школы № 110 Ташкента и показала эффективность применения учебных программ такого типа.

Эксперимент был положительно оценен преподавателями школы и института.

Чтобы разнообразить учебный процесс, можно создавать обучающе-контролирующие программы не только с использованием кроссворда, но и применяя всевозможные криптограммы, чайнворды, ребусы и другие игры.

Е. КРЫЛОВА, М. ПЕТРОВ

III Московская олимпиада по информатике

Решения

Школьный тур

IX класс

1—IX—Ш **Идея решения:** последовательно проверить, совпадают ли знаки первых координат точек, затем — знаки вторых координат точек; для проверки воспользоваться соображением, что знаки двух чисел совпадают тогда и только тогда, когда их произведение положительно.

алг КВАДРАНТ (вещ X1, Y1, X2, Y2,
лит ответ)

арг X1, Y1, X2, Y2

рез ответ

нач

ответ:=«нет»

если X1*X2>0

то если Y1*Y2>0

то ответ:=«да»

все

все

кон

2—IX—Ш **Идея решения:** в цикле увеличивать количество курьеров, складывая их число с самим собой.

алг ГОГОЛЬ (цел N)

арг

рез N

нач цел I

N:=1

I:=1000

пока I<30 000

нц N:=N+1

I:=I+I

кц

кон

3—IX—Ш **Идея решения:** пытаемся набрать нужную сумму прямым перебором купюр (три вложенных цикла).

алг РАЗМЕН (лит ответ)

арг

рез ответ

нач цел I, J, K

I:=0

ответ:=«нет»

пока I<=10 и ответ=«нет»

нц J:=0

пока J<=10 и ответ=«нет»

нц K:=0

пока K<=10 и ответ=«нет»

нц если I+3J+5K=25 и I+

+J+K=10

то ответ:=«да»

все

K:=K+1

кц

J:=J+1

кц

I:=I+1

кц

кон

X класс

1—X—Ш **Идея решения:** дневную «производительность труда» бригады богатырей можно посчитать по формуле $3 \cdot 24 \cdot B$, где B — число богатырей; надо увеличивать в цикле их количество до тех пор, пока не будет превзойдено число S, равное количеству голов всех З. Горынычей.

алг МУРОМЕЦ (цел ЗГ, B, цел таб ГОЛ
[1:ЗГ])

арг ЗГ, ГОЛ

рез B

нач цел I, S

S:=0

для I от 1 до 3Г

нц S:=S+ГОЛ[I]

кц

Б:=0

пока 3*24*Б<=S

нц

Б:=Б+1

кц

кон

2—X—III Идея решения: вычислить длину трех отрезков, используя вспомогательный алгоритм, и применить неравенство треугольника.

алг ПРЯМАЯ (вещ X1, Y1, X2, Y2, X3, Y3, лит ответ)

арг X1, Y1, X2, Y2, X3, Y3

рез ответ

нач вещ D1, D2, D3

ответ:="нет"

ОТРЕЗОК (X1, Y1, X2, Y2, D1)

ОТРЕЗОК (X2, Y2, X3, Y3, D2)

ОТРЕЗОК (X3, Y3, X1, Y1, D3)

если D1=D2+D3 или D2=D1+D3

или D3=D1+D2

то ответ:="да"

все

кон

алг ОТРЕЗОК (вещ A1, A2, B1, B2, D)

арг A1, A2, B1, B2

рез D

нач

D:=SQR((A2-A1)**2+(B2-B1)**2)

кон

3—X—III Идея решения: попарно сравниваем элементы двух массивов, начиная с наименьших, заполняем очередной элемент создаваемого массива меньшим из этих элементов; оставшийся элемент одного из исходных массивов сравниваем со следующим элементом второго исходного массива. В случае, если один из массивов окажется исчерпанным, приписать к создаваемому массиву оставшиеся элементы второго (уже упорядоченного по условию) массива.

Предложенный алгоритм не является оптимальным, возможны лучшие решения.

алг СЛИЯНИЕ (цел N, вещ таб A[1:N], B[1:N], C[1:2*N])

арг N, A, B

рез C

нач цел I, J, K

I:=1

J:=1

K:=0

пока I<=N и J<=N

нц K:=K+1

если A[I]<B[J]

то C[K]:=A[I]

I:=I+1

иначе C[K]:=B[J]

J:=J+1

все

кц

если I>N

то I:=J

пока I<=N

нц K:=K+1

C[K]:=B[I]

кц I:=I+1

иначе J:=I

пока J<=N

нц K:=K+1

C[K]:=A[J]

J:=J+1

кц

все

кон

Районный тур

IX класс

I—IX—P Идея решения: реализуется стандартный алгоритм нахождения наибольшего и наименьшего значений непрерывной функции на замкнутом промежутке с учетом того обстоятельства, что квадратный трехчлен имеет единственный экстремум, который вычисляется как $-B/(2*A)$.

алг МИНИМАКС (вещ A, B, C, X1, X2, МАКС, МИН)

арг A, B, C, X1, X2

рез МАКС, МИН

нач вещ X3, Y1, Y2, Y3

X3:=-B/(2*A)

Y1:=A*X1*X1+B*X1+C

Y2:=A*X2*X2+B*X2+C

Y3:=A*X3*X3+B*X3+C

если X3<=X1 или X3>=X2

то если Y1<Y2

то МИН:=Y1

МАКС:=Y2

иначе МИН:=Y2

МАКС:=Y1

все

иначе если Y1<Y2 и Y1<Y3

то МИН:=Y1

если Y2<Y3

то МАКС:=Y3

иначе МАКС:=Y2

все

иначе если Y1>Y2 и

Y2<Y3

то МИН:=Y2

если Y1<Y3

то МАКС:=Y3

иначе

МАКС:=Y1

все
 иначе МИН:=У3
 если У1<У2
 то
 МАКС:=У2
 иначе
 МАКС:=У1
все

нц Р:=J1*1000+J2*100+J3*10+J4
 если Р=(J1+J2+J3+J4)**3
 то ОТВЕТ:=«СУЩЕСТВУЕТ»
все
 J4:=J4+1

кц
 J3:=J3+1

кц
 J2:=J2+1

кц
 J1:=J1+1

кц

кон

все
кон 2—IX—Р Идея решения: сначала они должны укладывать коврики вдоль одной из стен, пока не уложат весь ряд целиком, даже если придется последний коврик положить внахлест, затем, смещаясь относительно другой стены, повторять описанную процедуру уже для рядов ковриков.
 алг КОВРИКИ (вещ А, В, С, цел S)

арг А, В, С

рез S

нач вещ L, M

S:=0

L:=0

пока L<=A

нц L:=L+C

M:=0

пока M<=B

нц S:=S+1

M:=M+C

кц

кц

кон

3—IX—Р Идея решения: четыре вспомогательные переменные последовательно принимают значения цифр так, чтобы из них формировались возрастающие четырехзначные числа. Для каждого из них проверяется выполнение условия задачи. Алгоритм заканчивает работу либо при окончании полного перебора, либо при получении положительного ответа.

алг ЧИСЛА (лит ОТВЕТ, цел Р)

арг

рез ОТВЕТ, Р

нач цел J1, J2, J3, J4

ОТВЕТ:=«НЕ СУЩЕСТВУЕТ»

J1:=1

пока J1<=9 и ОТВЕТ=«НЕ СУЩЕСТВУЕТ»

нц J2:=0

пока J2<=9 и ОТВЕТ=«НЕ СУЩЕСТВУЕТ»

нц J3:=0

пока J3<=9 и ОТВЕТ=«НЕ СУЩЕСТВУЕТ»

нц J4:=0

пока J4<=9 и ОТВЕТ=«НЕ СУЩЕСТВУЕТ»

Возможен и другой подход к решению: перебирать числа, начиная с 10, пока их кубы будут оставаться меньше 10000, в их кубах выделять цифры и, найдя их сумму, проверить условие задачи. Этот способ технически сложнее, но дает более эффективный алгоритм.

4—IX—Р Идея решения: робот смещается на восток до первой закрашенной клетки, после чего возвращается на один шаг и закрасивает последнюю чистую клетку из тех, что им пройдены, затем смещается на восток по всем закрашенным клеткам, пока не встретит чистую, и закрасивает ее и т. д. При каждом смещении на восток робот проверяет, достиг ли он уже конца коридора.

Программа для робота.

1. ШАГ НА ВОСТОК
2. КЛЕТКА ЗАКРАШЕНА? (8)
3. ШАГ НА ВОСТОК
4. КОНЕЦ КОРИДОРА?
5. КЛЕТКА НЕ ЗАКРАШЕНА? (3)
6. ШАГ НА ЗАПАД
7. ЗАКРАСИТЬ КЛЕТКУ
8. ШАГ НА ВОСТОК
9. КОНЕЦ КОРИДОРА?
10. КЛЕТКА ЗАКРАШЕНА? (8)
11. ЗАКРАСИТЬ КЛЕТКУ
12. КЛЕТКА ЗАКРАШЕНА? (3)

Красить одну и ту же клетку дважды не запрещается (в противном случае задача резко усложнилась бы). В предложенном решении на 12-м шаге условие обязательно выполняется; это фактически безусловный переход. Чтобы избежать выхода робота из коридора в случае, когда первая же клетка на его пути окажется закрашенной, он начинает выполнять программу не с основного цикла, описанного в идее решения, а со смещения до первой чистой клетки.

Х класс

1—X—Р Идея решения: вычислить для каждой вершины координаты вектора, выхо-

дящего из нее к следующей (по часовой стрелке) вершине, и координаты вектора, выходящего из следующей вершины, найти их скалярное произведение и, если оно равно нулю, увеличить на единицу количество найденных прямых углов.

Чтобы организовать перебор вершин К-угольника в цикле, добавим координаты вершин $K+1$ и $K+2$, совпадающие с координатами вершин 1 и 2.

алг МНОГОУГОЛЬНИК (цел К, ПРЯМУГЛЫ, вещь таб Х[1:K+2], Y[1:K+2])

арг К, Х, Y

рез ПРЯМУГЛЫ

нач цел J, вещь СХ1, СУ1, СХ2, СУ2,

СКАЛПР

ПРЯМУГЛЫ:=0

для J от 1 до К

нц СХ1:=X[J+1]-X[J]

СУ1:=Y[J+1]-Y[J]

СХ2:=X[J+2]-X[J+1]

СУ2:=Y[J+2]-Y[J+1]

СКАЛПР:=СХ1*СХ2*СУ1*СУ2

если СКАЛПР=0

то ПРЯМУГЛЫ:=ПРЯМУГЛЫ+1

все

кц

кон

2—X—P Идея решения: исходные данные — результаты в толчке и рывке, собственный вес спортсмена — заносятся в двумерный вещественный массив А, представляющий собой матрицу $3*N$, где N — число спортсменов. Затем дважды используется вспомогательный алгоритм ЧЕМПИОН, который выбирает чемпиона, т. е. формирует столбец номеров спортсменов, где на первом месте — номер чемпиона. Если чемпионов двое или трое, то их номера приводятся на первых двух, трех местах и т. д., а остальные места заняты нулями, так как по условию задачи другие результаты не зачитываются. Таким образом определяются победители в отдельных видах. Затем формируется массив $1*N$, в каждую строчку которого заносятся суммы результатов каждого спортсмена в обоих упражнениях, и полученная матрица, представляющая собой один столбец, также обрабатывается алгоритмом ЧЕМПИОН для определения абсолютного чемпиона. Во вспомогательном алгоритме производится выбор не просто номера наибольшего результата, но, в случае равенства некоторых из этих результатов, учитывается собственный вес спортсменов. Возможны совпадения и результатов, и собственных весов, поэтому ведется учет количества чемпионов (J1) и сохраняются все их номера (ЧЕМП [J1]).

Задачу в данной постановке решает алгоритм, который не расставляет всех спортсменов по занятым ими местам, а только выбирает чемпиона; однако на реальных соревнованиях требуется, разумеется, ранжировать всех спортсменов. Можно повысить оценку за решение задачи в такой, более полной, постановке (для такого решения можно упорядочить каждый столбец матрицы, из которого в предложенном решении только делается выборка, сформировать столбец номеров спортсменов, показавших соответствующие результаты, не забыв при этом учесть их собственный вес).

алг ШТАНГИСТЫ (цел N, вещь таб

A[1:3, 1:N], цел таб

ВИД1[1:N], ВИД2[1:N],

АВС[1:N])

арг N, A

рез ВИД1, ВИД2, АВС

нач цел J, вещь таб В[1:N]

ЧЕМПИОН (N,1,A,ВИД1)

ЧЕМПИОН (N,2,A,ВИД2)

для J от 1 до N

нц В[1,J]:=A[1,J]+A[2,J]

кц

ЧЕМПИОН (N,1,В,АВС)

кон

алг ЧЕМПИОН (цел КОЛИЧ, СТОЛЬ,

вещ таб ДАННЫЕ[1:3,1:КОЛИЧ],

цел таб ЧЕМП[1:КОЛИЧ])

арг КОЛИЧ, СТОЛЬ, ДАННЫЕ

рез ЧЕМП

нач цел J, J1, J2, вещь РЕЗУЛЬТАТ, ВЕС

ЧЕМП[1]:=1

J1:=1

для J2 от 2 до КОЛИЧ

нц ЧЕМП[J2]:=0

кц

РЕЗУЛЬТАТ:=ДАННЫЕ[СТОЛЬ,1]

ВЕС:=ДАННЫЕ[3,1]

для J от 2 до КОЛИЧ

нц выбор

при ДАННЫЕ[СТОЛЬ,J]>РЕ-

ЗУЛЬТАТ:

РЕЗУЛЬТАТ:=ДАН-

НЫЕ[СТОЛЬ,J]

ВЕС:=ДАННЫЕ[3,J]

ЧЕМП[1]:=J

для J2 от 2 до КОЛИЧ

нц ЧЕМП[J2]:=0

кц

при ДАННЫЕ[СТОЛЬ,J]=РЕ-

ЗУЛЬТАТ:

если ДАННЫЕ[3,J]<

<ВЕС

то РЕЗУЛЬТАТ:=

ДАННЫЕ[СТОЛЬ,

J]

ВЕС:=ДАННЫЕ

[3,J]

```

ЧЕМП[1]:=J
для J2 от 2 до
КОЛИЧ
нц ЧЕМП[J2]:=0
кц
иначе если ДАННЫЕ
[3,J]=ВЕС
то J1:=J1+1
ЧЕМП
[J1]:=J

```

все

все

кц

кон

3—X—P Пусть состав находится в А, паровоз в Т, обе стрелки переключены на тупик, нумерация вагонов начинается от Т.

Для написания программы ДИСПЕТЧЕР кроме предложенных в условии команд используем также процедуры ПРИСОЕДИНИТЬ и ПЕРЕДВИНУТЬ.

Процедура ПРИСОЕДИНИТЬ $W_i - W_j$:
 РАСЦЕПИТЬ W_j И W_{j+1} ;
 СЦЕПИТЬ S И W_i .

Процедура ПЕРЕДВИНУТЬ $W_i - W_j$:
 ПЕРЕМЕСТИТЬСЯ В А;
 ПРИСОЕДИНИТЬ $W_i - W_j$;
 ПЕРЕМЕСТИТЬСЯ В Т;
 ПЕРЕМЕСТИТЬСЯ В В;
 РАСЦЕПИТЬ S И W_i ;
 ПЕРЕМЕСТИТЬСЯ В Т.

Программа ДИСПЕТЧЕР:
 ПЕРЕМЕСТИТЬСЯ В А;
 ПРИСОЕДИНИТЬ $W_1 - W_2$;
 СТРЕЛКИ В АВ;
 ПЕРЕМЕСТИТЬСЯ В В;
 СТРЕЛКИ В Т;
 ПЕРЕМЕСТИТЬСЯ В Т;
 РАСЦЕПИТЬ S И W_1 ;
 ПЕРЕДВИНУТЬ $W_3 - W_{11}$;
 ПЕРЕДВИНУТЬ $W_{12} - W_{20}$;
 ПРИСОЕДИНИТЬ $W_1 - W_2$;
 ПЕРЕМЕСТИТЬСЯ В А;
 СТРЕЛКИ В АВ;
 ПЕРЕМЕСТИТЬСЯ В В;
 СЦЕПИТЬ W_2 И W_3 ;
 СЦЕПИТЬ W_2 И W_{12} .

4—X—P

алг МАКСПОДСТРОКА (лит СТР1, СТР2, МСТР)

арг СТР1, СТР2

рез МСТР

нач цел I, J, K, лит СТР

МСТР:=«»

I:=1

пока I < ДЛИН(СТР1) — ДЛИН(МСТР)

нц J:=1

пока J < ДЛИН(СТР2) — ДЛИН(МСТР)

нц K:=0

СТР:=«»

пока СТР1[I+K:I+K]=

=СТР2[J+K:J+K] и

I+K <= ДЛИН(СТР1)

и J+K <= ДЛИН

(СТР2)

нц СТР:=СТР+СТР1[I+K:

I+K]

K:=K+1

кц

если K > ДЛИН(МСТР)

то МСТР:=СТР

все

J:=J+1

кц

I:=I+1

кц

кон

Городской тур

IX класс

1—IX—Г Возможный алгоритм:

0. Начать

1. Ввод A и B

2. Проверить условие ($A=0$ или $B=0$); если верно, то идти на шаг 9.

3. Проверить условие $B > 0$; если верно, то идти на шаг 6.

4. Проверить условие $A > 0$; если верно, то (вывести «4-я четверть»; идти на шаг 10).

5. Вывести «3-я четверть»; идти на шаг 10.

6. Проверить условие $A > 0$; если верно, то (вывести «2-я четверть»; идти на шаг 10).

7. Вывести «1-я четверть».

8. Идти на шаг 10 (или «стоп»).

9. Вывести «Треугольника нет!».

10. Закончить.

2—IX—Г Первый вариант решения. Решение получается простым и коротким, если провести некоторый анализ задачи. Пусть $a*100 + b*10 + c$ — искомое число, abc — его десятичная запись. Тогда abc , acb , bca , bac , cab , cba — десятичные записи всех чисел, получающихся перестановками цифр искомого. Сложив их все, получим $(2*100 + 2*10 + 2)*(a+b+c)$, что по условию равно $6*(a*100 + b*10 + c)$, откуда $7*a = 3*b + 4*c$ и $a = (3*b + 4*c)/7$, а поскольку b и c могут принимать значения 0, 1, ..., 9, то остается только проследить, чтобы a получилось натуральным числом (в частности, $a > 0$).

алг ЗАДАЧА (нат N)

рез N

нач нат A, B, C

для B от 0 до 9

нц для C от 0 до 9

нц A:=(3*B+4*C)/7

если ЦЕЛЧАСТЬ(A)=A и
A>0

то N:=100*A+10*B+C
все

кц

кц

кон

Второй вариант решения получается прямым перебором с проверкой. Цикл для нахождения всех перестановок из трех цифр заводить все же не имеет смысла.

алг ЗАДАЧА (нат N)

рез N

нач нат A, B, C, M1, M2, M3, M4, M5, M6, S
N:=0

для A от 1 до 9

нц для B от 0 до 9

нц для C от 0 до 9

нц M1:=100*A+10*B+C

M2:=100*A+10*C+B

M3:=100*B+10*C+A

M4:=100*B+10*A+C

M5:=100*C+10*B+A

M6:=100*C+10*A+B

S:=M1+M2+M3+M4+

+M5+M6

если S=6*M1

то N:=M1

все

кц

кц

кц

кон

3—IX—Г Первый вариант решения. Пусть X — смещение Робота на север, Y — смещение на восток, тогда перед началом цикла и после любого числа повторений цикла выполнены равенство $N+Y-K*X=S$ и неравенство $-K \leq S < N$. По окончании цикла для конечного положения Робота X, Y выполнено еще и равенство $X+Y=N+K$, поэтому либо $X=N$, $Y=K$, либо $X=N-1$, $Y=K+1$, либо $X=N+1$, $Y=K-1$ и т. д., но неравенство $-K \leq S < N$ может быть выполнено лишь при $X=N$, $Y=K$, в остальных случаях $|S| \geq N+K$. Таким образом, по окончании цикла Робот приходит в Траляляйск. Легко убедиться, что траектория Робота целиком лежит в туннеле. Расстояние до осевой линии туннеля равно $|S|/\sqrt{N^2+K^2}$, что всегда меньше единицы.

алг ЗИГЗАГИ (цел N,K)

арг N,K

рез

нач цел S,L

S:=0

для L от 1 до N+K

нц если S>=0

то ШАГ НА СЕВЕР

S:=S-K

иначе ШАГ НА ВОСТОК
S:=S+N

все

кц

кон

Второй вариант решения. Так как $N > K > 0$, то идти надо на север и восток. Заведем переменную X, в которой будет храниться расстояние от осевой линии дороги. Перед каждым шагом проверяется условие $X > 0$ (правее дороги); если да, то на север, нет — на восток. В этом случае за дорогу Робот не выйдет, так как он избирает меньший по модулю X (стремится к дороге), а модуль приращения X меньше 1, так как равен либо $-\cos(\text{ANG})$ (север), либо $\sin(\text{ANG})$ (восток), где $\text{ANG} = \text{arctg}(N/K)$. Мимо пункта назначения он не проскочит.

Функция path возвращает 0, если все хорошо, и 1, если N и K «плохие».

int path(int n,k)

{float ang,x;

if (n<=0) || (k<=0) return(1);

ang=arctan(n/k) ; /* alpha */

x=0.

do {

if x=0 {

n--;

north; x=x-cos(ang);

else{

k--; east;

x=x+sin(ang);

}

} while (n==0)&&(k==0);

return(0);

4—IX—Г. 1. Алгоритм заполнения клеток (один из возможных):

5 DIM A%(30,30)

10 SCREEN 3

15 FOR I=1 TO 30

17 A%(I,1)=100:A%(I,30)=100

18 A%(1,I)=100:A%(30,I)=100

19 NEXT I

20 FOR I=2 TO 29

30 FOR J=2 TO 29

35 A%(I,J)=B*RND(2)+1:GOSUB 1020

50 NEXT J:NEXT I

55 FOR J=2 TO 29

70 FOR I=2 TO 29

90 IF A%(I-1,J)-A%(I,J)=1 THEN

A%(I,J)=A%(I-1,J)

100 IF A%(I+1,J)-A%(I,J)=1 THEN

A%(I,J)=A%(I+1,J)

110 IF A%(I,J-1)-A%(I,J)=1 THEN

A%(I,J)=A%(I,J-1)

120 IF A%(I,J+1)-A%(I,J)=1 THEN

A%(I,J)=A%(I,J+1)

125 GOTO 1020

150 NEXT I:NEXT J

170 GOTO 55
 1020 PSET (I*4, J*4), A% (I, J)
 1060 RETURN

Здесь цифры от 0 до 7 трактуются как цвет точки на растре; точки изображаются клеточками-квадратами, так как используется графический режим низкого разрешения SCREEN 3 (язык MSX-BASIC).

2. Результат зависит от порядка перебора. Пример: последовательность 432 при просмотре слева направо превратится в 442, а справа налево — в 433.

3. На каждом шаге сумма чисел в клетках увеличивается ровно на 1; но она не может превышать $30 \times 30 \times 7$, следовательно, процесс конечен.

X класс

1—X—Г

ПРОЦЕДУРА ЗАМЕНА (C1, C2 : СТРОКА);
 ПЕРЕМЕННЫЕ СИМВОЛ S; ЦЕЛЫЕ I, J
 ПРОЦЕДУРА СБРОС; (*записывает в C2 повторитель и символ*)

88

НАЧАЛО

ЕСЛИ $J > 1$ TO $C2 := C2 + \text{Ъ}$ ($\text{Ъ} +$
 +СимвольноеПредставление (J) +')

$C2 := C2 + S$ (*плюс — это конкатенация*)

КОНЕЦ процедура сброс

НАЧАЛО $C2 := "$; $J := 0$; (*строки — переменной длины*)

ЕСЛИ $C1 = "$ TO

НАЧАЛО $S := C1(1)$

ДЛЯ $I := 1$ ДО $C1$

ЕСЛИ $S = C1(J)$

TO $J := J + 1$

ИНАЧЕ: НАЧАЛО СБРОС;

$J := 1$; $S := C(1)$ КОНЕЦ;

СБРОС

КОНЕЦ

КОНЕЦ;

При написании программы на Бейсике можно использовать функцию STR\$(N), возвращающую символьное представление числа N.

```

10 INPUT A$
20 L=LEN(A$)
30 I=1
40 K$=""
50 C$=MID$(A$,I,1)
60 J=1
70 IF C$<>MID$(A$,I+J,1) GOTO 110
80 IF I+J>L GOTO 110
90 J=J+1
100 GOTO 70
110 I=I+J
120 IF J=1 GOTO 140
130 K$=K$+"(" + STR$(J) + ")"
140 K$=K$+C$
150 IF I<=L GOTO 50
160 PRINT K$
  
```

2—X—Г Представим сведения об актрисах в виде прямоугольной таблицы натуральных чисел размером $N \times 5$, в которой номер строки — номер актрисы, а номер столбца — номер роли. Элементы таблицы — оценки возможности данной актрисы сыграть данную роль. Номера актрис упорядочены так, что к первой режиссер относится лучше всех, к следующей хуже и т. д.

алг РЕЖИССЕР (нат N, нат таб A [1:N,1:5], нат таб РОЛИ [1:5])

арг N, A

рез РОЛИ

нач нат Й1 , Й2 , РРОЛЬ, ВОЗМ

для Й1 от 1 до 5 ком : ПЕРВОНАЧАЛЬНОЕ ПРИСВАИВАНИЕ

нц РОЛИ [Й1] := 0.

кц

РРОЛЬ := 2 ком : ВЫБИРАЕМ ДЛЯ ЛЮБИМОЙ РОЛЬ,

ВОЗМ := A [Й1 , 2]

ком : КОТОРУЮ ОНА МОЖЕТ СЫГРАТЬ ЛУЧШЕ ВСЕГО.

для Й1 от 3 до 5

нц если A [Й1 , Й1] > ВОЗМ

то РРОЛЬ := Й1

ВОЗМ := A [Й1 , Й1]

все

кц

РОЛИ [РРОЛЬ] := 1

РРОЛЬ := 1 ком : ВЫБИРАЕМ ДЛЯ НЕЛЮБИМОЙ РОЛЬ, КОТОРУЮ

ВОЗМ := A [Н , 1] ком : ОНА МОЖЕТ СЫГРАТЬ ЛУЧШЕ ВСЕГО.

для Й1 от 2 до 5

нц если A [Н , Й1] > ВОЗМ и РОЛИ [Й1] = 0

то РРОЛЬ := Й1

ВОЗМ := A [Н , Й1]

все

кц

РОЛИ [РРОЛЬ] := Н

для Й1 от 1 до 5 ком : РАСПРЕДЕЛЯЕМ ОСТАЛЬНЫЕ РОЛИ.

нц если РОЛИ [Й1] = 0

то ВОЗМ := 0

для Й2 от 2 до $\text{Н} - 1$

нц если РОЛИ [1] <> Й2

и РОЛИ [2] <> Й2 и

РОЛИ [3] <> Й2

и РОЛИ [4] <

> Й2 и РОЛИ

[5] <> Й2

то если A [Й2 , Й1] >

> ВОЗМ

то РРОЛЬ := Й2

ВОЗМ := A

[Й2 , Й1]

все

все

кц

РОЛИ (Й1):=ПРОЛЬ

все

кон

3-Х-Г Один из возможных алгоритмов:

0. Ввести N вершин многоугольника.

1. Ввести массивы координат $X(1:N)$, $Y(1:N)$.

2. Обнулить массив $L(1:N,1:N)$.

3. $I:=0$.

4. $I:=I+1$.

5. $J:=1$.

6. $J:=J+1$.

7. Переход на подпрограмму ДИАГОНАЛЬ ($X(I)$, $Y(I)$, $X(J)$, $Y(J)$) (она выдает значение ключа $P=2$, если $X(I)$, $Y(I)$, $X(J)$, $Y(J)$ является диагональю, и $P=1$ — в противном случае).

8. Проверить условие $P=1$; если верно, то идти на шаг 6.

9. $L(I, J)=SQRT((X(I) - Y(I))^2 - (X(J) - Y(J))^2)$.

10. Проверить условие $J < N$; если верно, то идти на шаг 6.

11. Проверить условие $I < N - 1$; если верно, то идти на шаг 4.

12. Найти элемент LMAX в массиве L.

13. Вывести LMAX.

14. Вывести ($X(I)$, $Y(I)$) и ($X(J)$, $Y(J)$).
Подпрограмма ДИАГОНАЛЬ ($X1$, $Y1$, $X2$, $Y2$, P).

1. $P1:=0$, $P2:=0$.

2. $K:=0$.

2.1. Проверить условие $X1=X2$; если верно, то идти на шаг 5.

3. $C(X1, Y1, X2, Y2):=((Y2 \cdot X1 - X2 \cdot Y1) / (X1 - X2))$.

4. $A(X1, Y1, X2, Y2):=(Y1 - Y2) / (X1 - X2)$ (A и C — процедуры вычисления коэффициентов прямой $y=ax+c$, проходящей через точки ($X1$, $Y1$) и ($X2$, $Y2$)).

5. $K:=K+1$.

6. Проверить условие ($X(K)=X1$ и $Y(K)=Y1$) или ($X(K)=X2$ и $Y(K)=Y2$); если верно, то идти на шаг 5.

6.1. Проверить условие $X1 <> X2$; если верно, то идти на шаг 7.

6.2. Проверить условие $X(K) > X1$; если верно, то ($P1:=1$; идти на шаг 9).

6.3. $P2:=1$; идти на шаг 9.

7. Проверить условие $Y(K) > A \cdot X(K) + C$; если верно, то ($P1:=1$; идти на шаг 9).

8. $P2:=1$.

9. $P:=P1+P2$.

10. Проверить условие $P=2$; если верно, то идти на шаг 12.

11. Проверить условие $K < N$; если верно, то идти на шаг 5.

12. Возврат в основную программу.

4-Х-Г Поле, перекрытое детекторами, удобно описывать таблицей, строки кото-

рой соответствуют координате X, а столбцы — координате Y. Для упрощения расчетов будем считать, что первый детектор строки 1 имеет координаты $X=0$, $Y=0$. Кратчайшее расстояние между центрами детекторов без ограничения общности можно принять равным 1.

Программа сводится к процедуре поиска пар точек, через которые частица попадает в исследуемую область и покидает ее. Для каждой такой пары необходимо проверить, существует ли хоть одна точка на фотографии, которая опровергает гипотезу о прямолинейной траектории.

Процедура поиска пар точек должна обеспечить просмотр всех возможных пар граничных точек. Эта процедура не рассматривает только угловые точки, поскольку засветка лишь угловой точки всегда может интерпретироваться как прямолинейная траектория.

Проверка гипотезы о прямолинейной траектории в нетривиальных случаях проводится путем вывода уравнения соответствующей прямой и анализа только тех точек на фотографии, через которые должна проходить такая прямая.

Предложенная программа заканчивает работу сразу после обнаружения искомой траектории.

PROGRAM FOTO(INPUT, OUTPUT);

CONST N=5;

M=5;

(* структура для отображения фотографии *)

TYPE DETECT=ARRAY [1..N,1..M] OF 0..1;

SI=1..N;

SJ=1..M;

(* процедура проверки гипотезы о следе пролета частицы *)

FUNCTION F(VAR D:DETECT; ENTI; SI;

ENTJ; SJ; EXTI; SI; EXTJ; SJ):BOOLEAN;

VAR A,B,Y1,Y2,R:REAL

I,J,I1,I2,J1:INTEGER;

F1:BOOLEAN;

BEGIN

F1:=TRUE;

IF D[ENTI,ENTJ]*D[EXTI,EXTJ]=0

THEN F1:=FALSE

ELSE

IF ENTI=EXTI

THEN

FOR J:=2 TO M-1 DO

BEGIN

IF D[ENTI,J]=0 THEN F1:=

=FALSE

END

ELSE

BEGIN

Y1:=ENTJ-EXTJ;

Y2:=ENTI-EXTI;

```

A:=Y1/Y2;
B:=ENTJ-1-A*(ENTI-1);
IF ENTI<EXTI THEN
  BEGIN
    I1:=ENTI;
    I2:=EXTI
  END
  ELSE
    BEGIN
      I1:=EXTI;
      I2:=ENTI;
    END;
I:=I1;
REPEAT
  Y1:=A*(I-1.5)+B;
  Y2:=A*(I-0.5)+B;
  IF Y1>Y2 THEN
    BEGIN
      R:=Y1;
      Y1:=Y2;
      Y2:=R
    END;
  J:=ROUND(Y1+1);
  IF J<1 THEN J:=1;
  IF ABS((Y2+0.5-ROUND(Y2))/(
  (Y2+0.5))<=1E-6
    THEN J1:=ROUND(Y2)
    ELSE J1:=ROUND(Y2)+1;
  IF J1>M THEN J1:=M;
  REPEAT
  IF D[I,J]=0 THEN F1:=FALSE;
  J:=J+1;
  UNTIL (J>J1) OR NOT F1;
  I:=I+1;
  UNTIL (I>I2) OR NOT F1;
END;
F:=F1
END

```

(* основная процедура*)
 VAR FOUND:BOOLEAN; (* эта величина имеет значение TRUE, *)
 T:DETECT; (* если есть информация, которая может *)
 I,J,K,L:INTEGER (* интерпретироваться, как след *)
 BEGIN
 (* прием исходных данных *)
 FOR I:=1 TO N DO
 FOR J:=1 TO M DO READLN(T[I,J]);
 (* отображение двоичной фотографии *)
 FOR J:=M DOWNT0 1 DO
 BEGIN
 FOR I:=1 TO N DO WRITE(T[I,J]);
 WRITELN
 END;
 (* начало поиска *)
 IF (T[1,1]=1)OR(T[1,M]=1)OR(T[N,1]=1)OR(T[N,M]=1)
 THEN FOUND:=TRUE
 ELSE

```

(* проверка возможных пар точек *)
BEGIN
  I:=2;
  REPEAT
  J:=2;
  REPEAT
  FOUND:=F(T,I,1,1,J);
  IF NOT FOUND THEN
    BEGIN
      K:=2;
      REPEAT
      FOUND:=F(T,I,1,K,M) OR F(T,I,
      J, K, M)
      IF NOT FOUND THEN
        BEGIN
          L:=2;
          REPEAT
          FOUND:=F(T,I,1,N,L) OR
          F(T,I,J,N,L) OR F(T,K,M,N,L);
          L:=L+1;
          UNTIL (L>M-1) OR FOUND
          END;
          K:=K+1;
          UNTIL (K>N-1) OR FOUND
          END;
          J:=J+1;
          UNTIL (J>M-1) OR FOUND
          I:=I+1;
          UNTIL (I>N-1) OR FOUND
          END;
  IF FOUND THEN WRITELN('YES') ELSE
  WRITELN('NO')
  END.

```

Отборочный тур

1—0

1. Процедура МЕНЬШЕ
 PROC SMALL(X,Y,FLAG)
 NAT(X,Y)
 BOOL(FLAG,F1)
 EQU(X,Y,F1)
 IF(F1, MOVE(FLAG, FALSE), SMALLER(X,Y,0,0,FLAG))
 END
 PROC SMALLER(X,Y,Z,T,FLAG)
 NAT(X,Y,Z,T)
 BOOT(FLAG,F1,F2)
 EQU(X,Z,F1)
 EQU(Y,T,F2)
 IF(F1, MOVE(FLAG, TRUE), IF(F2, MOWE(FLAG, FALSE), SMALLER(X,Y,INC(Z),INC(T),FLAG)))
 2. Процедура ВЫЧИТАНИЕ
 PROC MINUS(X,Y,Z)
 NAT(X,Y,Z)
 BOOL(FLAG)
 SMAL(X,Y,FLAG)
 IF(FLAG,MOVE(Z,0),MINUS1(X, Y, 0, Z))

```

END
PROC MINUS1 (X,Y,T,Z)
  NAT (X,Y,Z,T)
  BOOL (FLAG)
  EQU (X,T,FLAG)
  IF (FLAG, MOVE (Z,T), MINUS1 (X,
  INC (Y),INC (T),Z))

```

3. Процедура УМНОЖЕНИЕ

```

PROC MULTA (X,Y,Z) -
  NAT (X,Y,Z)
  MULTA1 (X,Y,0,Z)
  END
PROC MULTA1 (X,Y,T,Z)
  NAT (X,Y,T,Z)
  BOOL (FLAG)
  EQU (X,0,FLAG)
  IF (FLAG, MOVE (Z,T), MULTA1 (MI-
  NUS (X,1,X),Y,PLUS (Y,T,T),Z))

```

4. Процедура СТЕПЕНЬ

```

PROC EXP (X,Y,Z)
  NAT (X,Y,Z)
  BOOL (FLAG)
  EQU (X,0,FLAG)
  IF (FLAG, MOVE (Z,0), EXP1 (X,Y,1,Z))
  END
PROC EXP1 (X,Y,T,Z)
  NAT (X,Y,T,Z)
  BOOL (FLAG)
  EQU (Y,0,FLAG)
  IF (FLAG, MOVE (Z,T), EXP1 (X, MI-
  NUS (Y,1,Y),MULTA (X,T,T),Z))

```

Здесь считается, что ноль в степени ноль равен нулю, хотя можно было бы предусмотреть еще одну переменную, принимающую в этом случае определенное значение, указывающее на некорректность данных.

5. Процедура ДЕЛЕНИЕ

```

PROC DIV (X,Y,Q,R)
  NAT (X,Y,Q,R)
  BOOL (FLAG,ERR)
  MOVE (ERR,TRUE)
  EQU (Y,0,FLAG)
  IF (FLAG, MOVE (ERR, FALSE), DIV1
  (X,Y,0,R))
  END
PROC DIV1 (X,Y,Q,R)
  NAT (X,Y,Q,R)
  BOOL (FLAG)
  SMAL (X,Y,FLAG)
  IF (FLAG, MOVE (R,X), DIV1 (MINUS
  (X,Y,X),Y,INC (Q),R))

```

2—0. Схема одного из возможных алгоритмов решения такова.

1. Найти точки с наибольшей и наименьшей абсциссами (если эти абсциссы совпадают, то имеем вырожденное решение).

2. Провести через них прямую, которая разбивает все точки на два множества: множество точек, лежащих выше прямой, и множество точек, лежащих ниже прямой или на ней.

3. Упорядочив по возрастанию абсцисс, а если нужно, и по возрастанию ординат «нижнее» множество, все его точки можно соединить непересекающейся замкнутой ломаной.

4. Прделав те же манипуляции с «верхним» множеством, мы получим решение задачи.

Разумеется, существует ряд тонкостей (некоторые исключения для некоторых точек); они не оговорены в общей схеме.

Более подробное изложение алгоритма.

1. Ввод количества точек N.
2. Ввод X [1:N], Y [1:N].
3. Поиск NMAX, NMIN — номеров точек с наибольшей и наименьшей абсциссами.
4. Если X [NMAX] = X [NMIN], то конец программы, вырожденный случай.

5. Находим коэффициенты A и B уравнения «разделяющей» прямой $Y = AX + B$:
 $A = (Y [NMAX] - Y [NMIN]) / (X [NMAX] - X [NMIN])$

6. Заполнение «верхнего» и «нижнего» массивов номерами соответствующих элементов. При этом NMAX и NMIN включаем в оба массива.

F:=1

M:=0

K:=0

для I от 1 до N

нц если Y [I] > A * X [I] + B

то M:=M+1; F:=0

MV [M]:=I

иначе K:=K+1

MN [K]:=I

если Y [I] < A * X [I] + B

то F:=0

все

все

кц

J:=J+2

MV [J-1]:=NMIN

MV [J]:=NMAX

Если F=0, то все точки лежат на «разделяющей» прямой и мы опять имеем вырожденный случай.

7. Вызов процедуры ПОРЯДОК для массива MN [1:K]; получим его обход слева направо.

8. Вызов процедуры ПОРЯДОК для массива MV [1:K]; получим его обход слева направо.

9. Перезаписать массив MV, «перевернув» его и предварительно удалив последний элемент, как совпадающий с последним элементом массива MN. Это необходимо для организации обхода справа налево.

10. Прописать к массиву MN массив MV. Результирующий массив будет искомым решением.

Процедура ПОРЯДОК

Упорядочение массива номеров точек по возрастанию абсцисс (а при совпадении абсцисс по возрастанию ординат) точек.

Исключение: те точки, абсциссы которых совпадают с наименьшей абсциссой, упорядочиваются по убыванию ординат.

$A[1 : N]$ — массив номеров

$X[1 : N]$ — абсциссы точек

$Y[1 : N]$ — ординаты точек

$F := 1$

пока $F = 1$

нц $F := 0$

для I от 1 до $L-1$

нц если $X[A[I]] > X[A[I+1]]$

то обмен $A[I]$ и $A[I+1]$

$F := 1$

все

если $X[A[I]] = X[A[I+1]]$

то если $Y[A[I]] > Y[A[I+1]]$

то обмен $A[I]$ и $A[I+1]$

$F := 1$

все

кц

кц

кц

$I := 1$

пока $X[A[I]] = X[A[I+1]]$

нц $I := I+1$

кц

для J от 1 до $[I/2]$

нц обмен $A[J]$ и $A[I-J+1]$

кц

Возврат в основную программу

3—0. Предлагая школьникам задачу на создание информационно-поисковой системы, авторы совершенно не рассчитывали получить ее полное решение; в отведенное время это было невозможно, да и не нужно. Авторы хотели проанализировать уровень информационной культуры участников отборочного тура, который проявился в подходах к решению поставленной задачи.

Новый термин — новое направление

В публикациях, посвященных электронно-вычислительной технике, стало часто появляться слово «транспьютер». Что это за устройство?

Транспьютер схож с микропроцессором тем, что содержит арифметико-логическое устройство, устройство управления и изготовляется в виде одной микросхемы. Отличие же в том, что у транспьютера на том же кремниевом кристалле имеется также ОЗУ и несколько (обычно четыре) каналов связи.

Благодаря этому транспьютер может, во-первых, работать как автономный компьютер, во-вторых, обмениваться информацией с другими транспьютерами. Такая комбинация свойств позволяет изготовлять из них ЭВМ нового типа — параллельной обработки данных. В общих чертах работа этих ЭВМ выглядит так: центральный процессор получает данные и программу их обработки, затем распределяет работу между транспьютерами (их бывает от нескольких десятков до нескольких тысяч). Каждый из них, в свою оче-

редь, сделав часть вычислений, передает результаты соседям (если им эти результаты нужны для работы), получает результаты от соседей и продолжает работу с новыми данными, передает результаты центральному процессору.

Фактически данные, циркулируя по такой ЭВМ, обрабатываются одновременно многими компьютерами, что позволяет резко уменьшить продолжительность сложных вычислений.

Без скальпеля и ножовки

В 1881 г. Александр Белл, используя лучший по тем временам индуктор, головные телефоны и сбалансированный мост, пытался обнаружить пулю в теле смертельно раненного при покушении президента США Гарфилда. Пулю

обнаружить не удалось. Пациент скончался. Это было первое практическое применение вихревых токов.

Сегодня вихревые токи являются основным инструментом при неразрушающем контроле изделий из ферромагнитных материалов. Однако недостаточная точность такого метода контроля однородности металлов несколько ограничивала область его применения. Позднее были разработаны математические методы локализации источников магнитного возмущения, но подобный анализ, хотя и давал хорошие результаты, не был достаточно оперативным. С появлением микропроцессоров эта задача стала решаться практически мгновенно, что и дало мощный импульс развитию метода неразрушающего контроля изделий.

Современный инструмент, работающий по-прежнему на вихревых токах, позволяет обнаружить и с высокой точностью определить размеры и форму не только поверхностных повреждений, скрытых тонким слоем краски, но и глубинных раковин, неоднородностей и швов. И если бы у Белла был такой прибор, то неизвестно, как бы сложилась судьба 20-го президента Америки.

ЧТО?
МОЖЕТ
ЭВМ

МОЛОДЕЖНАЯ ИНИЦИАТИВА

3

Локальная сеть для «Агата»

Отсутствие локальной сети — серьезный недостаток ПЭВМ «Агат». На загрузку программ для каждой ПЭВМ уходит слишком много времени — сначала надо запустить операционную систему, чаще всего ДОС 3.3, а уж потом программу пользователя.

Оригинальная локальная сеть разработана и испытана в Павлодарском педагогическом институте. Обмен осуществляется через магнитофонный разъем, имеющийся на задней панели ПЭВМ (схема изображена на рисунке). Передача по сети осуществляется с помощью подпрограмм работы с магнитофоном в мониторе и в интерпретаторе Бейсика. Информация как бы «записывается» на передающей ПЭВМ и «считывается» на принимающей. Обратите внимание, что в схеме перекрестное соединение проводов. Кабель используется стереофонический, 4-жильный, от бытовых магнитофонов. Такой кабель продается в специализированных магазинах. Длина кабеля, соединяющего отдельные ПЭВМ, должна быть достаточной, чтобы он провисал. В качестве штеккеров используются применяемые в бытовых магнитофонах СШ-3.

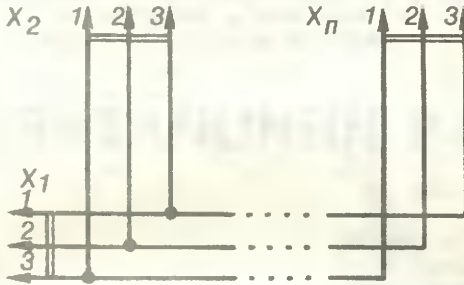
Данная сеть позволяет обеспечить передачу с «учи-

тельской» ПЭВМ на все «ученические». С «ученической» ПЭВМ передача возможна лишь на «учительскую». Скорость обмена 175 байт/с, или 1,37 кБод.

Как пользоваться такой сетью? В мониторе и в Бейсике есть команды для поддержания обмена с магнитофоном: в Бейсике — LOAD и SAVE (не путать с директивами ДОС 3.3 — LOAD (имя файла) и SAVE (имя файла), в мониторе — R и W. Обмен по сети осуществляется именно с помощью этих команд, т. е. для передачи не нужно никаких специальных управляющих программ (!). Для передачи Бейсик-программ необходимо ввести «ученические» ПЭВМ в режим ожидания, набрав LOAD; на «учительской» ПЭВМ требуемая программа загружается в память (например, с диска), затем дается команда передачи — SAVE. Время передачи всегда более 10 с, в случае ошибки обмена (из-за плохой пайки, помех от электроприборов, незаземленной металлической оплетки и т. д.) выдается сообщение ERR. Если передача прошла успешно, то программу можно запустить, набрав RUN (как на «ученической», так и на «учительской» ПЭВМ).

Передача программ в машинных кодах (В-файлов)

К "учительской"
ПЭВМ



осуществляется аналогично с помощью директив монитора R и W, но требует особых знаний и умений от «учителя» и от «ученика». Поэтому В-файлы лучше пересылать с помощью программы NET. Она работоспособна только в том виде, в каком приведена. Попытки внести в нее какие-либо изменения могут привести к непредсказуемым последствиям. NET следует набрать точно в соответствии с листингом, включая даже пробелы. Для контроля длины программы в Бейсике можно отдать команду PRINT PEEK (2739). Если будет напечатано 109, то в программе нет пропущенных или лишних символов. Во время набора нельзя использовать для проверки на наличие ошибок команду RUN. Дело в том, что NET видоизменяет сама себя. После того как программа набрана, проверена, ее записывают на диск и запускают уже с диска. О всех требуемых действиях NET сообщает сама.

Все вышеописанное можно свести в таблицу действий пользователя при передаче по локальной сети.

Эксплуатация сети выявила как ее достоинства, так и недостатки. К достоинствам относятся:

простота обращения;
низкая стоимость изготовления (порядка 7 рублей на рабочее место);

К "ученической"
ПЭВМ

простота изготовления;
отключенная ПЭВМ или автономная работа отдельного пользователя не влияют на обмен в целом;
возможность двустороннего обмена «учитель — ученик».

Сеть в перспективе позволит включить в работу машины с неработающими НГМД (сейчас разрабатывается передача по сети операционной системы ДОС 3.3 с интерпретатором Бейсика).

Среди недостатков — влияние на обмен по сети разброса электрических параметров магнитофонного канала отдельных ПЭВМ. Так, с одной из имеющихся в ППИ 13 ПЭВМ «Агат» сеть не работает.

При передаче В-файлов с помощью программы NET невозможна передача программ:

занимающих адреса \$801—\$17FF;
начинающихся с адреса

	Действия	
	учителя	ученика
LOAD (имя файла)		LOAD
SAVE		RUN
RUN		RUN
При передаче В-файлов		LOAD
RUN NET		RUN

```

10 TEXT= 2 : HOME : RIBBON= 2
20 IF PEEK ( $AB3 ) < > 109 THEN 320
30 VTAB 10 : HTAB 3
40 INPUT "ИМЯ ФАЙЛА ?" ; A$
50 PRINT CHR$ ( 4 ) ; "BLOAD" ; A$
60 HOME : VTAB 10 : HTAB 3
70 PRINT "НА ПРИНИМАЮЩИХ ПЭВМ"
80 PRINT:PRINT SPC(3);"НАБРАНО 'LOAD' ?";:GET R$
90 Q1 = PEEK ( $AA72 ) + PEEK ( $AA60 )
100 Q2 = PEEK ( $AA73 ) + PEEK ( $AA61 )
110 IF Q1 > 255 THEN 230
120 POKE $AB3 , Q1 : POKE $AB4 , Q2
130 POKE $AB5 , PEEK ( $AA72 ) + 1
140 POKE $AB6 , PEEK ( $AA73 )
150 GOSUB 240 : SAVE : GOSUB 270
160 HOME : VTAB 10 : HTAB 3
170 PRINT "НА ПРИНИМАЮЩИХ ПЭВМ"
180 PRINT:PRINT SPC(3);"НАБРАНО 'RUN' ?";:GET R$
190 GOSUB 240 : CALL - 304
200 HOME : VTAB 2 : HTAB 8
210 PRINT "ФЕДОСЕЕВ ППИ 1988"
220 END
230 Q1 = Q1 - 256 : Q2 = Q2 + 1 : GOTO 120
HOME : VTAB 14 : HTAB 7
PRINT "П О Д О Ж Д И Т Е"
RETURN
POKE $3C , PEEK ( $AB5 ) - 1
POKE $3D , PEEK ( $AB6 )
POKE $3E , PEEK ( $AB3 )
POKE $3F , PEEK ( $AB4 )
RETURN
GOSUB 240 : GOSUB 270 : CALL - 256
POKE $AB7 , $CE : POKE $AB8 , $B5
POKE $AB9 , $0A : POKE $ABA , $6C
POKE $AB8 , $B5 : POKE $ABC , $0A
CALL $AB7 : PRINT "МУРЛЫКАЛАХ"

```


§XYFF, где X и Y — шестнадцатиричные цифры (например, §10FF);

кончающихся адресом §XY0D, где X и Y — шестнадцатиричные цифры (например, §100D, §780D);

кончающихся адресом §XY6D, где X и Y — шестнадцатиричные цифры (например, §106D, §786D).

Пока обнаружены только три программы, передача

которых по сети невозможна, — это FID (работа с файлами), FACE (игра реверси, М. Левин), HONIX (игра ксоники, С. Ветютнев).

Перечисленные недостатки программы NET объясняются тем, что она написана на языке Бейсик. Вариант на ассемблере обещает быть лучше.

В. ФЕДОСЕЕВ

г. Павлодар, СШ № 3, IX класс

Быстрая рассылка программ в КУВТ «Ямаха MSX-1»

Нам часто приходилось наблюдать, как преподаватели рассылают программы на ученические машины. При этом затрачивается много времени, нередко из-за большой длительности передачи возникают сбои и операцию приходится повторять. Бывает и так, что пропадает первая строка программы. Программы же в машинных кодах, в том числе большинство игр, пересылке вообще не поддаются.

В нашем компьютерном клубе при Тульском государственном университете им. Л. Н. Толстого создана программа, значительно ускоряющая и облегчающая работу с сетью. Освоить ее очень легко.

Файлы на Бейсике и в машинных кодах, подлежащие рассылке, должны быть заранее специально оформлены (заданием последних трех символов имен) и записаны на диск.

Диск с программой вставляется в дисковод, и делается сброс (кнопкой reset или выключением - включением питания), после чего через некоторое время на экран выводится список файлов на диске.

Преподаватель должен указать курсором нужное имя и нажать клавишу про-

бела (для загрузки программы только на свою машину) или F1 (для рассылки программы по сети).

Дальнейшее происходит автоматически. На ученические машины пересылается авангардный модуль, который очищает экран и выводит сообщение «Loading...». Достаточно беглого взгляда на класс, чтобы узнать, на какие

Студенческий научно-производственный отряд математиков-программистов МИЭМ «ПРОМЕТЕЙ»

Студенческий отряд «Прометей» работает круглый год в Московском институте электронного машиностроения с ноября 1985 г. при участии студентов всех курсов. Бойцы отряда специализируются на разработке программного обеспечения учебного и производственного назначения для мини-ЭВМ и персональных компьютеров. Научное руководство деятельностью отряда осуществляет кафедра кибернетики.

С ноября 1985 по май 1987 г. СНПО «Прометей»

машины пересылается файл.

По окончании на экран ученической машины вновь выводится список файлов, и система снова готова к работе.

Чтобы организовать такой процесс, в программе было установлено одно из максимальных значений скорости передачи данных интерфейса RS-232C. Приемная и передающая программы написаны в машинных кодах, а обмен данными происходит напрямую через порты интерфейса, без использования подпрограмм ввода-вывода. Это позволило уменьшить время пересылки с 1—2 мин до 10 с.

Описанная программа используется преподавателями ТГПИ уже более трех месяцев. Нареканий не было. Получить ее можно по адресу: Тула, просп. Ленина, 125, ТГПИ, кор. 4, кафедра информатики.

Роман ТРУНОВ,

ученик IX класса СШ № 17,
г. Тула

участвовал в разработке программного обеспечения по заказу Института проблем информатики АН СССР. Мы подготовили более 20 учебных, деловых, инструментальных и игровых программ для лабораторного практикума по школьному курсу ОИВТ на базе «Ямахи». Лучшие программы переданы для опытной эксплуатации и рецензирования в ИПИ АН СССР, в педагогические и политехнические вузы страны. Бывший Минпрос СССР включил учебную программу «Азбука» (авторы —

О. Кудинова, И. Яшина) в состав стандартного пакета программ по информатике И-86. За разработку пакета учебных, деловых, инструментальных и игровых программ для персональных компьютеров СНПО «Прометей» награжден первой премией 1-го Московского городского конкурса молодых ученых и специалистов по учебным и игровым программам, Почетными дипломами МГК ВЛКСМ и МГС НТО. Программы, разработанные отрядом («Азбука», «Чертилка», «Станок» и др.), вошли в состав пакета учебных программ для компьютеров «Ямаха», признанного лауреатом II Всесоюзного конкурса на лучшую обучающую программу для ЭВМ.

С июля 1987 г. СНПО «Прометей» сотрудничает с Московским городским центром научно-технического творчества молодежи «Внедрение». По заказу центра бойцы отряда (М. Федосов, Д. Бирюков, С. Афонин и др.) выполнили ряд договоров на разработку алгоритмического и программного обеспечения автоматизированной системы учета поставок продукции и бухгалтерского учета для Останкинского экспериментального завода безалкогольных напитков. Было проведено исследование структуры информационных потоков в планово-финансовых службах предприятия, формализованы задачи учета поставок продукции и бухгалтерского учета, разработан методика оперативной обработки информации. Внедрение результатов работы позволило механизировать ввод исходных данных, обеспечить накопление и надежное хранение информации на магнитных носителях, автоматизировать формирование и печать выходных планово-финансовых документов.

СНПО «Прометей» активно

участвует в выполнении комплексных целевых программ Научно-исследовательской части МИЭМ. Бойцы отряда разрабатывают алгоритмическое и программное обеспечение гибких производственных систем. На летние каникулы бригада Ирины Щербининой выезжала в Ужгород для внедрения системы оперативного календарного планирования участка ГПС и автоматизированной транспортно-складской системы в производственном объединении «Тисса». Работа в этом направлении продолжается в течение учебного года.

Бойцы СНПО «Прометей» успешно сочетают учебную, научно-производственную и общественную работу. Несколько лет проводятся летние компьютерные школы в пионерских лагерях «Литвиново» и «Метеор». Школьники изучают основы информатики и программирования, получают практические навыки работы с персональными компьютерами. Обучением школьников руководит комиссар отряда Ольга Гучкова.

Бойцы отряда активно пропагандируют компьютерную грамотность, участвуют в работе научно-технических выставок и семинаров, демонстрируют свои разработки на различных общественно-политических мероприятиях.

Осенью прошлого года ЦК ВЛКСМ и БММТ «Спутник» организовали международную молодежную акцию «Москва — Брюссель-88». Отбор участников шел по конкурсной системе. Каждый претендент должен был представить свои предложения по подготовке и проведению акции.

СНПО «Прометей» для участия в конкурсе рекомендовал Д. Морозовского — одного из ветеранов отряда, специалиста по системному программированию для пер-

сональных компьютеров MSX. Он продемонстрировал учебное и системное программное обеспечение, рассказал о научно-производственной и общественно-политической деятельности отряда.

8 ноября 1988 г. советская молодежная делегация вылетела в Брюссель. В ее состав вошли инженеры и студенты, повара и архитекторы, молодые кинематографисты, представители творческой и научной интеллигенции. Программа предусматривала проведение научных и политических семинаров и дискуссий с представителями бельгийской молодежи, менеджерами и общественными деятелями. Участникам акции особо запомнились встречи с бургомистром Брюсселя и ведущими политическими деятелями страны в брюссельской ратуше и советском посольстве.

Для специалистов в области информатики наибольший интерес представляли участие в научном семинаре по современным проблемам науки и техники в Монсе и встречи с представителем Всебельгийской ассоциации по проблемам компьютерного обучения, студентами и преподавателями бельгийских университетов, представителями компьютерных клубов. С некоторыми из них удалось установить творческие контакты, обменяться программами и договориться о сотрудничестве.

В конце прошлого года мы заключили договор о совместной работе в области компьютеризации и обучения с Молодежным компьютерным общественным центром при Московском опытном заводе «Тизприбор». Условия договора предусматривают широкое сотрудничество центра, райотдела народного образования Свердловского района и СНПО «Прометей» при активном

участии Свердловского РК ВЛКСМ. В настоящее время на базе центра организовано обучение учащихся школ района основам информатики и программированию, работают компьютерные кружки и курсы компьютерной грамотности. Формируется пакет заказов на разработку программного обеспечения для КУВТ «Ямаха» и УКНЦ.

Перестройка высшего образования и активизация вузовской науки стимулируют дальнейшее развитие нашего отряда. В этом году мы работаем по четырем основным направлениям: школьная ин-

форматика, компьютеризация обучения в высшей школе, программное обеспечение ПЭС, автоматизация сборочных процессов в судостроении.

В целях повышения эффективности управления решено реорганизовать СНПО «Прометей» в объединение научно-производственных отрядов математиков-программистов. Каждый отряд имеет свое научное направление. Их деятельность строится на принципах полного хозрасчета и финансовой самостоятельности. Эксперимент продолжается.

Н. МЕШКОВ, В. РОКОТОВ

Кружок «Педагогическая информатика»: набор по конкурсу

Начал свою работу кружок педагогической информатики на математическом отделении. Позади конкурсный вступительный экзамен. Думаю, что всем студентам-математикам будет интересно узнать, что же за задачи решали первокурсники — вновь приобщенные адепты новой научной дисциплины.

Каждый вариант содержал две задачи. Первая — программа на неизвестном языке программирования (если честно, то специально придуманном для такого случая): надо было понять, что делает эта программа. Вторая задача — методическая: требовалось в неверно решенном школьником упражнении найти ошибки и для каждой из них построить профилактические задания. Важно заметить, что никаких методических знаний у первокурсников нет, поэтому задача проверяла скорее наличие у них педагогического чутья.

Вот пример варианта.

1.
100 READ TEXT A
200 MAKE S=0
300 REPEAT LEN (A)
400 ON PIECE (A, 1, 2)=
«F» DO S+1
500 RALL
600 ON S. GE. 10 DO OFF
«TO MANY»

700 OFF «NUMBER=»S
800 ALL

$$2. \frac{a-b}{a^2-b^2} + \sqrt{a^2-2ab+b^2} \times \frac{1}{a-b} = a+b + \frac{a-b}{a-b} = a+b+1=2,1+7,2+1=10,3.$$

В первой задаче многие сообразили, что программа подсчитывает количество букв «F» в заданном тексте, и если оно не меньше 10, то выдает сообщение «TOO MANY», в противном случае выдается это количество. Вторая задача содержала подводный камень. Ошибку $\frac{a-b}{a^2-b^2} = a+b$ заметили все.

К сожалению, никто не увидел ошибки $\sqrt{a^2-2ab+b^2} = a-b$, хотя из решения видно, что $a=2,1$; $b=7,2$, т. е. $a < b$. Правильно будет: $\sqrt{a^2-2ab+b^2} = |a-b| = b-a$ (тогда и ответ 8,3).

Из 21 желающего в кружок приняты 6 человек. Что их ждет? Очень серьезная работа и, смею надеяться, хорошие результаты. Подробнее можно будет узнать из бюллетеня, который мы намерены иногда выпускать.

**В. ГУЗЕЕВ,
МОПИ**

КЛУБ ПОЛЬЗОВАТЕЛЕЙ БК

Как записать картинку

Проверить, хорошо ли БК-0010 стыкуется с магнитофоном, совсем нетрудно. Включите машину и переведите ее в режим работы с тестами (ЗАГЛ П Т). Признаком того, что БК готов к работе, будет знак + на экране. Теперь можно выйти в режим «Директивы отладки» (РУС Т С). На экране появится знак денежной единицы — буква X с кружочком посередине.

Можно начинать проверку. В графическом режиме чертим рисунок из прямых линий. Когда он будет готов, запишем его в ОЗУ, начиная с 1000 адреса (40000 А 100000 Д 1000 П). Из ОЗУ данные скопируем на магнитную ленту (МЗ АДРЕС-1000 ДЛИНА- 40000 ИМЯ-ПРОБА). Перемотав ленту к началу и очистив экран, считываем только что созданный

файл в экранную память компьютера (МЧ АДРЕС- 40000 ДЛИНА- 40000 ИМЯ- ПРОБА). Пока идут запись и считывание данных, как раз и должны проявиться всевозможные ошибки, связанные с плохим качеством подключения кассеты или магнитофона. По мере загрузки экран, строка за строкой, начнет заполняться исходным изображением. Любая ошибка немед-

ленно проявится на экране как искажение картинки, а по их количеству можно будет судить и о качестве сопряжения.

Настроив магнитофон, вы сможете пользоваться этим способом для записи на ленту любых картинок, созданных на компьютере или самим компьютером.

А. ПОГАЕВСКИЙ
(пос. Провидения)

Программная орбита

Этот выпуск программной орбиты для БК мы составили по материалам «Каталога-описания программ», изданного кооперативами «ЭВМ» (Ленинград) и «СБИС» (Сыктывкар).

26. SPRINT (Г. Прис, Москва). Машинные коды. Новая подпрограмма Минифонда SPRINT — это многофункциональный драйвер печати. Программа проверена и отлажена с печатающим устройством ROBOTRON, CM 6329.01/К 6311 М. Программа управляет всеми функциями принтера: сменной шрифтов, установкой интервалов, табуляцией и т. п., печатает тексты, содержащие латинские и русские заглавные и строчные буквы, цифры и допустимые для БК-0010 символы, включая символы псевдографики. Печатает графическое изображение экрана или любой его части, программы в восьмеричном коде и программы на Фокале, печатает переменные Фокала в десятичной или восьмеричной системах счисления. Подпрограмму SPRINT можно включать в программы, написанные на Фокале.

27. ПРОТЗ.4, ПРОТЗ.4 (В. Прохоров). Машинные коды. Монитор-отладчик для

создания программ в кодах процессора K1801BM1. Удобен также для изучения принципов работы процессора и системы команд, принятой в компьютерах типа БК-0010. Позволяет выводить информацию на любой принтер, причем, запуская программу из отладчика, можно переадресовать все обращения к дисплею на принтер. Всего 4К байт.

28. АФИША (В. Прохоров). Машинные коды. Набор описаний для системы ПРОТЗ.4.

29. DIPROT MONSTR 4.AFP (В. Прохоров). Машинные коды. Позволяет распечатать любые тексты, созданные с помощью редактора, EDASP или тексты программ на ассемблере МИКРО на любом принтере, подключенном к БК к параллельному или последовательному интерфейсу. Ни редактор, ни ассемблер не требуются. С помощью этой программы можно создавать автономно работающие инструкции, письма и прочие тексты. Обеспечивает автоматическую загрузку, компоновку, запись и страничный просмотр.

30. RPC3, USU, ORC (С. Камнев, Москва). Машинные коды. Программа RPC3 и набор модулей позволяют составлять, редак-

тировать и отлаживать программы в кодах, записывать многостраничные тексты непосредственно с экрана, создавать картинки для компьютерной мультипликации и мелодии.

31. Prez (Президент), Фокал. Одна из множества программ для БК, призванных имитировать деловые игры и развивать логическое мышление. Вас выбрали президентом острова. Останетесь ли вы в власти на следующий срок — зависит от вашего благосостояния. Оно складывается из доходов от сельскохозяйственного производства, от индустрии туризма и от сдачи в аренду земли иностранным компаниям. Появление промышленности вызывает приток туристов, разумеется, до определенных размеров, поскольку неудержимое развитие промышленности нарушает экологическую обстановку, снижается и объем сельскохозяйственной продукции. На борьбу с загрязнением можно выделить особые средства. Рост урожайности находится в зависимости от расходов на образование.

32. Wolk (Красная Шапочка), Фокал. Сюжет по мотивам «Красной Шапочки» Ш. Перро. Красная Шапочка с вашей помощью в очередной раз направляется к своей бабушке. На этот раз кроме пирожков она решила порадовать старушку цветами. Волк, как обычно, охотится на Красную Шапочку, но, как это ни странно, бегаёт медленнее ее. Единственный шанс для волка — воспользоваться тем, что Красная Шапочка останавливается, чтобы сорвать цветок. Попробуйте помочь Красной Шапочке собрать как можно больше цветов, избежав встречи с волком.

33. ПСП (Пакет стандартных программ), коды. Этот пакет разработан для всех поклонников Ассемблера из

восьми модулей — он позволяет избавиться от рутинной работы. Каждый модуль является перемещаемым в памяти. Модули: упаковывает/распаковывает восьмеричные числа с обычной точностью без знака (1), умножает/делит целые числа с обычной точностью со знаком (2), упаковывает целое десятичное число с удвоенной точностью со знаком, распаковывает целое число с обычной/двойной точностью в десятичное целое число (3), умножает/делит целые числа с удвоенной точностью без знака (4), упаковывает десятичное число в нормализованное с плавающей запятой, распаковывает

нормализованные числа в десятичные с удвоенной точностью со знаком, распаковывает нормализованные числа с плавающей запятой в десятичное в Е-формат (5), вычисляет сумму/разность/произведение/частное, изменяет знак нормализованного числа с плавающей запятой или нормализует число с плавающей запятой (6), пересылает числа с плавающей запятой, сравнивает их, преобразует в целые с обычной и удвоенной точностью, преобразует целые числа с обычной/удвоенной точностью в нормализованные с плавающей запятой (7), а также вычисляет функции sin, cos, atan, ln, exp и sort (8).

жать с любого свободного адресу ОЗУ, если предварительно зарезервировать память оператором CLEAR. Новая программа несколько модернизирована. Так, все данные о мелодии (частота и длительность нажатия очередной клавиши) записываются в ОЗУ. Это означает, что в любой момент можно отдать БК команду повторить фрагмент мелодии. После того как работа с программой завершена, данные легко записать на магнитофон, а с помощью вспомогательной программы преобразовать их в изображенные партитуры.

Управляют программой тремя специальными клавишами:

С — очистка памяти и подготовка к записи нового фрагмента;

П — повтор записанного фрагмента;

Д — удаление данных о последнем записанном звуке (клавиша);

Ф — выход из программы в монитор (или Бейсик). Выход в Бейсик осуществляется тогда, когда программу «ЭМИ» запускают из Бейсика с помощью функции USR.

Программа на рисунке приведена в машинных кодах начиная с адреса 1000. Восьмеричная константа 200, записанная по адресу 1040,

Электромузыкальный инструмент с памятью

Одной из первых программ, превращавших БК-0010 в электромузыкальный инструмент, была программа П. Коноплева (Наука и жизнь, 1987, № 4.). К сожалению, для БК-0010.01 эта программа едва ли годилась — расположение клавиш в машинах различно. Был и еще один дефект. Вычисленный по формуле звукоряд на практике не соответствовал

истине — в программе не были учтены задержки, связанные с опросом клавиатуры.

Новая программа написана для БК-0010.01. Звучат клавиши от «;» до «/» (второй ряд) и от «ТАБ» до «ВС» (третий ряд), показанные на рисунке.

Программа написана в позиционно-независимом коде, то есть ее можно загрузить

Адрес	0	2	4	6	0	2	4	6
1000	12737	100	177660	10700	62700	334	10067	416
1020	10700	62700	424	10067	406	10067	404	62767
1040	200	376	16767	370	372	32737	100	177716
1060	1374	13700	177662	16701	344	122100	1034	5046
1100	111116	12601	62701	25	5000	4767	174	5200
1120	32737	100	177716	1771	26767	306	306	101745
1140	10177	300	62767	2	272	10077	266	62767
1160	2	260	732	105721	1340	122100	1723	122100
1200	1014	26767	236	230	3715	162767	4	224
1220	32737	100	177716	1774	710	122100	1004	42737
1240	100	177660	207	122100	1017	16700	162	26700
1260	162	101673	12001	5201	12002	4767	14	77203
1300	12702	40000	77201	763	660	10103	12737	100
1320	177716	62703	5	77301	10103	12737	0	177716
1340	77301	207	177411	167473	160112	150461	142103	133525
1360	125463	120113	112464	105505	100465	74116	70107	64067
1400	60133	54470	51135	46132	43069	40110	35455	33072
1420	30457	26137	24023	0	42123	50106	20040	

резервирует 128 байт — это достаточно, чтобы хранить данные о 32 клавишах. Если вы захотите увеличить размер памяти для данных, достаточно увеличить константу по адресу 1040.

Чтобы ввести программу, напечатанную в журнале, рекомендуем воспользоваться Бейсиком. Текст программы записывают в операторы DATA, при этом к каждому коду текста машинной программы не забудьте добавить признак восьмеричной константы «&», например: 1000 DATA & 0012737 & 0000100, & 0177660, & 0010700 и т. д.

Основная программа, которая загружает машинную программу из операторов DATA, может выглядеть, например, так:

```
100 A%=& 034000 начальный
адрес загрузки
110 FOR 1%-A% TO A% +
+&0436 STEP 2
120 READ N%
130 POKE 1%, N%
140 NEXT 1%
150 DEFUSR-A%
160 N%-USR (0) запуск про-
граммы
```

И. ЗАХРЕБЕТКОВ,
г. Владимир

лем со считыванием программ с магнитофона.

И. Горячук,
г. Балаково

Известно немало программ для БК, обрабатывающих прерывания от клавиатуры. Если речь идет о Фокале, то все программы обладают общим недостатком: в тех случаях, когда необходимо различать несколько клавиш, конструкция из набора последовательных операторов IF очень замедляет работу программы. В то же время на Фокале можно реализовать структуру, напоминающую оператор ONGOTO Бейсика. Она основана на том, что аргументом операторов GO и DO может быть не только номер строки, но и имя переменной (но не арифметическое выражение).

И. Лисицин,
Ленинград

Чего нет в описании

Работая на БК-0010 с Бейсиком 00001-01.35.02, я перенумеровывал строки в длинной программе и по ошибке набрал оператор RENUM не полностью. Однако БК выполнил приказание. Тогда я решил проверить и остальные служебные слова, и вот что получилось (в скобках — необязательная часть слова).

Набранные таким образом операторы при выводе по команде LIST читаются полностью.

Надеюсь, это поможет владельцам БК, набирающим большие программы.

```
COL(OR)
REN(UME)
RES(TORE)
DA(TA)
FO(R)
IN(PUT)
CI(RCLE)
PS(ET)
LIN(E)
MO(NIT)
ST(OP)
OP(EN)
SA(VE)
FI(ND)
```

```
DEL(ETE)
RET(URN)
GOS(UB)
REA(D)
NEX(T)
DI(M)
PA(INT)
DR(AW)
PRES(ET)
BE(EP)
EN(D)
CLOS(E)
LOA(D)
CLE(AR)
```

Алеша КОЗЛОВ
VIII класс ЦШ № 9, г. Пермь

```
1.10 SG (8)= 10,25; SG (25)=
= 10,3; SG (76)= 10,35;
SG (27)= 10,4
10.10 SA=(FX(1,177662)); X
FK (28,12) DC A
10.15 X FK(28,12); T« »;
G 10.10
10.20 R; С нажата клавиша,
код которой не обра-
батывается
10.25 T «влево»; С клавиша с
кодом 8
10.30 T «вправо»; С клавиша
с кодом 25
10.35 T «вверх»; С клавиша с
кодом 26
10.40 T «вниз»; С клавиша с
кодом 27
```

Маленькие хитрости

Выбирая магнитофон для работы с БК, советую придерживаться принципа «чем ниже класс, тем лучше». Прежде всего, частотный диапазон таких магнитофонов существенно уже, на качество записи сигналов от БК это практически не влияет, а вот действие высокочастотной составляющей помехи

существенно уменьшается. Кроме того, магнитофон 4-го класса стоит намного дешевле, чем 2-го или даже 3-го. Год назад я приобрел «Протон-402». Вся доработка заключалась в том, что из схемы АРУ я удалил транзистор V12 (KT3156), чтобы разорвать цепь АРУ. С тех пор у меня нет никаких проб-

Счастливые обладатели принтера CRA-80 («Compute mate-100»). Если у вас полностью израсходован ресурс красящей ленты — не отчаивайтесь. К вашему аппарату подойдет лента для расширенного принтера EPSONLQ 800. Кассета, в которую уложена эта лента, больше по размеру, чем кассета SP-80, но ее можно вставить в принтер так, что ведущий вал попадает точно

в соответствующее отверстие кассеты. При этом левый край кассеты будет упираться в левую стенку принтера, а правый край окажется между металлическими фиксаторами и креплением двигателя.

А. Чипов,
Москва

Нередко, работая с Бейсик-программой, можно «выскочить» в пусковой монитор. Если попытаться вернуться в Бейсик командой G, то программа будет безвозвратно потеряна. В то же время, ее можно сохранить, если в ответ на приглашение монитора «?» отдать команду? G 123222 (Ввод). После появления знакомого «OK» нажмите клавишу СТОП. Программа будет спасена, но перестанут работать ключи «AR2 + (0...9)». Этот способ годится для всех БК, у которых Бейсик хранится в ПЗУ.

Е. Романов,
г. Воронеж

Транслятор Бейсика-БК различает переменные по двум первым символам. Тем не менее переменным можно присваивать одно и то же имя — Бейсик не спутает их, если переменные будут различаться типом, например S\$, S %, S!, S#, S\$(1), S %(1), S! (1), S# (1).

Д. Усенков,
Москва

Как известно, Бейсик-Вильнюс, зашитый в ПЗУ БК-0010.01, требует для своей работы много памяти, поэтому большие программы не умещаются в ОЗУ. Из этой затруднительной ситуации можно выйти, если есть второй компьютер (для этого подойдет БК-0010). Я соединяю компьютеры многопроводным кабелем и запускаю программу, имитирующую принтер на БК-0010. На БК-0010.01 даю команды AP2/СБР AP2/СБР и CLEAR,

& 070000 и запускаю программу на Бейсике. Предварительно необходимо все операторы PPINT заменить на LPRINT. Есть только одно ограничение — способ хо-

рош лишь для тех программ, которые не требуют ввода с клавиатуры.

А. Иванов,
г. Свердловск

Доска объявлений

Предлагаю обмен программами и опытом эксплуатации БК. Интересуюсь различными доработками БК. Адрес: 413800, Саратов обл., г. Балаково, ул. 30— лет Победы, д. 5, кв. 48. **И. Н. Гарячук.**

Мукачевский клуб пользователей персональных компьютеров предлагает обмениваться программным обеспечением, а также идеями по его созданию, как с клубами, так и с отдельными пользователями. Адрес: 295400, Закарпатская обл., г. Мукачево, ул. Ив. Франко, д. 148/62. **А. А. Семенов.**

Веду в школе кружок программирования. В нашем распоряжении КУВТ-86 с ДВК-2, в сеть подключен кассетный магнитофон, так что есть возможность вводить информацию с любого носителя. Хотелось бы перейти с Бейсика на Форт, но где его взять? Адрес: 620116, Сверд-

ловск, ул. Спортивная, д. 4, кв. 9. **Л. Н. Мезенцев.**

Располагаю библиотекой игровых, системных и обучающих программ для БК, рад буду обменяться со всеми желающими. Адрес: 380044, Тбилиси, ул. Гарекхетинская, д. 2, кв. 10. **А. А. Вадачкария.**

Администрация межшкольного учебно-производственного комбината, на базе которого создан районный центр информатики, просит помощи в приобретении программных средств для КУВТ-86. Адрес: 659305, г. Бийск, ул. Васильева, д. 4, МУПК 2. **Е. А. Макушин.**

Обменяюсь программами с владельцами БК. Интересуюсь расширениями ОЗУ, способами подключения джойстика и мыши. Адрес: 222035, Минск, ул. Тарханова, д. 8, кв. 22. **С. В. Асташенков.**

Привет, ребята!

Это я, маленький Таро Тасин из Страны восходящего солнца, приветствую вас в большом мире информатики. По воле и стараниями издательства «Мир» мне выпала честь поздравить вас с Международным днем защиты детей и предложить вам в качестве подарка путешествовать по удивительному,

простому и сложному, объёмному миру информатики. Подарков было не много — всего лишь 125 тыс., и очереди в Московском Доме книги в день, когда продавали нашу книжку, не было только благодаря хорошей работе продавцов, которые едва успевали распаковывать пачки с книгами. Поэтому я



и хочу, хотя бы вкратце, наметить путь, которым вы могли бы пройти, используя наш путеводитель.

Конечно, у всех вас есть вопросы. Как я об этом узнал? О, это было очень просто — Хасегава-сан (автор книги) с помощью глобальной информационной системы собрал все вопросы и передал мне. Итак, вперед! Сначала оглянемся вокруг, посмотрим, где могут использоваться компьютеры, и в чем их преимущества. А уж потом я расскажу, как компьютер устроен и с какими данными он может работать. Для очень любознательных можно углубиться в такие тонкости устройства компьютера, что ни в сказке сказать, ни пером описать, ни под микроскопом увидеть. А раз так, то мы представим все на привычных нам примерах и посмотрим, как все это работает с точки зрения отдельных частей компьютера.

Вот говорят: программное обеспечение или программа, что это такое? Я сразу скажу, что без программы компьютер не работает. И тут мы попадаем в такие дебри всяких специальных терминов, что я убедительно про-

шу вас не отставать. Я расскажу, точнее — покажу, вам, как программы создаются и как они работают, что такое языки программирования и для чего они нужны. Только у нас вы можете увидеть работу самой ГЛАВНОЙ ПРОГРАММЫ — ОС (операционная система). Я также покажу вам, какие еще программы входят в программное обеспечение компьютера.

Только после того как мы посмотрели, как устроен компьютер, можно представить, что происходит с данными. Например, почему так быстро можно получить деньги по кредитной карточке или как быть в случае сбойной ситуации. На примере работы банка и поликлиники мы можем рассмотреть любой случай из жизни компьютерной системы. Не менее интересно проследить за процессом разработки компьютерной системы. Все как при создании кинофильма «Звездные войны».

Но данные нужно каким-то образом передать компьютеру для обработки и еще важной получить от него в виде результатов. Для начала посмотрим, как компьютер печатает. В нашей

коллекции почти все виды принтеров и дисплеев. А после этого можно посмотреть устройства, способные читать и распознавать буквы, написанные от руки. Некоторые устройства, подключенные к компьютеру, могут даже говорить, они называются устройствами речевого вывода. Не менее удивительны устройства, которые умеют слушать. Мы можем показать еще много устройств ввода-вывода, но надо переходить к другой теме, точнее — в другую книгу.

Если нам потребуется передать информацию, например, в издательство «Мир», не обойтись без сетей, не рыболовных, конечно, а компьютерных. Ведь должны компьютеры обмениваться своей и нашей информацией. Как это можно сделать по телефонной сети и кто в этом может помочь? Что такое частные каналы и спутниковые каналы цифровой связи? Каковы их возможности? Какие есть способы управления тем громадным объемом информации, которая летает, вращается, путешествует по сетям? Все это можно посмотреть у нас. Можно даже заглянуть в будущее.

Но у нас еще остались вопросы. Как, например, эти сети работают? Зачем нужна передача данных? У нас вы можете «изучить анатомию передачи данных», посмотреть как ведут себя «воспитанные» компьютеры, даже заглянуть в историю взаимоотношений компьютеров. Можете узнать, что привело к созданию единого «дипломатического» протокола, и как «переехать в другую страну», т. е. преобразовать протокол передачи данных.

А самые непонятные слова в последнее время — это автоматизация учреждений, локальные сети, безбумажная информатика. Вот это и надо хорошенько изучить. Какие функции выполняются

в учреждениях и в чем может помочь компьютер? Информационные системы, электронные секретари привели к полностью безбумажному производству. Посмотрите, как проводится электронное совещание. Даже такие модные слова, как «рабочая станция» и «полиэкранный отобразитель», для нас не новинка. Для их эффективного использования пришлось создавать «локальные сети», правда, в нашей коллекции только два их типа. Постепенно мы соединяем с компьютерами все, что можно.

И вот результат — всякая большая система часто ломается, а уж если проберется злоумышленник, то все может кончиться очень плохо. Поэтому мы покажем, отчего и как нужно защищать ком-

пьютер и информацию, а также каким образом это делается.

Я с вами прощаюсь в надежде, что Вы захотите осуществить это путешествие сами, взяв в руки книгу. Но прежде два слова о ее создателях.

Хисахико Хасегава более 15 лет был связан с разработками систем передач данных. Устав от монополии сухих и специальных слов, он решил создать книгу с рисунками, поясняющими сложные на первый взгляд технические решения. Однако для реализации этой идеи нужно было, во-первых, найти издательство, которое ее одобрит, во-вторых, отобрать большое количество конкретных примеров, облегчающих понимание столь сложной техники, в-третьих,

найти опытного художника-иллюстратора, который бы понимал эту технику. К счастью, японское издательство «Омся» любезно предложило свои услуги, а советское издательство «Мир» приняло эстафету. Превосходным иллюстратором стал М.Ваки, а немногочисленные подписи к рисункам перевел Ю. Чернышев.

Издательству «Мир», по-видимому, понравилось выпускать хорошие книги, и будем надеяться, что эта серия будет продолжена.

С самыми добрыми пожеланиями ваш

Таро ТАСИН.

(«перевел» с японского
А. СЕВЕРНЫЙ)

Хасегава Х. Мир компьютеров в вопросах и ответах: В 2 кн. М.: Мир, 1988.

ЧТО МОЖЕТ ЭВМ?

Библиотека в кармане

Любители кроссвордов используют самые разные словари: толковые, специализированные по областям наук, частотные, орфографические, словари иностранных слов, энциклопедические, атласы. Все эти справочники — солидные, с широким корешком, иногда в нескольких томах, с трудом уместяющиеся на книжной полке. Поэтому пользоваться ими во время прогулки по парку, на солнечном пляже или в городском транспорте, не всегда удобно.

Приобретя карманный «процессор слов», разработанный одной из американских фирм, страстные кроссвордисты не только смогут всегда иметь при себе емкий словарь, вмещающий тысячи слов, но и получат в свое распоряжение хорошего помощника, который в считанные секунды найдет нужное слово по нескольким произвольным буквам, запомнит новые слова, внесет изменения в свой архив. В специализированном

компьютере размером всего десять на восемь и толщиной два сантиметра спрятано 4,5 мегабайта памяти для хранения слов и два специализированных микропроцессора для поиска и обработки текстовой информации.

Теперь разработчикам «процессора слов» стоит, наверное, подумать над созданием карманного «генератора кроссвордов» для обеспечения своего детища работой.

Умный компас

Если вы плохо ориентируетесь на местности, или решили исследовать запутанный лабиринт карстовых пещер, или просто любите собирать грибы — вашим проводником может стать «умный» компас.

В верхней части электронного прибора, уместающегося на ладони, расположен цифровой жидкокристаллический дисплей, на который, в зависимости от установленного режима, выводится либо азимут вашего движения в градусах, либо время, в течение которого вы идете в выбранном направлении.

Но самым замечательным качеством электронного компаса является то, что он детально запоминает маршрут движений и может точно воспроизвести его в любой момент. Поэтому, если вы потеряли надежду найти и грибы и обратную дорогу, обратитесь к компасу, и он напомнит вам весь ваш маршрут, который хранится в его памяти в виде последовательности пар чисел: азимут и время движения в эту сторону. Пределав указанный путь в обратном направлении, вы вернетесь в исходную точку. И пусть это будет не самая короткая дорога к вашему дому, зато она пройдет по уже знакомым вам местам.

НАУЧНО-МЕТОДИЧЕСКОЕ ОБЪЕДИНЕНИЕ НЕПРЕРЫВНОЕ ОБРАЗОВАНИЕ

предлагает

- обучающую программу для пользователей программно-методического комплекса «Автограф-845»;
- обучающие программы по освоению русской и латинской клавиатур ЭВМ: ДВК, СМ-1420, СМ-1810, Роботрон 1715, ЕС-1840, Искра-1030, IBM PC в операционных системах RT-11, RSX-11, CP/M, MS-DOS, БЭС 1810. Программы разработаны на основе методики интенсивного обучения и позволяет за несколько часов овладеть слепым десятипальцевым методом работы;
- информационную систему для отечественных и зарубежных ЭВМ (совместимых с IBM PC), облегчающую труд руководителя и секретаря, связанный с ведением картотеки, подготовки текстовых документов, планированием рабочего времени, ведением расчетов.

**Пишите и заходите:
107140, Москва, ул. Русаковская, 12.
Звоните по телефонам:
386-69-33, 946-38-19.**

ВНИИ профессионально-технического образования

СОЗДАН

отраслевой фонд алгоритмов и программ (ОФАП)
профессионально-технического образования.

Его основной задачей

является накопление и распространение
программных средств учебного назначения:

- педагогических программных средств (ППС) по различным предметам профессионально-технического цикла обучения ●
- ППС для внеурочной работы и технического творчества учащихся ●
- инструментальных средств для разработки ППС ●
- системных программ для отечественных микро-ЭВМ всех типов ●
- программные средства автоматизации управленческой деятельности ●

Для ОФАП создан экспертный совет, который осуществляет экспертизу поступающих программных средств. Программные средства, прошедшие экспертизу, поступают в ОФАП для тиражирования.

**ОФАП покупает и продает
педагогические программные средства
учебного назначения.**

**191119, Ленинград,
ул. Черняховского, 2, тел. 112-24-75**

Р. ДИМЕНШТЕЙН, А. ЯКОВЛЕВ
Институт программных систем АН СССР

Информатика или компьютерное дело?

Цзы-лу спросил Конфуция: «Вэйский правитель намеревается привлечь вас к управлению государством. Что вы сделаете прежде всего?» Учитель ответил: «Необходимо начать с исправления имен». Цзы-лу спросил: «Вы начинаете издали. Зачем нужно исправлять имена?» Учитель сказал: «Как ты необразован, Ю! Благородный муж проявляет осторожность по отношению к тому, чего не знает. Если имена неправильны, то слова не имеют под собой оснований. Если слова не имеют под собой оснований, то дела не могут осуществляться».

(Из китайского трактата
V в. до н. э. «Лунь юй»)

В последние годы в англо-американской специальной и популярной литературе получил широкое распространение термин *computer science*. Согласно словарю [1], это «общее название для группы дисциплин, занимающихся различными аспектами применения и разработки ЭВМ», и с таким пониманием, видимо, не сможет не согласиться подавляющее большинство специалистов. Очевидные практические потребности привели к необходимости адекватного перевода этого термина на русский язык. Буквальный перевод — *компьютерная наука* — не устраивал почти никого, поскольку получалось, что это наука о компьютерах. Не нашли признания и другие варианты, основанные на попытках непосредственного перевода слова *computer*, означающего, как известно, *вычислитель*. Такие сочетания, как *наука о вычислениях*, *вычислительная наука*, прижиться не смогли, поскольку отражали не различные, а лишь один: аспект разработки и применения ЭВМ (впрочем, название «Вычислительные науки» получил издающийся с

1988 г. реферативный журнал ВИНТИ).

105

Трудности с дословным переводом привели к вынужденному употреблению термина-заместителя, призванного «исполнить обязанности» точного эквивалента. Речь идет о слове *информатика*, этимологически оторванном и от компьютеров, и от вычислений. С самого начала такой перевод не мог не вызвать возражений, поскольку к описываемому моменту различные авторы уже пытались использовать это слово. Расходясь в трактовке самого понятия, все они были едины в том, что под информатикой следует понимать некое научное направление. На этом фоне новая информатика тоже стала претендовать на статус науки (научного направления), тем более что у ее англо-американского аналога слово *science*, основной смысл которого — наука, внесено непосредственно в название.

В результате сложилось явно ненормальное положение. Под наукой в русском языке понимается сфера человеческой деятельности, ставящей своей целью «описание, объяснение и предсказание процессов и явлений действительности» (БСЭ, 3-е издание), но никак не разработку и применение чего-либо. Поэтому не относятся к науке, скажем, дисциплины, занимающиеся изучением процессов изготовления изделий из дерева, шитья одежды. В речевой практике русского языка дисциплины, имеющие своим предметом специфическую профессиональную (чаще всего производственную) деятельность, принято называть словом *дело*. Так, в приведенных нами примерах речь шла о плотницком деле, портняжном деле. Существуют также горное дело, военное дело, морское дело и т. д. Значение слова *дело* как *дисциплины, круга знаний* можно найти во

всех толковых словарях русского языка.

Дело в указанном смысле принципиально отличается от науки: дело конструктивно, наука феноменологична, дело уделяет основное внимание технологии, наука — установлению закономерностей, дело может полностью измениться со временем, наука обязательно содержит стабильную основу. При этом дело зачастую тесно связано с некоторой наукой (или группой наук), как, например, горное дело с геологией, но отнюдь не сливается с ней. Таким образом, последовательно проводя указанное разделение, приходится признавать, что дисциплину или, если угодно, группу дисциплин, «занимающихся различными аспектами применения и разработки ЭВМ», уместнее всего назвать *компьютерным делом*.

Компьютерное дело, хотя и не является наукой, представляет собой весьма наукоемкую дисциплину. Из числа наук, с которыми связано компьютерное дело, можно назвать математику, психологию, лингвистику и ряд других.

Теперь возникает вопрос: насколько точно переводится термин computer science сочетание *компьютерное дело*? Согласно фундаментальному Оксфордскому словарю английского языка, одно из значений слова science — «определенная отрасль знаний либо область исследований». При этом словарь уточняет, что иногда «термин science употребляется распространительно для обозначения некоторого вида практических занятий, связанных с использованием определенных знаний и сознательным применением некоторых принципов». Из всего контекста статьи «science» в Оксфордском словаре становится ясно, что это понятие кроме основного смысла (наука) включает в себя и дело. Таким образом, термин computer science переводится на русский язык сочетанием *компьютерное дело* фактически без искажений. (Следует заметить, что совмещение значений наука и дело у слова science сбивает порой с толку и англоязычных читателей [2].) Следовательно, употребление термина *компьютерное дело* решает как задачу адекватного перевода, причем в обе стороны, так и задачу уточнения статуса соответствующей дисциплины.

Легко видеть, что термин *информатика* не справляется ни с первой, ни со второй задачей. Обратный перевод этого слова затруднен в высшей степени. Причина прежде всего в том, что под информатикой, как уже было сказано выше, разные русскоязычные авторы понимают совершенно различные дисциплины. Так, в [3] приведены шесть достаточно распространенных определений этого термина и предложено седьмое (кстати,

ВИНИТИ просто не мог дать упомянутому реферативному журналу наименование «Информатика», поскольку реферативный журнал с таким названием уже много лет издавался тем же ВИНТИ, хотя и по совершенно другой тематике!). Кроме того, в английском языке для большинства этих определений отсутствует готовый эквивалент (науки informatics англоязычные специалисты пока не знают). Задачу же уточнения статуса computer science термин *информатика* не только не решает, но и усложняет, поскольку, согласно большинству определений, информатика как раз является полноценной наукой.

Неясность этого термина вызывает неудобства далеко не у одних переводчиков. Постоянно встречающиеся сочетания типа «информатика и вычислительная техника» (например, в названии школьного курса), «информатика, вычислительная техника и автоматизация» (в названии отделения АН СССР), «программирование и информатика» (в названии словаря [1], хотя сам же составитель включил программирование в информатику) свидетельствуют о неуверенности в том, следует ли изучение вычислительной техники, других средств автоматизации и программирования относить к занятиям информатикой или нет. Из проведенного же терминологического анализа ясно видно, что если под информатикой понимается дело, то ответом на этот вопрос будет безусловное «да», а если наука — то безусловное «нет». Как следствие полная неразбериха возникает и при попытке отличить информатизацию от компьютеризации и автоматизации.

Встает закономерный вопрос: не следует ли вообще отказаться от термина *информатика*? Мы полагаем, что как раз наоборот: освободившийся от связи с computer science термин можно использовать именно для обозначения научного направления. Каким же должно быть его содержание? Не вдаваясь сейчас в сопоставление различных точек зрения (кроме семи определений из [3] еще несколько можно найти, например, в [4]), заметим лишь, что нам представляется наиболее перспективной трактовка, связанная с пониманием информатики как науки о формализованном общении. Мотивировка такого понимания требует, разумеется, отдельного обсуждения; здесь отметим лишь, что сравнительный анализ определений привел автора [3] к сходному подходу.

Вообще же обсуждаемая тема носит отнюдь не только академический характер. Так, в настоящее время широко известно, с какими трудностями сталкивается препода-

давание курса «Основы информатики и вычислительной техники» в рамках среднего общего и специального образования. Что понимается под информатикой в названии курса? Создается впечатление, что, согласно первоначальному замыслу, речь шла именно о преподавании основ науки информатики (хотя и не вполне ясно, как она понималась). В результате же получилось преподавание основ компьютерного дела (и, в частности, программирования), т. е. специфической профессиональной деятельности, причем не только для проявивших к этому склонность учащихся, но и в качестве одного из основных учебных предметов. Нам вполне понятен энтузиазм специалистов в области компьютерного дела, развернувших широкую кампанию за изучение его в средней школе. Однако если сегодня ради одной из профессий оказывается возможным нарушить один из главных принципов общего образования — давать фундаментальные, а не специфически профессиональные знания, то завтра может оказаться, что аналогичные кампании начнут автомобилисты, электронщики или кто-нибудь еще, и в случае их успеха выпускники наших школ будут обладать конгломератом сведений из различных профессиональных областей, владея соответствующими профессиями, естественно, на дилетантском уровне.

Заметим, что особенно странно выглядит «компьютерный всеобуч» в средней школе на фоне того, что высшая школа профессионалов в области компьютерного дела практически не готовит. Много ли в нашей стране вузов, ведущих подготовку по специальности «Программирование на ЭВМ», в сопоставлении с реальной потребностью в таких специалистах?

Пока что сложившаяся в средней школе ситуация вызывает массовое недовольство как преподавателей, так и учащихся. При-

чина этого понятна: терминологическая путаница, о которой выше шла речь, повлекла за собой смешение трех фактически почти не связанных друг с другом новаций: использования ЭВМ как технического средства обучения; преподавания основ компьютерного дела; преподавания основ информатики.

На наш взгляд, первое уместно всюду, где имеется соответствующее техническое и программное обеспечение, второе — в кружках, на факультативах и в рамках специального образования (для будущих специалистов по работе с ЭВМ), третье — по всей видимости, и в школе, но не ранее того как завершится процесс становления новой фундаментальной науки — информатики.

Литература

1. Борковский А. Б. Англо-русский словарь по программированию и информатике (с толкованиями). М.: Русский язык, 1987.
2. Ball G. Report on "Birds of a feather" session: where is the "science" in computer science? // Proc. Software Process Workshop, Egham, Febr., 1984. Silver Spring (Md), 1985. P. 161—162.
3. Белошанка В. О языках, моделях и информатике // Информатика и образование. 1987. № 6. С. 12—16.
4. Кухтенко А. И. Об интегрирующей (системообразующей) роли кибернетики // Управляющие системы и машины. 1987. № 6. С. 28—38.

От редакции.

Можно спорить о том, насколько точно и однозначно интерпретация тех или иных терминов и понятий, окончательно ли они вошли в нашу жизнь, имеет ли смысл пытаться что-то менять и, наконец, насколько новые варианты совершеннее старых. Нельзя не признать, однако, что проблема трактовки терминов, их соответствия зарубежным аналогам существует и требует обсуждения.

Синие яблоки

Раньше подобные плоды не встречались в природе, а недавно стали попадаться в некоторых местах удивительно часто — в среднем одно синее на 75 обычных. Появление этих экзотических «фруктов» не следствие генноинженерных экспериментов или злоупотребления инсектицидами и стимуляторами, а дело рук американских электронщиков. По их мнению, шарообразные приборы, содержащие пьезоэлектрический

ЧТО?
МОЖЕТ
ЭВМ

датчик ударных воздействий, микропроцессор и память, лучше всего подходят для анализа причин повреждения яблок на пути от сада к прилавку. Пройдя этот путь, приборы накапливают в своей памяти информацию о всех испытанных ударах, что позволяет исследователям усовершенствовать технологию обработки настоящих плодов.

А синий цвет — вы уже догадались? — помогает выбирать приборы из кучи яблок в конце пути.

ЭВМ в народном хозяйстве

Овладение хозрасчетными методами хозяйствования, их эффективное использование — одна из ключевых задач экономической реформы. Важная роль в её решении отводится вычислительной технике. Предлагаемая авторами экспертно-обучающая система позволяет не только поднять на более высокий уровень управление хозяйственной деятельностью предприятий, но и что не менее важно, организовать экономическую учебу, подготовку управленческого персонала и более широкого круга специалистов к работе в новых условиях.

Н. ИСАЕВА, Н. МИРОНОСЕЦКИЙ, Л. ПАРФЕНОВА, Ю. ЩЕГЛОВ

Компьютер и хозрасчет

(элементы компьютерного учебника)

Внедрение экономических способов управления народным хозяйством, переход предприятий на полный хозрасчет и самофинансирование требуют разработки новых экономических концепций управления предприятием и высокой профессиональной грамотности в применении соответствующих методов. Развитие гласности и перестройка народного хозяйства обуславливают широкое участие трудящихся в создании и развитии конкретных методов управления, в формировании планов предприятия (текущих, годовых, перспективных, стратегических), а также мероприятий по их реализации. Хозяйственная самостоятельность предприятий требует умения использовать появившуюся экономическую свободу для улучшения работы предприятия, что в свою очередь диктует необходимость разработки экономико-математического инструментария, обеспечивающего улучшение планирования и управления производством с обязательным участием трудового коллектива.

Для реализации нового хозяйственного механизма требуется высокий профессионализм, отсюда вытекают следующие задачи: создание «демократичной» окружающей обстановки для ознакомления с экономико-математическим аппаратом, рассчитанным на широкое применение специалистами предприятий, непрофессионалами в экономико-математических методах;

обучение использованию этими методами, включая режимы диалога на этапе формирования модели с подключением режима оптимизации;

создание стендов, позволяющих не только испытывать создаваемые модели в целях их адаптации к конкретным производственным условиям, но и проводить глубокие иссле-

дования новых экономических механизмов управления;

адаптация модели к реальным производствам в различных сценариях.

Для кого и для чего
предназначена система

Опишем подход к созданию электронного учебника на базе ПЭВМ, содержащего комплекс средств многоцелевого применения по следующим направлениям:

первоначальное обучение новому хозяйственному механизму в соответствии с Законом о государственном предприятии (объединении) и межотраслевыми инструкциями, играющими роль подзаконных актов; углубленное изучение принятых в законе форм хозрасчета;

адаптация хозяйственного механизма к отраслевым инструкциям и к индивидуальным нормативам, спущенным предприятиям министерствами;

разработка реальных агрегированных планов с широким применением деловых игр, позволяющих учитывать противоречивые интересы участников трудового процесса на предприятии;

анализ выполнения планов с применением средств имитационного моделирования;

проведение исследовательской работы по виртуальной модификации экономического механизма и возможным последствиям их внедрения, проверке различных гипотез, касающихся тенденций действия индивидуальных нормативов на основные показатели деятельности предприятия;

использование в качестве стенда для отработки различных сценариев стратегического развития предприятия.

Разработанную автоматизированную экспертно-обучающую систему анализа хозяйственной деятельности предприятия в условиях хозрасчета «ЭКОНА» можно использовать в экономическом образовании школьников, студентов, аспирантов, научных сотрудников и специалистов предприятий.

С чего начинать

Для быстрого освоения и эффективного использования данная система содержит специальную учебную подсистему. Цель учебной подсистемы — автоматизированное обучение и контроль знаний об основных технико-экономических показателях, определяющих формирование и распределение хозрасчетного дохода предприятия.

Весь процесс обучения разбит на этапы — один этап соответствует изучению одного показателя (искомого). Количество и последовательность шагов определяются выбранным уровнем сложности. На первом уровне сложности требуется выбрать из предложенного списка те показатели, которые нужны для вычисления искомого (1-й шаг). На втором уровне, когда уже определены все составляющие искомого показателя, необходимо правильно сформировать на экране формулу (2-й шаг). На третьем уровне сложности обучаемому предлагается дополнительный вопрос по теме занятий (3-й шаг). Обучаемый работает с клавиатурой и компьютерным экраном, разделенным на четыре окна, каждое из которых несет определенную смысловую нагрузку (рис. 1).

В окне «цель» выводится наименование и мнемокод искомого на данном шаге показателя.

Во втором окне — «задача» — перечисляется ряд показателей, из которых можно рассчитать искомый.

Все приводимые показатели помимо своего сокращенного наименования имеют уникальный мнемокод.

Третье окно — «диалог» — используется для принятия ответов, для вывода результатов анализа ответов, а также в случае, если пользователь хочет получить дополнительную информацию в качестве помощи для решения текущей задачи.

Четвертое окно экрана — «счетчик» — содержит текущее количество очков.

В системе реализуется игровой принцип, согласно которому необходимо набрать как можно больше очков. Очки начисляются за правильный ответ и за правильно построенную формулу на каждом шаге обучения. Соответственно снимаются очки за неправильные ответы.

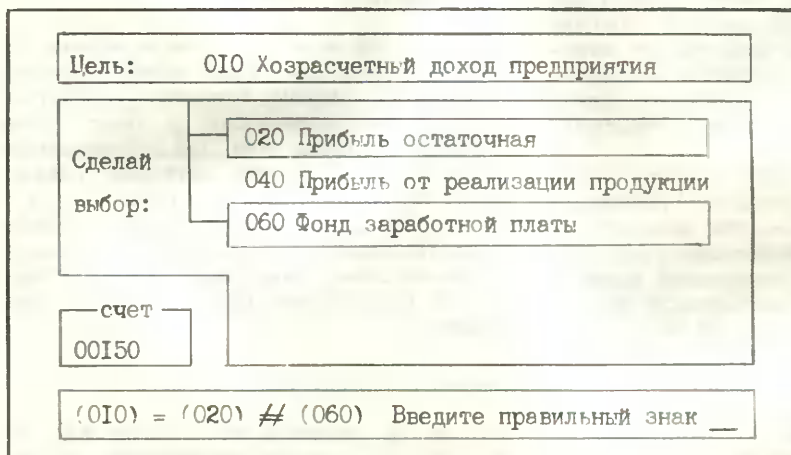
После того как пользователь закончил работу с очередным показателем, он переходит на следующий этап обучения, где в качестве искомого выбирается один из показателей, необходимых для расчета предыдущего. Оценивать результаты обучения можно по количеству набранных очков. Кроме того, система ведет протокол обучения и может представить развернутый анализ результатов.

Учебная подсистема кроме контрольно-обучающей части содержит учебный пример по вычислению и распределению хозрасчетного дохода предприятия по двум различным моделям хозяйственного расчета. На экране визуализируются схемы, изображенные на рис. 3, и дается возможность моделировать различные ситуации.

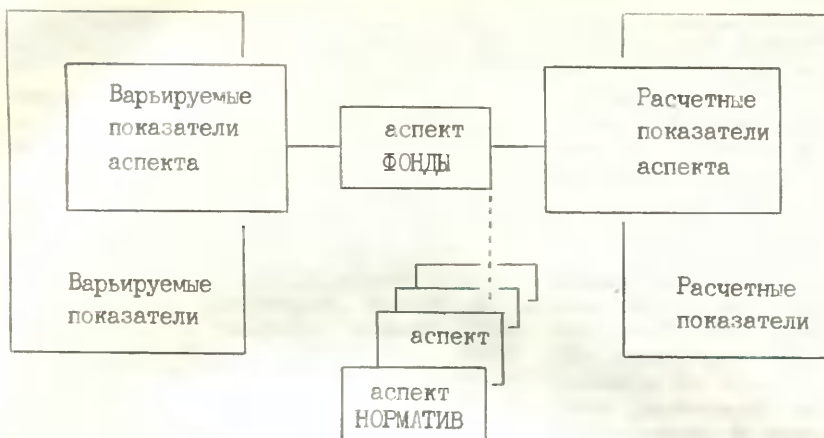
Что определяет множество показателей, с которыми вы будете работать

После того как вы решились использовать нашу систему, требуется выбрать аспект рассматриваемой проблемы. Есть воз-

109



1. Структура экрана обучающей подсистемы.



2. Общая структура системы.

110 возможность сосредоточиться на следующих направлениях:

анализ влияния различных нормативов на основные технико-экономические показатели предприятия (аспект «Нормативы»);

работа с производственными фондами (аспект «Фонды»);

работа с банковским кредитом (аспект «Кредит»);

анализ влияния выпуска продукции на интегрированные финансовые показатели (аспект «Номенклатура»);

анализ основных направлений распределения балансовой прибыли и образования прибыли, остающейся в распоряжении предприятия (аспект «Прибыль»);

анализ формирования фонда заработной платы (аспект ФЗП);

анализ образования фондов экономического стимулирования (аспект ФЭС);

прогнозирование динамики развития предприятия (аспект «Динамика»).

Выбор каждого из перечисленных направлений определяет множество показателей, значения которых могут изменяться по желанию пользователя системы. Множество этих показателей в дальнейшем называется «варьируемые показатели аспекта». Объединение их всех по различным аспектам дает множество варьируемых показателей системы.

Варьируемые показатели аспекта однозначно определяют множество расчетных показателей аспекта, т. е. тех показателей, которые изменяют значения при изменении значений варьируемых показателей аспекта. Эти показатели будут называться расчетными показателями аспекта. На рис. 2 представлены общие принципы организации показателей.

После того как выбран требуемый аспект проблемы нет необходимости иметь дело со всем множеством показателей системы. Сис-

тема активизирует только необходимое множество показателей, выбирая варьируемые показатели аспекта, расчетные показатели аспекта, а также некоторые другие, которые могут представлять интерес при рассмотрении проблемы в данном аспекте. Все эти показатели являются активными для данного аспекта и могут быть визуализированы на экране видеотерминала.

Как представляются показатели на экране

Система формирует экран в соответствии с выбранным аспектом. В тех случаях, когда количество активных показателей большое и превышает возможности экрана, используется постраничная организация или разделенный экран. В первом случае нажатие соответствующей клавиши переводит экран на нужную страницу. Во втором случае варьируемые и расчетные показатели на экране располагаются в соответствующих местах, называемых окнами.

Что такое окно

Окно — это выделенная часть экрана, где представляются значения и идентификаторы показателей. Любые активные показатели могут быть установлены в окне путем листания страниц или последовательным просмотром множества активных показателей. Листание страниц осуществляется с помощью клавиши, указывающей номер страницы. Последовательный просмотр реализуется клавишами, прокручивающими все множество показателей вперед, назад, влево, вправо.

Зачем нужны два окна

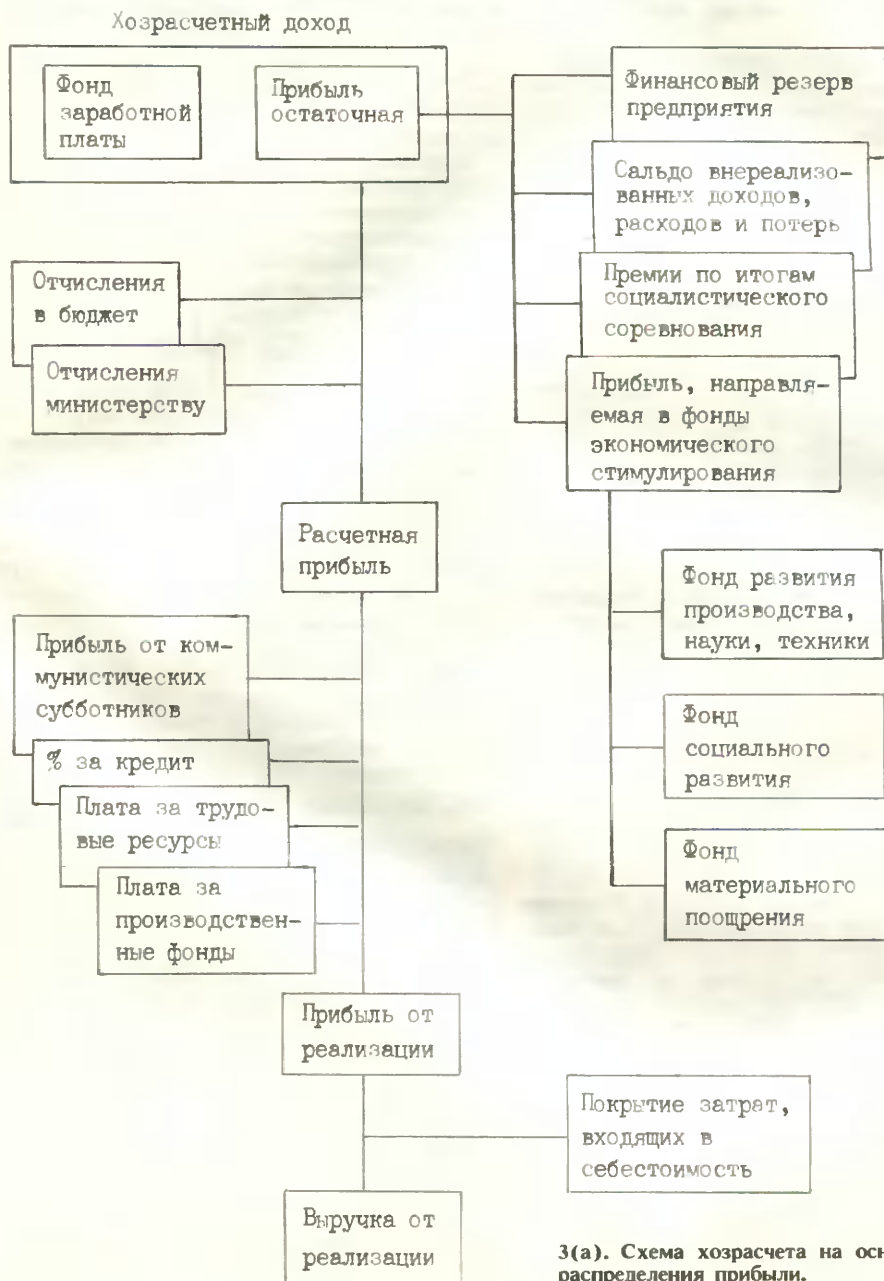
Если вы хотите, изменяя один или несколько варьируемых показателей, наблю-

дать, как влияют эти изменения на интересующие вас расчетные показатели, то целесообразно воспользоваться возможностью работы с двумя экранными окнами. В первом окне вы устанавливаете интере-

сующие вас варьируемые показатели, во втором — расчетные.

В первом окне с помощью курсоралокатора (специальный эффект, выделяющий на экране некоторый показатель) указы-

ПЕРВАЯ МОДЕЛЬ ХОЗРАСЧЕТА



3(а). Схема хозрасчета на основе нормативного распределения прибыли.

ваете тот показатель, который хотите изменить, и вводите новое значение. Расчетные показатели тут же пересчитываются, и вы наблюдаете результат ваших изменений во втором окне.

Как представляются результаты вариантных расчетов

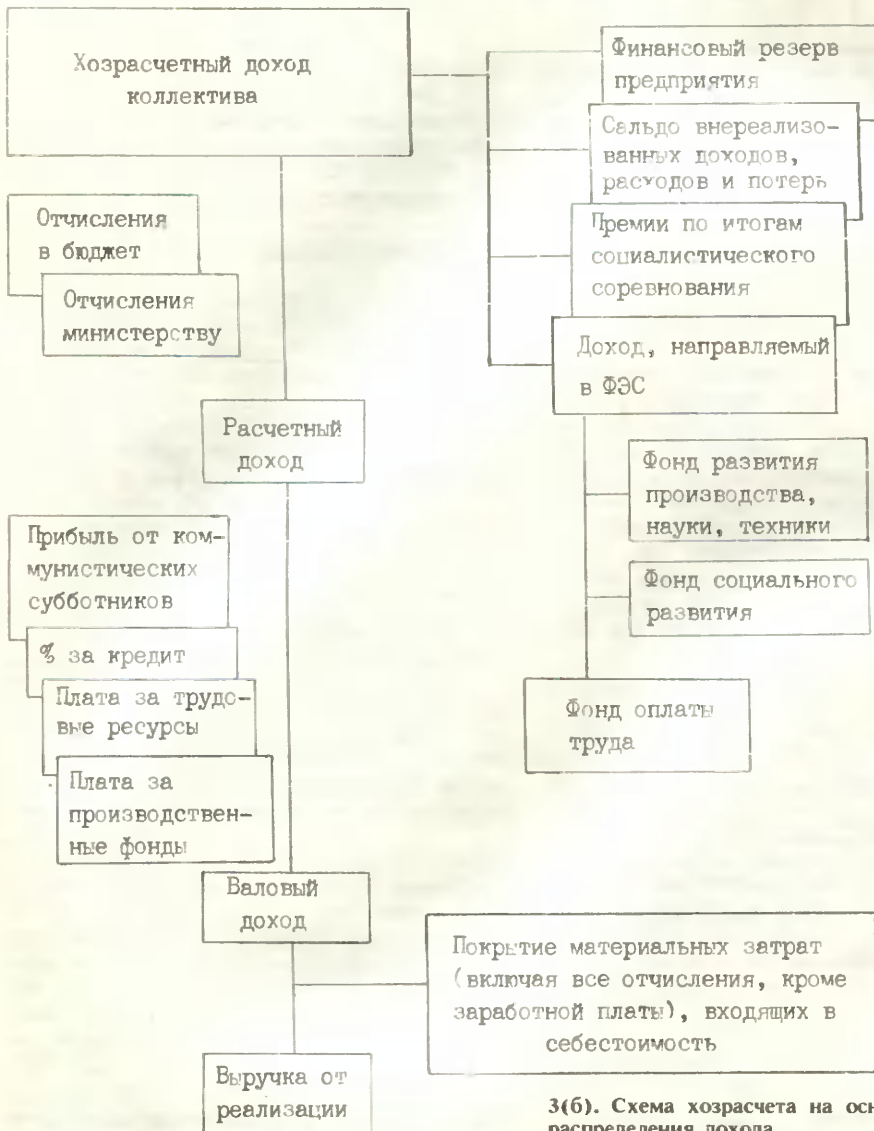
Кроме того, что вы непосредственно наблюдаете изменения интересующих вас расчетных показателей на экране, более глубо-

кие выводы можете получить, если результаты расчетов будут представлены в виде обработанных отчетов или графиков.

Каким образом связаны показатели между собой

Все множество показателей системы связано между собой функциональной зависимостью. На рис. 3 отображены два варианта распределения хозрасчетного дохода предприятия, предусмотренные Законом о

ВТОРАЯ МОДЕЛЬ ХОЗРАСЧЕТА



3(б). Схема хозрасчета на основе нормативного распределения дохода.

государственном предприятии: первая модель основана на нормативном распределении прибыли, вторая — на нормативном распределении дохода. Учебный пример системы включает перечисленные два подхода и использует приведенные схемы при ведении диалога для визуализации функциональной зависимости.

В силу особенностей сегодняшнего состояния экономической теории и практики (значительное обновление и большой динамизм) система снабжена возможностью быстрой и удобной корректировки функциональной зависимости показателей. С этой целью в системе каждому из показателей приписан его код, с помощью которого значения расчетных показателей могут быть выражены через значения варьируемых показателей и некоторых постоянных параметров системы. В любой момент с помощью специального переключателя вы можете визуализировать и наблюдать определение функциональной зависимости показателей на экране.

Как взаимодействовать с системой

Для того чтобы работать с системой, вы должны быть зарегистрированы как ее пользователь и иметь пароль. Можно выбрать различные режимы взаимодействия.

Режим с меню-путеводителем предполагает наличие минимальной подготовки пользователя и представляет возможность выполнять основные функции как по изменению значений варьируемых показателей, так и получению результатов изменений в виде значений отдельных показателей, отчетов или графиков. В этом случае по подсказке на экране вы выбираете необходимую функцию системы и нажимаете соответствующую клавишу. Система осуществляет необходимые действия или уточняет функцию выводом следующего кадра с подсказками.

Режим команд предполагает более расширенные возможности, но требует более квалифицированной работы с системой. В распоряжении пользователя более 40 команд, позволяющих выполнять основные функции системы, модифицировать и развивать ее.

В любом случае сеанс взаимодействия с

системой начнется в режиме меню. В начале сеанса вы можете попросить восстановить ваш предыдущий сеанс (если такой был) или начать новый. При наличии нескольких предыдущих сеансов, что соответствует нескольким незаконченным вариантам ваших расчетов, система предложит выбрать интересующий вас.

С целью хранения и последующей идентификации сеансов, если вы начинаете новый сеанс, вас попросят дать ему имя, по которому вы и будете в дальнейшем его отличать.

Вопрос о сохранении сеанса решается при выходе из системы, когда вы на соответствующий вопрос системы указываете сохранить данный сеанс или нет. Результаты сохраняются (если вы это указали) в специальном файле сеанса.

Доступ к данным

Если вы считаете необходимым ограничить доступ со стороны к значениям показателей, которые имеются в системе, можно выбрать один из двух или комбинацию следующих вариантов:

защитить вход в систему паролем,

определить степень доступа к чтению данных системы дифференцированно для различных пользователей. В этом случае каждый пользователь получает код доступа от 1 до 15: код 15 обеспечивает доступ к любым данным, коды в промежутке от 1 до 14 обеспечивают доступ к показателям с соответствующим уровнем защиты.

Наконец, вы можете хранить значения показателей на отдельной дискете у себя в столе.

Некоторые технические ограничения

Описанная система реализована для 16 разрядных персональных ЭВМ. Необходимым условием ее применения является наличие оперативной памяти в объеме не менее 512К байт и памяти на внешнем накопителе. Поддерживается работа с цветным монитором.

Справки о системе по новосибирскому телефону 35-53-43 (Н. А. Исаева)

В. ТИХОНОВ

Можно ли выиграть... знания?

Как и чему учить, если есть на чем учить.

Итак, у нас есть компьютер, можно изучать его, а можно изучать нечто с его помощью. Но если изучать легче архитектурно более простую машину, то для создания удобной и методически продуманной обучающей программы необходим достаточно мощный компьютер, имеющий большую оперативную и емкую и быструю внешнюю память, высококачественный цветной графический терминал, удобные устройства для обеспечения интерактивной работы.

В школы поставляются обычно дешевые и простые ЭВМ, возможностей которых часто бывает недостаточно для работы с существующими обучающими программами. К тому же обучающие программы, получившие распространение в нашей стране в настоящее время, в подавляющем большинстве представляют собой упрощенную систему работы с дисковыми или ленточными файлами, в которых хранится та же информация, что и в обычном учебнике, только доступ к ней имеют лишь те, у кого есть возможность поработать с компьютером. Не вдаваясь в технологические подробности написания такого рода обучающих программ, скажем лишь, что, как правило, наиболее трудоемкой частью таких систем является работа по перепечатке текста учебника в память ЭВМ.

Возникает парадоксальная ситуация: есть компьютеры, но нет полноценной возможности улучшить с их помощью учебный процесс. В подобной ситуации оказалось компьютерное образование и в США. Правда, с небольшими поправками: многие колледжи и университеты имеют возможность приобрести компьютеры достаточной

мощности, но такие ЭВМ кроме учебных имеют и богатую библиотеку намного более увлекательных игровых программ, которые в буквальном смысле вытесняют с дисков своих обучающих конкурентов. И наиболее ярко эта борьба проявляется в высшей школе.

Вот мы и подошли вплотную к причинам, мешающим развиваться обучению с помощью компьютеров. Попытаемся классифицировать их. Первой и наиболее серьезной является несовпадение ожидаемого педагогами эффекта от внедрения новой техники и тем, что эта техника действительно может дать. Нередки ситуации, когда школьнику, прежде чем он начнет получать предметные знания, приходится вначале изучить принципы работы обучающей системы. Подобная компьютеризация порой сильно деформирует задуманную авторами обучающей программы методику обучения.

Второй причиной является слабо развитое обучающее программное обеспечение. Это наиболее серьезная проблема, препятствующая возникновению активного и глубокого интереса к компьютерам и компьютерному обучению у школьников и педагогов.

И третья, самая, пожалуй, неожиданная причина — конкуренция игровых программ, победить в которой можно лишь созданием увлекательных обучающих программ с игровым сюжетом.

В сущности, у всех этих проблем один корень — отсутствие хороших обучающих программ, которые могли бы конкурировать с игровыми, заставляя студентов получать необходимые знания в увлекательной форме. Помочь выйти из создавшегося положения могут такие на первый взгляд простые меры, как создание обучающих программ,

способных вытеснить игровые не только с дискет студентов, но и с большого рынка прикладных программ. Однако сильные программисты неохотно проектируют обучающие программы, так как в США, например, проводимые до последнего времени конкурсы неадекватно оценивали качество создаваемого обучающего программного обеспечения, принимая во внимание в первую очередь коммерческие перспективы пакетов. Поэтому можно предположить, что в области проектирования обучающего ПО не реализованы большие потенциальные возможности.

В нашей стране, несмотря на нехватку компьютеров, тоже проводятся конкурсы программ. И даже готовится команда для участия в международных соревнованиях компьютерных программ модного в определенных кругах «Боя в памяти». Об этой уникальной игре много писалось и говорилось, и это не случайно. «Бой в памяти» — наиболее яркий из существующих сейчас примеров обучающих программ с игровым пользовательским интерфейсом. Говоря проще, эта программа учит программировать на некотором алгоритмическом языке, но делает это не напрямую, перечисляя все премудрости языка, а косвенно, заставляя игрока (или учащегося?) искать и находить такие языковые особенности, которые позволят ему выиграть бой с другой программой, автор которой этот язык и поведение программ написанных на нем, пока знает лучше.

Но пока в этих конкурсах, увы, участвуют в основном игровые и сервисные программы, хотя некоторые из них и содержат элементы обучения. Итак, конкурсы программ проводятся. А вот конкурса обучающих программ, которые могли бы любому растолковать теорию относительности или интересно рассказать о развитии эмбриона, у нас пока нет. Кстати говоря, не было до последнего времени такого конкурса и в США. В 1987 г. журналом «Эдьюком» был организован первый такой конкурс, с результатами которого мы и хотим вас познакомить [1]. И пусть описание программ, занявших призовые места, послужит толчком к развитию подобных конкурсов и у нас в стране.

Основная цель конкурса — стимулировать создание высококачественных обучающих программ. Понимая, что одними призывами писать хорошие программы не обойтись, организаторы конкурса поставили перед

собой три задачи, решение которых помогло бы преодолеть неприязнь программистов к этому разделу прикладного программирования.

Первая — разработка, широкая публикация и обсуждение критериев оценки обучающего программного обеспечения.

Вторая — отбор показательных образцов обучающего ПО в качестве примеров для его последующих разработчиков и участников конкурса.

Третья — награждение участников конкурса, предложивших наиболее удачные программы или идеи их создания, а также тех, кто сумел найти наиболее удачное приложение уже существующим необучающим программам для обучения.

В конкурсе мог принять участие каждый желающий. Организаторы с самого начала решили, что внимания заслуживает не только написанные, отлаженные и документированные обучающие программы, но и перспективные идеи, на основе которых можно построить интересную обучающую систему, потому что чаще всего программам такого рода не хватает как раз «изюминки».

Конкурс проходил в два этапа. На первом эксперты оценивали обучающий пакет с точки зрения его работоспособности и оригинальности. Во втором раунде каждую программу тщательно изучали по крайней мере два специалиста — психолог и педагог-предметник.

Заключительный этап конкурса был организован следующим образом. Эксперты независимо оценивали каждую из программ, затем собиралась вся группа и сравнивала обучающие программы одной подгруппы между собой, отбрасывая каждый раз худшую. Таким образом, в каждой подгруппе оставались только лучшие программы.

В процессе отбора лучших обучающих программ критерии, опубликованные до конкурса, менялись. Большое количество самых разнообразных пакетов, поступивших на конкурс, продемонстрировало, что исходные критерии не совсем точно отражают качество оцениваемого ПО.

Сначала основным критерием было отсутствие в программе ошибок. Затем стали учитывать простоту работы с программой, дружелюбность программы по отношению к учащемуся (исправление явных опечаток и других случайных ошибок), ее быстрдействие, качество вспомогательных материалов, облегчающих изучение предлагаемого пакета, наличие в программе контекстозависимых подсказок, дизайн.

Во втором раунде конкурса, после отсева технологически несовершенных программ, не имеющих интересной идеи, критерии были

смещены от качества программы к качеству обучения с их помощью. В классе обучающихся программ это, например, методика обучения и достижимость конечной цели обучающей программы (для формализуемых дисциплин это, например, достижимость решения).

Однако для многих обучающихся программ невозможно определить точную конечную цель работы. Особенно явственно это проступило в программах по тем дисциплинам, в которых трудно формализовать конечную цель и способы ее достижения. По этой и другим, более субъективным причинам многие обучающие программы не могли быть точно оценены по существующим критериям и оценивались экспертами как уникальная система, позволяющая повысить качество образования.

Все программы, поступившие на конкурс, делились на две категории: оригинальные обучающие программы и адаптированные программы. К первой категории были отнесены программы, которые были реализованы автором или группой авторов от идеи до готового загрузочного модуля самостоятельно.

Во вторую группу попали обучающие программы, авторы которых сумели удачно адаптировать уже существующие пакеты к конкретной методической области или создали удобную оболочку, позволяющую превратить исходную программу в интересную обучающую систему.

В первой группе программы были разбиты, в свою очередь, на четыре класса по принципу их работы:

1. Обучающие программы (Tutorial software)

Программы, относящиеся к этому классу, предназначены для обучения студентов путем предоставления им фактического материала в той или иной форме и контроля знаний с помощью тестов, практических заданий и упражнений. Обучение происходит под контролем преподавателя или компьютера и дает хорошие результаты при изучении таких дисциплин, как вычислительная математика, статистика, аналитическая химия, и др., имеющих формализованное представление знаний. Постоянный контроль уровня знаний студентов со стороны программы, обеспечивая обратную связь в течение всего курса обучения, позволяет корректировать сложность заданий и темп продвижения по изучаемому курсу.

2. Моделирующие программы (Simulation software)

Программы этого класса являются конкретной реализацией какого-либо раздела одной дисциплины, позволяющие детально

изучить его. Моделирующие программы очень похожи на динамичный мультфильм по конкретной дисциплине, сценарий которого определяется воздействиями учащегося на модель изучаемого раздела во время ее работы. Манипулируя таким образом доступными для изменения параметрами, студент по реакции моделирующей системы определяет диапазон их допустимых изменений и осознает сущность происходящих под его управлением процессов.

Например, в модели экологической системы студент биологического факультета может изменить процентный состав грызунов и хищников и проследить за всеми изменениями, которые произойдут в устойчивой до этого системе.

Моделирующие системы, как и другие обучающие программы, могут предоставлять для изучения как небольшие разделы дисциплины, так и большие процессы, требующие знания нескольких предметов. Такие процессы могут сопровождать студента в виде постоянно подстраиваемых им моделей на протяжении всего курса обучения, сохраняя в себе весь комплекс знаний, полученных студентом за это время.

3. Инструментальное программное обеспечение (Tools software)

Инструментальное ПО не имеет, как правило, какой-либо ориентации на конкретный изучаемый курс или методику обучения. Подобные программы предоставляют своим пользователям возможность самостоятельно решать определенные задачи на более короткое время и с меньшими усилиями, избавляя их от рутинной вычислительной и статистической работы и предоставляя свободу в выборе методов решения конкретных задач и простор для творчества. Инструментальное ПО дает прекрасную возможность научить студентов конструированию пользовательской среды и развить у них творческое мышление.

4. Интегрированные обучающие программы (Integrated software)

К последнему классу были отнесены все программы, которые комбинировали в себе признаки двух или более вышеперечисленных классов или включали в свой состав уже существующие модули.

В группе адаптированных обучающих программ (прикладных) разбиения на подразделы не было. Основными критериями оценки попавших в эту категорию программ были качество и дизайн пользовательского интерфейса, предоставляемого разработчиком, а также остроумие в применении имеющихся программных пакетов к изучению конкретных дисциплин.

Выделение подобного рода программ и

идей по их написанию в отдельный раздел подчеркивает то внимание, с которым организаторы конкурса отнеслись к тем его участникам, которые, возможно, никогда в жизни не напишут хорошую программу, но зато имеют прекрасные идеи, как это можно сделать. Яркими примерами удачного применения готовых прикладных задач в обучающем процессе являются использование электронных таблиц для составления расписаний, адаптация управленческих пакетов, генерация ПО.

Кроме деления на оригинальные и адаптированные программы все поступившие пакеты классифицировались по дисциплинам, изучение которых они поддерживают. Так, на конкурс поступили 25 работ по социальным дисциплинам (социология, политические науки), 62 естественнонаучные работы (химия, биология, физика), 20 гуманитарных (философия, история). 16 языковых (английский язык) и 40 математических (математика, вероятность, статистика). Из них 31 работа была отнесена к классу обучающихся программ, 28 — моделирующих, 65 работ были предназначены для исследования окружающей среды или являлись инструментом для пользователя, 33 пакета были интегральными и 18 — модифицированными прикладными задачами.

Поэтому каждая программа участвовала как бы в двух конкурсах: в своем классе (обучающая, моделирующая и т. д.) и в своей дисциплинарной секции. Например, обучающая программа «Стоячие волны» оценивалась как в группе моделирующих пакетов, так и в группе программ по физике.

Первое место в классе обучающихся программ занял пакет «Химия». Его основой является динамичная игра в «догонялки» в пределах запутанного лабиринта. Противником студента, представленного... колбой с каким-либо химическим реактивом, является

весьма агрессивный стакан с нейтрализующим веществом. Стакан пытается догнать «колбу—студента» и выиграть таким образом игру. Продвижение по лабиринту осложняется тем, что он заполнен различными химическими веществами, взаимодействие с которыми может быть как нежелательным для колбы, так и, наоборот, подкрепляющим ее состояние (например, увеличивается pH содержимого колбы).

Программа размещается на десяти флоппи-дисках и включает справочник по химическим элементам, каталог реактивов, измерительную систему и игру в химические реакции. Сложность программы может изменяться в зависимости от уровня знаний обучаемого.

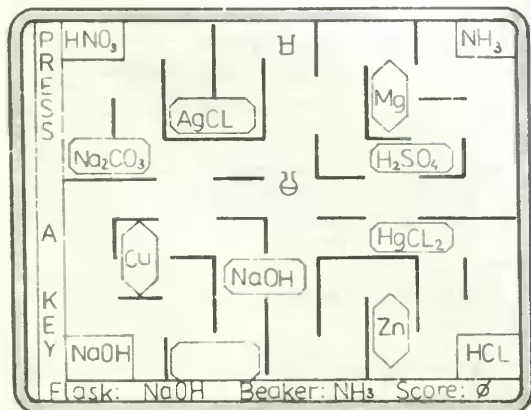
Лучшей моделирующей программой стал пакет, позволяющий наблюдать и изучать механические свойства мускулатуры. Программа может быть использована в курсах зоологии, анатомии и физиологии. Простота работы с пакетом и хороший пользовательский интерфейс — неоспоримое достоинство. В пакете представлено несколько вариантов программ, один из которых выглядит следующим образом: на экране показан движущийся велосипедист, мускульная активность которого и подвергается изучению. Рядом представлены графики зависимости скорости велосипедиста и совершенной им работы от времени. Учащийся с помощью простых одноклавишных команд может изменять усилия велосипедиста или скорость движения и получать свой график, отличающийся от идеального. В процессе моделирования проясняется зависимость этих параметров друг от друга.

Лучшей моделирующей программой по физике признан пакет «Стоячие волны». С помощью этого пакета студенты могут изучать такие важнейшие понятия классической физики, как интерференция, дифракция, эффект Доплера для звука и света; смогут наблюдать движение идеального газа, диффузию, гравитацию, различные электромагнитные явления.

Программа позволяет производить моделирование в замедленном и ускоренном темпах, останавливать процесс и изменять в это время параметры моделируемых явлений (например, форму волны).

Звуковое оформление делает воздействие системы еще более убедительным.

Не оставлена без внимания и математика. Лучшим математическим инструментом был признан пакет для исследования малых групп. Студент получает возможность оперировать математическими понятиями и сразу же проверять их доказуемость и



реализуемость, не совершая рутинной вычислительной работы. Четкая цель программы «Исследование малых групп» и невозможность достижения этой цели некомпьютерным методом обучения выгодно отличают этот пакет.

Среди программ по гуманитарным наукам был отмечен пакет, моделирующий социальную структуру общества во Франции времен Луи XIV. Игрок «живет» жизнью французского буржуа и принимает экономические и социальные решения, которые влияют на судьбу страны. В процессе обучения ему предоставляется обширная справочная информация, вводящая его в курс

событий того времени. В конце игры сообщается, насколько точно студент осуществлял свою миссию.

Кроме того, было представлено большое количество моделирующих пакетов по различным дисциплинам и инструментальных программ, облегчающих обучение. Это и такие известные, как MICROCALC и своеобразные программы для моделирования генетического наследования животных или развития штаммов в определенных условиях, и многие другие.

Словом, конкурс такого рода — дело чрезвычайно полезное.

ВНИМАНИЮ ПРЕДРИЯТИЙ, ОРГАНИЗАЦИЙ, УЧЕБНЫХ ЗАВЕДЕНИЙ,

**желающих эффективно использовать вычислительную технику
в подготовке и переподготовке рабочих,
специалистов, учащейся молодежи.**

***Успешно справиться с этой задачей
вам помогут специалисты межвузовского
научно-технического объединения***



«И Ф О Р К О М».

Основным видом деятельности МВ НТО «ИФОРКОМ» является разработка обучающих программ по дисциплинам, изучаемым в школах, училищах, техникумах, вузах.

***Обучающие программы, разработанные МВ НТО «ИФОРКОМ»,
отличаются:***

**применением современных методик компьютерного обучения;
организацией дружественного диалога с обучаемым;
высококачественным программированием;
максимальным использованием возможностей компьютера;
широким применением инструментальных средств и возможностей локальных
сетей.**

Сотрудники МВ НТО «ИФОРКОМ» являются высококвалифицированными специалистами в этой области, имеют опыт международного сотрудничества. Обучающие программы, разработанные ими, были отмечены медалями ВДНХ СССР в 1987, 1988 гг.

Если Вас заинтересовали возможности МВ НТО «ИФОРКОМ», обращайтесь по адресу:

620012, г. Свердловск, ул. Машиностроителей, 11,
МВ НТО «ИФОРКОМ», тел 37-35-48.

Луч вместо провода

Некоторые высококачественные бытовые приборы помимо панели управления на корпусе снабжаются дистанционным пультом. С помощью такой выносной панели управления можно, не вставая с кресла, переключить телевизионный канал, перемотать ленту видеомэгнитофона или настроить приемник на другую станцию. Дотянуться до всех этих приборов вам поможет инфракрасный луч дистанционного пульта управления.

Лучшие образцы устройств дистанционного управления имеют встроенный процессор, а это по сути маленький компьютер, который может управлять своим прибором по заранее заданной программе. Кроме того, несколько таких устройств можно подключить к мощному персональному компьютеру, который сможет включать и выключать с их помощью сразу несколько бытовых приборов по более сложной и длительной программе. Единственное, что может помешать слаженной работе такого комплекса, — это преграда на пути теплового луча. Так что, если не хотите переходить на ручное управление, — сидите смиренно!

Без чего не может ЭВМ?

«Голый» компьютер, у которого сразу после включения в память нет ни одной программы, может понять не более двухсот очень простых команд. Только после того, как в его память будут переписаны программы, компьютер «оживет». Но программа, какой бы замечательной она ни была, не сможет ни задать коровам корм, ни включить лампы в теплице, ни посадить самолет.

Создано достаточно много устройств, назначение которых — сопряжение ЭВМ с внешними устройствами, которые и определяют «лицо» компьютера. Эти устройства, в зависимости от назначения, называются интерфейсами или контроллерами. Быстрое увеличение числа устройств сопряжения привело в свое время к созданию унифицированного модуля САМАС.

Однако для таких простых случаев, как управление электрической мешалкой или слежение за температурой раствора, САМАС

слишком дорог и громоздок. Для таких случаев фирма «Компьютер континьюм» (США) предлагает использовать локальную шину LAB 4F. Шина позволяет преобразовывать сигналы в стандартный TTL — уровень и управлять устройствами на расстоянии до ста футов. Кроме того, с помощью этой шины можно нара-

ЧТО?
МОЖЕТ
ЭВМ

щивать мощность компьютера, подключая к нему дополнительные устройства управления. От обычной шины LAB 4F отличается тем, что являются не просто кабелями, а кабелем, внутри которого находится компьютер, управляющий его работой.

Ну а если вы приобрели компьютер только для того, чтобы сделать из него лабораторную установку для сбора и анализа данных эксперимента, то удобнее всего будет воспользоваться готовой платой, которая превратит вашу ЭВМ в высокопроизводительный мультимедийный измеритель. Для компьютеров IBM PC и совместимых с ним такую плату разработала фирма ACE. Плата имеет стандартный разъем, легко устанавливается в компьютер и является, по существу, одноплатной ЭВМ, облегчающей работу центрального процессора.

Вот таким непосредственным образом богатство средств сопряжения компьютера с внешними устройствами и определяет широту его применения.

Компьютер в сумочке

Сшить сумочку, в которую уместится бы любой компьютер, видимо, можно, хотя и не всегда нужно. Создать компьютер, который уместится бы в любой сумочке, — мечта многих разработчиков электронной техники. Наиболее

полно превратить эту мечту в реальность удалось разработчикам компьютера PARAVANT RHC-88.

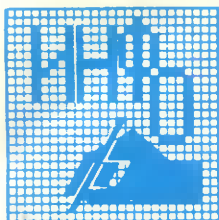
Их детище, имеющее размер книги, умеет делать многое из того, что делает настольный персональный компьютер IBM PC, занимающий значительно больший объем и требующий сетевого питания. Этот карманный компьютер, несмотря на свои размеры, имеет 512К байт оперативной памяти, жидкокристаллический дисплей, на который может быть выведено 16 текстовых строк по 40 символов в каждой и полнозначную клавиатуру. Это позволяет работать с RHC-88 дома, в гостях и, конечно, на работе. А все созданное его владельцем в нестационарных условиях можно легко перенести и использовать на IBM PC, поскольку RHC-88 с ним полностью совместим.

Новости международного рынка

На долю европейских компаний в 1987 г. приходилось лишь 10 % мирового производства полупроводников — неутешительный результат. Поправить положение европейцы рассчитывают с помощью 5-летней программы MEGA Project, к которой в 1984 г. приступили западногерманская фирма Siemens и нидерландская Philips.

Цель этой программы — разработка конструкции и технологии производства высококачественных микросхем памяти: быстродействующих на 1М бит и более медленных, но и более дешевых на 4М бит. Быть может, вложенные в нее 6 млрд. западногерманских марок позволят несколько потеснить Японию на рынке запасающих устройств.

Наукоемкими товарами стремится торговать и Индия. Условий для производства микросхем там нет и даже компьютеров пока не хватает, однако индийцы не отчаиваются. Они всерьез занялись созданием программного обеспечения, и вот результат: в 1986 г. доходы от экспорта программ составили 27 млн. долларов, в 1987 г. — 57,5 млн., а в 1988 г. — уже 133 млн. Прекрасный результат, особенно если учесть, что программы продаются преимущественно в Англию и США.



Нужен ли школе совместитель?

Очень хотелось бы на страницах вашего журнала продолжить обсуждение темы «Учитель информатики — каким ему быть?». Информатика — это новый и совершенно особый предмет для нашей общеобразовательной школы. Введение его в учебный процесс должно привести к формированию нового образа мышления — мышления человека «информационной» эры. Но особенность его и в том, что отсутствуют единые взгляды на методику преподавания (что очень естественно), нет единого учебника (каждый год — новый), нет хорошей вычислительной техники, а во многих местах ее нет вообще... Все это накладывает на учителя большую ответственность. Ориентироваться в огромном потоке книг, журнальных и газетных публикаций, учебников, везти на себе воз кабинета информатики с постоянно ломающейся техникой, пытаться собрать из разрозненных источников неплохую программную библиотеку может только человек, истинно преданный делу. Итак, кто он, учитель информатики, и каким ему быть?

Поделюсь своими наблюдениями. Во многих школах информатику не считают за основной предмет (хотя ее «вес» никак не уступает «весу» любого другого предмета), поэтому ее «навешивают» на учителей математики и физики. Очень много среди них совершенно «информационно» неграмотных, что сказывается и на учителе, и на ученике. Многим (и тем более женщинам) очень трудно «заботиться» о кабинете, о текущем ремонте ЭВМ. Если же кабинета нет и приходится ездить в УПК или на базовые предприятия, то сталкиваешься с иными трудностями: сегодня одни машины, операционные системы, завтра — другие, а ведь предстоит 6 уроков по своему основному пред-

мету... Я считаю, для учителя, который преподает информатику, главным предметом должна быть именно информатика, а вторым может быть любой другой технический предмет. Может показаться, что в этом случае возникнут сложности в материальном отношении (во многих школах IX—X классов значительно меньше, чем остальных, а информатика пока преподается только для старшеклассников). Но здесь, по-моему, нет проблемы. Во-первых, компьютерными классами школы пока обеспечены недостаточно, и можно было бы преподавать для 2—3 соседних школ в одном кабинете. Кроме того, существуют такие формы, как факультативы для старшеклассников, кружки для младших классов, хозрасчетные компьютерные клубы при школах. Кстати, такой клуб уж точно не под силу учителям, для которых информатика — второй предмет.

Другой аспект проблемы — наличие среди учителей информатики довольно большого числа совместителей (мне думается, что по их числу информатика с большим отрывом опережает другие предметы). Хорошо это или плохо? Это очень важный вопрос, и хотелось бы остановиться на нем подробнее. Как я уже писал, учителю необходимо находиться в центре огромного потока информации по нашему предмету, своевременно его перерабатывать и выдавать ученикам в виде готового продукта, т. е. уроков, наглядных пособий, интересных заданий, новых идей и т. п. Это возможно только при тесном контакте науки и образования, который можно осуществлять либо через сотрудничество научного учреждения и школы, либо усилиями преданных своему делу учителей, либо с помощью совместителей. Эти пути имеют полное право на взаимное сосуществование.

У каждого свои достоинства и недостатки, которые не следует ни преувеличивать, ни преуменьшать. Какой путь эффективнее, должна показать практика, причем в каждом конкретном случае решение может быть свое. Только таким, а не искусственно насаждаемым способом этот вопрос нужно решать.

Что за люди идут в совместители? Прежде всего это специалисты, глубоко понимающие важность проблемы компьютеризации и информатизации нашего общества. Это преданные делу люди, для которых материальная сторона не является определяющей. (Сейчас программистом можно, если захотеть, зарабатывать на порядок больше, чем дает школа.) Немало среди таких людей выпускников тех спецшкол, которые еще не так давно пытались изменить общественное мнение по отношению к информатике, демонстрируя свой опыт, хотя было это очень и очень трудно. Поэтому люди, пришедшие в школу, хорошо знают этот предмет и «изнутри», что очень важно в учебном процессе. Кроме того, свежие программистские новости, доступ к великолепным научным библиотекам, к последним публикациям в зарубежных и отечественных журналах, постоянная работа на ЭВМ различных типов, широкий круг общения по специальности и по смежным дисциплинам (математика, физика) — вот небольшой перечень преимуществ человека, совмещающего научную работу и преподавание. Появление таких людей в школе надо только всемерно поощрять. Информатика пришла в школу именно благодаря развитию науки, и отлучать ее (читай — совместителей) от школы было бы просто безграмот-

но. Немаловажным для современной школы является и тот факт, что подавляющее большинство совместителей — мужчины. Нашим детям очень нужно, чтобы именно они были проводниками новых идей в образовании. Помочь делу должно и постановление Совета Министров СССР о совместительстве. Думаю, что снятие искусственных преград привлечет в школы много талантливых и заинтересованных ученых.

Сам я тоже являюсь совместителем (работаю в школе второй год), причем отношусь к этому с принципиальных позиций. В школах преподают и многие мои знакомые. И нас встретила статья в ИНФО (1988, № 4) о подготовке учителей информатики в Ленинграде, где, как оказалось, существует схема соцсоревнования, по которой большее количество баллов получает район, в котором меньше число совместителей. Нельзя же по этому критерию оценивать работу учителей! Это искусственный зажим одного из путей развития. Каждый конкретный случай надо рассматривать индивидуально, но возводя проблему в ранг политики. Этим можно оттолкнуть людей, и, я думаю, школа от этого много потеряет.

Интересно было бы познакомиться с различными точками зрения на эту проблему, причем не только с мнениями учителей школ и совместителей, но и должностных лиц, от которых зависит развитие преподавания информатики в школах.

А. ФЕДЯНИН,
преподаватель информатики школы № 843
Москвы

Мы верим!

Репортаж Я. Пенькова «Хотелось бы поверить» в «Учительской газете» от 19.01.89 г. осветил ряд проблем, в решении которых пресса могла быть не просто сторонним наблюдателем и критиком, а активным пропагандистом и агитатором новых способов распространения информации и внедрения педагогических программных средств и методов компьютерного обучения.

Трехлетний опыт компьютеризации школ показал, что традиционными методами централизованного управления разработками ППС не удастся достигнуть желаемого результата, т. е. быстрого и массового распространения качественных программных продуктов и устранения параллелизма в работе программистов и преподавателей.

Причины такого положения с распространением ППС правильно освещены в статье А. Н. Козырева «О перестройке в области информатики» в первом номере журнала «В мире персональных компьютеров».

Можно еще раз назвать их: нерешенность вопросов охраны авторских прав и механизма выплаты вознаграждения за право тиражирования программ; монополизм в распределении заказов на разработку программного продукта; отсутствие доступной технической базы для массового тиражирования и информационного обслуживания пользователей.

Организаторы тематической выставки «Компьютеризация образования (средства, методы, передовой опыт)» провели большую работу по сбору и экспертизе педагогиче-

ских программных средств для различной техники, используемой в школах.

Всего было собрано и прошло экспертизу специалистов ведущих организаций около 1000 программных разработок, из которых более 700 было рекомендовано экспертной комиссией к распространению. Издан каталог ППС и организована их демонстрация в период работы выставки на ВДНХ. По существу, это первый опыт демонстрации программных разработок в стране.

Однако вопросы массового тиражирования программных средств при сохранении авторских прав разработчиков устроители выставки, естественно, не могли решить за короткий период работы выставки и передали эту функцию специализированному кооперативу по оказанию информационно-вычислительных услуг «ПОЛИНОМ» (так же как и техническую поддержку учебных классов, установленных в павильоне ВДНХ в период работы выставки).

Согласно договору между Госкомитетом по народному образованию и кооперативом «ПОЛИНОМ», кооператив взял на себя обязательства по тиражированию ППС, прошедших экспертизу ГКНО. При этом авторы программ на основании соглашений с кооперативом имеют право на отчисления от продажи каждой копии программы (так же как и В/О Союзприбор ГКНО, которое организовало экспертизу программных средств и издание каталога).

К настоящему времени более 100 программ и пакетов оформлены для продажи соглашениями с авторами; подготовлен каталог на машинных носителях, продолжается прием заявок на тиражирование.

В Центре компьютерного обучения (Москва, ул. Василевского, д. 9, к. 1) можно ознакомиться с каталогом ППС, получить консультацию, оформить заявки на тиражирование и получить копии ППС, по которым уже оформлены соглашения с авторами. Кооператив получил 200 заявок от пользователей из всех уголков страны и разослал всем авторам проекты соглашений о тиражировании программных средств.

Наиболее серьезными пока остаются проблемы взаимодействия с авторами; затягиваются сроки оформления соглашений, а назначаемая авторами стоимость ППС иногда столь высока, что многим учебным заведениям они будут не по карману.

В то же время ряд коллективов разработчиков уже дал разрешение кооперативу на бесплатное тиражирование программ, что, конечно, может только приветствоваться и пропагандироваться.

Из нерешенных вопросов следует также назвать проблему проведения экспертизы вновь разработанных ППС, так как в настоящее время нет постоянно действующих экспертных комиссий при ГКНО, которые могли бы оперативно оценивать вновь разработанные ППС и давать заключения о возможности их тиражирования.

Специализированные издания должны предоставить свои страницы для регулярного рекламирования вновь разработанных и прошедших экспертизу педагогических программных средств. Это может существенно приблизить сроки осуществления проектов компьютеризации образования, о которых говорилось в репортаже «Хотелось бы поверить».

И. ПОЛЯКОВ

122

Информатика и русский язык

В пробном учебном пособии по курсу ОИВТ под ред. А. П. Ершова и В. М. Монахова, а также в статье В. Исакова «Исполнение алгоритмов» (Информатика и образование. 1987. № 2) приводится алгоритм правописания приставок на «З» («С»). В обоих случаях алгоритм ошибочен (см., например: *Кайдалова А. И., Калинина И. К.* Современная русская орфография. М., Высшая школа. 1983). Правильный алгоритм выглядит так:

алг

нач

если корень слова начинается со звонкой согласной или гласной

то на конце приставки писать «з»

иначе на конце приставки писать «с»

все

кон

В. ПЛАКСИЙ,

доцент Харьковского государственного университета

Семинар на ВДНХ

На ВДНХ СССР в декабре 1988 г. проходил всесоюзный семинар «Передовой опыт разработки и внедрения педагогических программных средств в учебных заведениях системы профтехобразования». В Москву съехались почти 200 преподавателей и мастеров профтехучилищ, методистов учебно-методических кабинетов, работников территориальных органов управления. Участники семинара ознакомились с тематической выставкой «Компьютеризация образования (средства, методы, передовой опыт)».

На семинаре рассмотрен широкий спектр проблем: пути перестройки и концепция развития системы профтехобразования, общеметодические аспекты создания и применения ППС, вопросы преподавания курса информатики.

Многие выступавшие отмечали резкое ухудшение дел с поставками техники. Планы поставок срываются, заводы отказывают поставщикам, а та техника, которую все же удается «добыть» учебным заведениям, в подавляющем большинстве некомплектна и неработоспособна. В этих условиях заслуживает внимания организация работ в вычислительном центре Минобразования УзССР, о котором рассказал М. М. Юсупов. Центр (который раньше находился в ведении Госпрофобра УзССР) взял на себя решение всех вопросов по наладке и обслуживанию вычислительной техники в училищах.

В отношении формы проведения подобных семинаров участники пришли к мнению, что при нынешнем состоянии дел и уровне квалификации разработчиков необходимы либо специализированные семинары специалистов по видам программного обеспечения (инструментальные средства, программные тренажеры, педагогические программные средства и пр.), либо секционная работа в рамках единого семинара.

В целом можно отметить деловую атмосферу семинара, показавшего, что, несмотря на организационные трудности, работа по компьютеризации профтехобразования продолжается.

По инициативе участников семинара была организована рабочая группа, которая подготовила рекомендации семинара. Мы предлагаем их читателям.

Рекомендации

Всесоюзного семинара «Передовой опыт разработки и внедрения педагогических программных средств в учебные заведения системы профтехобразования»

123

Отмечая исключительную важность проблемы создания и внедрения новых информационных технологий в образовании, крайне неудовлетворительное состояние дел с поставками и обслуживанием вычислительной техники в профтехучилищах страны, участники Всесоюзного семинара считают необходимым сосредоточить внимание органов управления всех уровней, научных, методических и производственных организаций профтехобразования на решении следующих вопросов.

1. Рекомендовать создание в территориальных органах управления народным образованием подразделений или постоянно действующих рабочих групп по проблеме внедрения новых информационных технологий в образовании.

2. Прекратить поставку в учебные заведения профтехобразования устаревших средств вычислительной техники, в первую очередь КУВТ-86, как не соответствующего ни педагогическим, ни гигиеническим, ни техническим требованиям. Приложить усилия к выполнению поставок современных КУВТ УКНЦ и «Корвет» в объемах, предусмотренных решениями директивных органов, требуемого качества и надежности.

3. Рекомендовать республиканским органам управления народным образованием сохранить и развить сеть региональных центров, опорных училищ по разработке и внедрению педагогических программных средств в учебный процесс, подчинить их республиканским органам управления народным образованием, обеспечить целевое финансирование, первоочередное оснащение новейшими отечественными и зарубежными средствами вычислительной техники, комплектование штатов. Разработку программных средств в региональных центрах, опорных училищах вести преимущественно на основе госзаказа Госкомитета СССР по народному образованию, используя различные формы организации работ: создание временных научно-творческих коллективов; развитие хозяйственной деятельности с привлечением отдельных разработ-

чиков, авторских коллективов, учреждений и кооперативов.

4. Разработать методики и определить порядок проведения экспертизы программных и технических средств учебного назначения.

5. Организовать переподготовку преподавательского состава и повышение квалификации руководящих кадров и специалистов профтехобразования в области новых информационных технологий в образовании на базе действующей системы повышения квалификации на соответствующем уровне. Для этого оснастить Всесоюзный институт повышения квалификации и его филиалы, республиканские институты повышения квалификации с филиалами достаточным количеством учебной и профессиональной вычислительной техники, начать разработку специализированных пакетов программ для системы повышения квалификации.

6. Организовать разработку и производство действующих моделей производственных объектов, управляемых ПЭВМ; средств коммуникации, сопряжения с аудиовизуальной техникой. Приступить к проектированию вычислительной техники следующего поколения для учебных заведений профтехобразования, разработке соответствующих программных средств.

7. Распространить опыт ГВЦ Минобразования УзССР (бывш. ГВЦ Госпрофобра УзССР) по решению проблем комплексного централизованного обслуживания учебных заведений путем создания хозрасчетных специализированных предприятий по поставке, пуску, наладке, техобслуживанию и ремонту средств ВТ, ведению региональных ОФАП, разработке и экспертизе программных средств учебного назначения, информационно-вычислительному обслуживанию органов управления народным образованием.

Рекомендовать учреждениям народного образования, учебным заведениям профтехобразования участвовать в работе Международного компьютерного клуба (МКК). Одобрить и поддержать инициативу УНО Львовского облисполкома об организации филиала МКК в г. Львове.

8. Продолжить практику проведения подобных семинаров с учетом специализации тематики по видам программного обеспечения (инструментальные средства, автоматизированные обучающие системы, программные тренажеры, педагогические программные средства). Их целесообразно проводить поочередно во всех республиках.

В. СОКОЛОВ,
ведущий методист Гособразования СССР

Международная встреча в Ленинграде

В январе на базе ВНИИ профтехобразования состоялось рабочее совещание и научный семинар с участием делегаций НРБ, ВНР, ГДР, МНР, ПНР, СФРЮ, ЧССР.

На совещании рассмотрены итоги выполнения работ в 1988 г. по проблеме 1.2.7. «Совершенствование системы образования на основе применения средств вычислительной техники» (профтехобразование). Отмечена необходимость изменить характер и направление работ, перейти от теоретических разработок к созданию реальной продукции (педагогических программных средств вычислительной техники, технических средств обучения и тренажеров на базе ЭВМ и микропроцессоров), методических материалов, ориентированных непосредственно на преподавателей. Обсуждая прогноз развития дальнейших исследований и проект Детализированной программы сотрудничества на 1991—1995 гг., стороны пришли к выводу, что в следующей пятилетке самое пристальное, внимание следует уделить проблеме разработки инструментальных программных средств. Они позволят автоматизировать процесс создания компьютерных обучающих программ, расширить круг разработчиков за счет привлечения опытных преподавателей. Подчеркнута перспективность работ по проектированию автоматизированных рабочих мест преподавателей профессионально-технических учебных заведений, мастеров производственного обучения, учащихся, а также важность исследований в области социально-психологических последствий компьютеризации образования.

На семинаре, который проводился по теме «Внедрение информационных технологий в процесс обучения при подготовке квалифицированных рабочих», выступили представители делегаций НРБ, ВНР, ГДР, МНР, ПНР, СССР, СФРЮ. Отметим наиболее интересные выступления.

В докладе Д. Бадарча из Монгольского политехнического института излагалась концепция компьютерной технологии обучения в профтехучилищах МНР. Докладчик отметил, что сейчас в МНР больше 10 центров (первый создан в 1973 г.). Освоено экспериментальное производство 16-рядных компьютеров, совместимых с IBM PC, что формирует базу для компьютеризации образования в стране.

Для профтехучилищ МНР первоочередное значение в плане компьютеризации имеют следующие вопросы: разработка содержания и методики обучения в условиях компьютерной технологии; разработка комплексного методического обеспечения по предмету; создание учебников и методических пособий по различным общетехническим и специальным предметам нового типа с соответствующей программной поддержкой; развитие и внедрение новых форм и методов обучения на базе ЭВМ.

Р. Нойберт из Центрального института профессионального образования ГДР изложила концепцию внедрения информационных технологий в профессиональное образование своей страны. Основу концепции составляют следующие положения: обучение информатике в профессиональном об-

разовании преследует цель содействовать формированию у будущих рабочих умений решать различные задачи, включающие использование современной техники переработки информации, быстро перекалцифицироваться при внедрении новой техники;

изучение информатики должно происходить дифференцированно (от 32 до 72 ч), с учетом конкретной специальности и перспектив использования вычислительной техники в будущей профессиональной деятельности.

Большая роль в концепции отводится использованию компьютеров в процессе обучения для демонстрации и тренинга.

В докладе О. Бака из Центрального института педагогики ВНР подчеркивалось, что ключевой педагогической идеей Венгрии в настоящее время является идея интеграции учебных предметов. Признано целесообразным не изучать основы информатики как отдельный учебный предмет, а формировать знания учащихся в области информатики и вычислительной техники на основе интеграции знаний из разных учебных предметов.

П. Хотомски, директор Политехнического института технического факультета им. М. Пупин из СФРЮ, посвятил доклад проблемам моделирования физических явлений на учебных ЭВМ. Он подчеркнул, что внедрение ЭВМ в учебный процесс дает возможность использовать абстрактные машинные модели вместо проведения реального физического эксперимента. Такой подход удобен при подготовке к реальному эксперименту; как замена дорогостоящего, опасного или сложного эксперимента; для управления экспериментом и повышения качества обработки экспериментальных данных. Машинные модели следует применять в тех случаях, когда модель дает возможность наблюдения и изучения каких-либо внутренних отношений в самом объекте либо отношений объекта к внешнему миру, которые в естественной форме наблюдать трудно или невозможно.

Большая группа докладов была представлена делегацией Народной Республики Болгарии. Среди них отметим два, посвященные новой образовательной технологии МИКРОЛАБ. Доклады были

сделаны сотрудниками Высшего машино-электротехнического института им. В. И. Ленина в Софии В. Василевым и К. Мечковым. Сущность новой образовательной технологии состоит в следующем.

Учебный компьютер подключается к исследуемому объекту (режим *on-line*), при этом в единое целое объединяются процессы получения, обработки, хранения и распечатки результатов лабораторного эксперимента. Технология МИКРОЛАБ требует хорошей предварительной подготовки учащихся и составления индивидуального расписания лабораторных работ для каждого из них, а также обеспечивает индивидуальную работу в ходе выполнения лабораторных упражнений, автоматизацию лабораторного эксперимента, запись результатов эксперимента на индивидуальную дискету или их распечатку на принтере, возможность объективной оценки каждого учащегося по результатам его работы, постоянную гибкую обратную связь с учащимся.

На семинаре были продемонстрированы программные средства, ориентированные на систему профессионально-технического образования. Большой интерес участников вызвало инструментальное программное средство «Свиток», созданное в региональном центре по разработке ППС ВТ СПТУ № 11 Киева под руководством В. Д. Леонского. «Свиток» является тренажером для решения задач по общеобразовательным, общетехническим и специальным предметам, который может использовать каждый преподаватель для обучения учащихся составлению алгоритмов решения задач по различным учебным предметам.

Интересные инструментальные средства на базе графического редактора были продемонстрированы И. Панковым из ВМЭИ (НРБ).

Материалы семинара будут опубликованы во ВНИИ профтехобразования.

Желающие их приобрести могут обращаться по адресу: 191119, Ленинград, ул. Черняховского, 2. ВНИИ профтехобразования.

**В. АНТОНОВА, Н. ГОРЕЛЫШЕВА,
М. ЛЕБЕДЕВА**
ВНИИ профтехобразования

Два слова о съезде

17 февраля 1989 г. в Москве состоялся Учредительный съезд Всесоюзного общества информатики и вычислительной техники (ВОИВТ). Оргбюро съезда под председательством заместителя председателя ГКВТИ И. Н. Букреева были подготовлены проекты Устава Общества и резолюции съезда. Делегаты, прибывшие в Колонный зал Дома союзов, должны были обсудить эти проекты, выбрать членов правления и президиум и определить сроки проведения I съезда.

Задачи Общества во вступительном докладе охарактеризовал И. Н. Букреев: «...создаваемое Общество призвано консолидировать творческую интеллигенцию для решения таких задач, как формирование единых требований пользователей к средствам ВТ, постановка принципиально новых задач разработки СВТ и МПТ, определение рациональ-

ной структуры фондов Государственной системы баз данных и структуры услуг, оказываемых ОГСПД и Государственной системой сервисного обслуживания прежде всего населению. То есть Общество должно выполнить функцию посредника между потребителями и пользователями и Государственным заказчиком (в лице ГКВТИ СССР) и производителями СВТ и информационных служб и систем...

Задачей Общества должны стать не столько аттестация и распространение программного обеспечения, сколько оперативное информирование пользователей (посредством своего печатного органа) о том, кто и где разработал или разрабатывает пригодные для массового тиражирования или продажи технические и программные средства; создание эффективного механизма обмена инфор-

мацией в рамках проводимых различными организациями совместных работ по созданию элементной базы, средств ВТ, аппаратуры связи, математического и программного обеспечения различных систем управления, организация выставок современной отечественной и зарубежной техники и литературы; ознакомление с планами издательства в указанной области, обеспечение членам Общества преимуществ в доступе к литературе, стандартам, образцам техники.

Одной из важнейших функций Общества является организация помощи в обучении массовых пользователей ВТ, обмен опытом работающих в этой области, распространение и внедрение автоматических обучающих курсов (АОК) по различным дисциплинам; организация на общественных началах обучения и переподготовки кадров как у нас в стране, так и за рубежом.

В Обществе могут входить люди разных специальностей и профессионального уровня, от академиков до студентов, которые увлечены данным направлением в науке и технике и готовы посвящать ему свободное время. И вообще, каждый, кто имеет какую-то научную, инженерную идею, не должен быть формально отвергнут...

В Уставе Общества, который всем присутствующим представлен на рассмотрение, сделана попытка объединить все задачи в виде направлений деятельности Общества».

Однако полноценного обсуждения Устава Общества — ключевого документа, — на мой взгляд, так и не состоялось. Альтернативные варианты, представленные делегациями Ленинграда и Казахской ССР, не нашли отражения в принятом документе. И если ленинградцы еще имели возможность выступить со своими предложениями (член-корреспондент АН СССР В. С. Лавров), то вариант Устава, подготовленный делегацией Казахской ССР, так и не был оглашен.

Не получили возможности выступить многие из присутствовавших делегатов. К этому не располагал и регламент съезда: на обсуждение проекта Устава было отведено всего 4 часа. Не удивительно, что поступило немало предложений считать этот

съезд предварительным совещанием. Их суть — избрать временный координационный совет, избрать прямым тайным голосованием его председателя, опубликовать для широкого обсуждения проект Устава с альтернативными вариантами, т. е. основательно подготовиться к проведению настоящего учредительного съезда с избранием на него делегатов демократическим путем. (Один делегат с трибуны заявил, что его никто не выбирал, на что председательствующий заметил: «Вас выбрали тайно от вас».)

В еще более жестком режиме прошла вторая часть съезда. В течение 2 часов были проведены голосование за проекты Устава и резолюции (без изменений и дополнений), выборы членов правления (98 человек), Президиума (18 человек) и председателя Общества (И. Н. Букреев).

Голосование проводилось поименно, открыто и настолько стремительно, что, конечно же, об обсуждении кандидатур не могло быть и речи. У меня сложилось впечатление, что многие просто не знают, за кого голосуют. Не все из предложенных кандидатов в члены правления были заранее оповещены о включении в список. (Делегат, сидевший рядом, услышал свою фамилию с неподдельным изумлением.) А более трети выбранного состава членов правления вообще в этот день отсутствовали...

И все же съезд состоялся и цель достигнута — мы, наконец, имеем Всесоюзное общество информатики и вычислительной техники, 29-е в составе Союза научных и инженерно-технических обществ. Но создавалось оно, на мой взгляд, в обход демократических канонов. И сама процедура проведения съезда — тому подтверждение. Недаром многие делегаты покидали Колонный зал, не дожидаясь результатов голосования.

...А I съезд Общества намечено провести через два года.

Хочется надеяться, что проблемы компьютеризации образования найдут отражение в деятельности Общества.

А. КРАВЦОВА

Медленно и плохо или быстро и хорошо?

Что такое скорость реакции управляющей ЭВМ на поступающие сигналы, хорошо знают те, кто хоть когда-нибудь играл в компьютерные игры, написанные на Бейсике. Азартные игроки порой нажимают на клавиши так быстро, что интерпретатор не успевает «переваривать» поступающие команды, чем вызывает справедливое недовольство партнера. А ведь от таких устройств, как, например, радиолокатор, сигналы поступают значительно чаще и в огромных количествах.

В подобных случаях могут помочь специализированные встраиваемые ЭВМ, реализующие

те функции прибора, которые не могут быть выполнены им самим. Не ухудшая его характеристик, встроенные ЭВМ повышают удобство обращения с прибором и расширяют его функциональные возможности.

Так, одна из американских фирм предлагает цифровой четырехканальный осциллограф с памятью, в которой можно зафиксировать осциллограммы быстропротекающих процессов. Полученный таким образом временной «срезы» динамического процесса оператор может рассмотреть более детально, а впоследствии и обработать временные функции поступивших на входы осциллографа и хранящихся в его памяти графиков тока или напряжения. Для этого встроенная ЭВМ пред-

лагает различные возможности по переработке накопленной информации: сравнение различных кривых между собой и эталонной функцией с вычислением разницы в процентах, определение параметров полученных кривых, их экстраполяция и т. д.

Кроме того, осциллограммы и результаты их обработки могут быть воспроизведены на бумаге с помощью графического принтера. Раньше для фиксации изображения с экрана осциллографа был необходим фотоаппарат с сопутствующим оборудованием для обработки фотоматериалов. Производители осциллографа предлагают исследователям быстропротекающих и периодических процессов распрощаться с этой громоздкой техникой.

Задачи по программированию

Вниманию читателей предлагается книга «Задачи по программированию»* из серии «Библиотечка программиста» (вып. 56). Задачник содержит 1052 задачи, тематика которых весьма разнообразна. Подбор задач осуществлен на основе обобщения практики преподавания программирования и опыта работы авторов в школе и вузе. Ответы и решения отсутствуют, приводятся лишь пояснения к задачам повышенной трудности. Большинство задач не ориентировано на конкретный язык программирования, лишь часть из них посвящена языкам Бейсик и Паскаль. Включены задачи, решение которых предусматривает использование специального оборудования и соответствующих программных средств (задачи по графике и звукогенерации).

Книга состоит из двух глав. Задачи главы I позволяют освоить элементарные приемы программирования (разветвления, простейшие циклы, обработка последовательностей символов, вложенные

* *Абрамов С. А., Гнездилова Г. Г., Капустина Е. Н., Селюн М. И.* Задачи по программированию. М.: Наука, 1988.

циклы, использование процедур, файлы и т. д.). Специально выделены темы «Простейшая целочисленная арифметика», «Простейшие графические построения».

Отработке навыков программирования и обработке информации посвящена глава II «Задачи по темам». В ней представлено достаточное число задач по геометрии, физике, биологии. Много внимания уделено задачам сортировки и перебора. В главу включены темы «Случайные числа», «Тексты», «Календарь», «Криптография», «Графика и движение. Мультипликация», «Звукогенерация», «Игры».

Книга адресована начинающим программистам, студентам вузов, пользователям ЭВМ. В ней учителя и преподаватели средних учебных заведений и вузов смогут подобрать нестандартные задачи для индивидуальных заданий в соответствии с интересами обучаемых. Задачник поможет обогатить набор упражнений и задач, используемых в практике преподавания информатики.

И. АНТИПОВ,
НИИ школ МНО РСФСР

127

Везет физикам...

Эта книга вышла в конце 1988 г. и сразу привлекла наше внимание*. В книге много полезного материала для учителя физики, информатики и для того, кто сочетает преподавание этих предметов. В ней содержится большое количество рисунков, схем, диаграмм, удачных фотографий. Некоторые схемы настолько хорошо проработаны методически, что не нуждаются в пояснениях. Содержание и названия ряда параграфов радуют своей новизной, например: «Ознакомление с элементной базой МК и ЭВМ в системе прикладных вопросов курса физики», «Урок-экскурсия в научную лабораторию,

оснащенную ЭВМ», «Программирование задач, направленных на развитие мышления и творческих способностей учащихся» — из них учителя могут почерпнуть много практических рекомендаций, советов.

В книге приводятся решения задач с конкретным практическим содержанием и структурные схемы алгоритмов их решения, которые можно использовать для организации карточки компьютерно-ориентированных задач курса.

Книга буквально пронизана межпредметными связями. Она содержит множество междисциплинарных сведений справочного характера (правила построения структурных схем алгоритмов; данные по элементной базе, классификации и параметрам МК и ПЭВМ; рекомендации по компоновке, оснащению, оформлению и организации работы кабинетов физики, содержащих комплекс оборудо-

* *Извозчиков В. А., Ревунов А. Д.* Электронно-вычислительная техника на уроках физики в средней школе. М.: Просвещение, 1988. (Б-ка учителя физики).

вания физического лабораторного практикума, ЭВТ, ТСО).

Тот факт, что глава о методике применения микрокалькуляторов в преподавании физики превосходит по объему главу о применении ЭВМ, на наш взгляд, не может быть поставлен авторам в вину, так как отражает реальное состояние материальной базы школьных кабинетов.

В книге встречаются мелкие неточности и опе-

чатки, но, как говорится, «дорога ложка к обеду». Если бы книги подобного типа появились для учителей химии, географии, русского языка и практически всех остальных школьных дисциплин, то дело информатизации народного образования пошло бы чуть-чуть быстрее...

Б. МЕДВЕДЬ, С. ИЛЬИН,
Ленинград

Библиография

1. Фролова Г. В. Педагогические возможности ЭВМ. Опыт. Проблемы. Перспективы. Новосибирск: Наука. Сиб. отд-ние, 1988.

В книге дано развернутое изложение опыта использования ЭВМ при обучении. Представлены подробные сведения об организации преподавания с использованием компьютеров в ряде школ, профтехучилищ и вузов Новосибирска. Анализируется история внедрения вычислительной техники в школу, обсуждаются возможные перспективы развития этого процесса. Отдельная глава посвящена изложению некоторых особенностей компьютеризации в социалистических странах, а также в США. Несмотря на перегруженность книги общими рассуждениями и перечислениями заслуг отдельных специалистов, она будет полезна учащимся школ, профтехучилищ и учителям.

2. Машниц Е. И. Психолого-педагогические проблемы компьютеризации обучения. М.: Педагогика, 1988.

Книга представляет собой одну из первых попыток (с точки зрения психологии) комплексного анализа проблемы компьютеризации обучения. На основе изучения различных отечественных и зарубежных публикаций (обширный список которых приведен в книге), автор изложил свою точку зрения на процессы, происходящие в школе. В книге отмечается односторонность современного понимания роли компьютера в обучении, низкое качество и ненужность большинства создаваемых в настоящее время «обучающих» систем. Одним из возможных выходов из создавшегося положения, по мнению автора, является более серьезное изучение психолого-педагогических аспектов применения компьютеров в школе; разработка принципиально новой системы учебных курсов.

Неконкретность и недостаточная убедительность ряда положений книги несколько не принижают ее полезность. Материал этой работы позволяет

взглянуть на проблему компьютеризации через призму психологии, что может способствовать выработке более ясных представлений о перспективах компьютеризации обучения. Книгу можно рекомендовать для изучения психологам, методистам и учителям.

3. Джехани Н. Язык АДА. М.: Мир, 1988.

Несмотря на то что и школьники, и студенты вряд ли будут в ближайшее время использовать язык Ада, данная книга будет полезна всем желающим больше узнать о современной методологии программирования. Книга содержит множество достаточно понятных и характерных примеров, которые иллюстрируют возможности пакетов и модулей. Одно из основных достоинств языка — возможность организации параллельного выполнения программ; большое внимание в книге уделено средствам управления работой параллельными процессами: «рандсву» и «семафорам». Наряду с изложением концептуальных возможностей языка построения больших систем реального времени во второй части книги приведено справочное руководство по языку Ада.

Изложенный материал будет полезен всем, кто интересуется теоретическими и практическими вопросами современных языков программирования.

4. Язык Пролог в пятом поколении ЭВМ. М.: Мир, 1988.

Книга представляет собой сборник статей, посвященных вопросам использования языка Пролог в компьютерах пятого поколения. Рассмотрены особенности языка и его возможности для создания экспертных систем, баз знаний, использования в системах искусственного интеллекта. Книга поможет старшеклассникам лучше представить себе тенденции и перспективы развития средств вычислительной техники, а учителям даст возможность организовать тематические обзоры.

Сост. Д. ВОЛКОВ

УВАЖАЕМЫЕ ЧИТАТЕЛИ!

*Открыта свободная подписка
на наш журнал
на 1990 г.*

*Индекс журнала «ИНФОРМАТИКА И ОБРАЗОВАНИЕ»
в каталогах «Союзпечати» — 70423.*

Стоимость годовой подписки — 3 руб. 60 коп.



БЮ 11-41

60 коп.
Индекс 70423

OldPC.ru
7002
музей компьютеров



ИНФОРМАТИКА
И ОБРАЗОВАНИЕ

