

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1991





**НАУЧНЫЙ ЦЕНТР
ПРОГРАММНЫХ СРЕДСТВ ОБУЧЕНИЯ
ПРЕДЛАГАЕТ**

ДЛЯ УЧЕБНЫХ КЛАССОВ УК-НЦ

**ПОПУЛЯРНЫЕ
СОВЕТСКИЕ И АМЕРИКАНСКИЕ
ТЕХНОЛОГИИ
ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**MIT-LOGO • WRITER • ВAM • ИнфоМИР • РОБОТЛАНДИЯ
ИНФОРМАТИКА - 91 • МИФ-91 • ИГРУШКА • КВАНТ**

**КАЖДАЯ ПРОГРАММА
ПОСТАВЛЯЕТСЯ В КОМПЛЕКТЕ
С МЕТОДИЧЕСКОЙ ЛИТЕРАТУРОЙ**

ЗАКАЗЫ НАПРАВЛЯТЬ ПО АДРЕСУ:

**109004, Москва, ул. Б. Коммунистическая, 9А
ТЕП. 272-11-25, 272-26-71**

О предлагаемых программах читайте на стр. 24, 57, 125



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

OldPC.su

7 0 0 4

музей компьютеров

Содержание

Общие вопросы

Сергеева Т. Новые информационные технологии и содержание обучения	3
Уваров А. Компьютерные коммуникации	11
Бобыкин В. Информатика в межшкольных УПК	22

Методика обучения

Авербух А. Как от Е-практикума перейти к устройству ЭВМ	25
Марголис Я., Иванов А., Баранкина Э. Содержание и методы непрерывного обучения информатике	33
Гузев В. Работа группами с компьютерной поддержкой	38

Кабинет ВТ

Григорьев С. Обработка списков на Прологе-Д	41
Плетнев Ю. УКНЦ: Бейсик с магнитофона	43
Гляков П., Варова Р. Химические шашки на ЭВМ	44
Тищенко В. Работа с простейшим компьютером	47
Зайцев В., Сухарев П. Интерфейс пользователя для локальной сети УКНЦ	51
Козлов Ю., Лепин Л. Две утилиты для «Ямахи» MSX-2	53
Краснов Ю. Вопросы с однозначным ответом	54
Коломиец Ю. Функции пользователя на «Агате»	56

Клуб БК

Зальцман Ю. Архитектура и ассемблер БК	59
Принципы работы БК-0010	66
Конвертор Бейсик-ассемблер для БК	69
«Электронный диск» для БК-0010	72
Защита программ от несанкционированного доступа	74
Калейдоскоп	75

Педагогический опыт

Забара И., Раков С. Тренажер «DIANA»	81
Ушакова В., Володин Ю., Мардашев Ю. Программа «Допуск»	82

Внеклассная работа

«Уральские» задачи	84
--------------------	----

Соболенко Д., Дудочкин В., Кузькина Т. Международный компьютерный лагерь «Байтик»	90
III Всесоюзная олимпиада по информатике	93
Каймин В. Могут ли программы не содержать ошибок?	95

Точка зрения

Матюшкин-Герке А. Школьная информатика: уроки, проблемы и перспективы	99
Белошапка В. Три учебника информатики	111

ЭВМ в народном хозяйстве

Перцев Д. Банк данных изобразительного искусства в Третьяковской галерее	116
--	-----

Нам пишут

Мы за «Агат»	123
Против помех на УКНЦ	124
«Правец» на уроках математики	124
Ищем единомышленников	125

Информация

Школа-семинар в Ленинграде	126
----------------------------	-----

Книги

Играйте с компьютером!	127
------------------------	-----

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)

Обложка Э. Бажилина
Редактор отдела *А. Кравцова*
Научный редактор *Н. Копытина*
Зав. редакцией *Н. Игнатова*
Художественный редактор *Л. Коновалова*
Корректоры *А. Сорнева, Л. Яковлева*

Сдано в набор 22.11.90. Подписано в печать 8.01.91.
Формат 70×100^{1/16}. Бумага офсетная № 1. Печать офсетная.
Усл. печ. л. 10,40. Усл. кр.-отт. 42,88. Уч.-изд. л. 12,96.
Тираж 54 760 экз. Заказ 2300. Цена 1 руб. 20 коп.

Издательство «Педагогика» Академии педагогических наук СССР
и Государственного комитета СССР по печати.
Почту направлять по адресу: 119034, Москва, Смоленский б-р, д. 4.
Издательство «Педагогика». Журнал «Информатика и образование».
Адрес редакции: Лефортовский пер., д. 8.
Телефон: 261-11-29.

Ордена Трудового Красного Знамени Чеховский полиграфический
комбинат Государственного комитета СССР по печати.
142300, г. Чехов, Московской обл.

Т. СЕРГЕЕВА

Новые информационные технологии и содержание обучения

(на примере предметов естественно-научного цикла)

Проблема содержания обучения и новых информационных технологий (НИТ) рассматривается в данной работе в контексте размышлений о двух моделях обучения — традиционной и альтернативной — лично-ориентированной. В связи с этим рассуждения о содержании обучения невозможны без обращения к теоретическим тезисам общего характера.

Первый тезис. Любая деятельность, в том числе и по внедрению НИТ* в обучение, требует обращения к методологическим средствам. В роли одного из них выступает представление о средстве в его понятийном аспекте. В широком смысле термин «средство» может охватывать все то, что стоит между субъектом и желаемым результатом деятельности. В наших рассуждениях важно следующее утверждение: средство, включенное в ту или иную деятельность, оказывает существенное влияние на нее, если оно (средство) проявляет специфические, присущие только ему функции.

В ситуации, когда средство используется как заместитель (полностью или частично) «чужих» функций, принципиальных изменений в деятельности не

происходит. Так, в учебном процессе педагог сам может быть рассмотрен как средство достижения целей обучения. Для усиления своей деятельности он прибегает к использованию других средств, «овеществляющих» часть его педагогических функций. В частности, включение компьютера в учебный процесс приводит к передаче машине таких функций учителя, как контроль, коррекция, тренинг типовых умений и др. Изменяет ли это учебный процесс и в чем суть этих изменений? Обращение к зарубежному и отечественному опыту свидетельствует о том, что изменения происходят главным образом по двум параметрам: параметру времени (сокращается время на составление контрольных заданий, их тиражирование и предъявление учащимся, обработку результатов контроля и доведение их до каждого ученика и др.) и параметру «охват учащихся» (в ситуации массового обучения, когда на одного педагога в среднем приходится по 35—40 учеников, главная сложность связана с организацией и проведением индивидуального тренинга по обработке важнейших предметных и общеучебных умений и навыков у каждого из них). Компьютер спасает положение там, где оказывается в затруднении учитель — в удовлетворении индивидуального запроса каждого ученика. Но эти изменения относительно целей обучения не носят принципиального характера.

* Под НИТ в данной работе будем понимать совокупность внедряемых в систему образования принципиально новых методов и средств обработки данных, передачи, хранения и отображения информационного продукта с наименьшими затратами и в соответствии с закономерностями учебно-воспитательного процесса.

Второй тезис. Введение компьютера в традиционную модель обучения не приводит к ее «переналадке» ни в ценностном, ни в технологическом, ни в результативном аспектах, поскольку компьютер при этом реализует не свои специфические функции, а функции педагога (контроль или репетиторство) или учащегося, таким образом, в лучшем случае функции машины суживаются, в худшем — деформируются.

4 Более того, можно предположить, что внедрение НИТ в традиционную модель обучения усугубит ее негативные стороны. В частности, еще более заформализует учебный процесс, уменьшив степень общения в школьном коллективе, а если говорить шире — усилит социальную изоляцию, вместе с тем усложнит деятельность педагога, превратив его не в консультанта-помощника, не в организатора учебной коммуникации, а в «многостаночника», разрывающегося между учениками и компьютерами. Внедрение НИТ в существующую модель учебного процесса приведет к возможному обеднению форм предметной деятельности.

Изложенные положения позволяют сформулировать гипотезу: новые информационные технологии окажут принципиальное (желаемое) воздействие на процесс обучения в том случае, если они будут включены в новую модель обучения, а их средства — реализовывать свои специфические функции**.

Адекватной основой проектирования новых информационных технологий обучения, на наш взгляд, выступает психологический принцип деятельности и педагогический принцип «выращивания».

Согласно первому принципу, развитие учащегося основывается на активном присвоении им с помощью учителя общественно-исторических способов деятельности или средств общения. Обучение при этом выступает как организация условий присвоения учащимися тех

или иных форм общения и деятельности [1].

Согласно второму принципу, признается двойственный характер педагогического воздействия [4]. С одной стороны, реализуя социальный заказ, педагог управляет становлением личности, с другой — управление осуществляется на основе сознательного учета педагогом индивидуальных качеств учеников. «Выращивание» личности обучаемого происходит в условиях организации самоопределения последнего, при максимальном осознании характера усваиваемой деятельности (только в этом случае он считает ее «своей»). Когда изменение учащегося в целом остается субъективно самоизменением, педагог может лишь способствовать желаемому изменению, создавая через общение с ним «естественные» условия. При этом знания «отдаются» ученику под сформированную в процессе предыдущего учебного общения потребность.

Указанные выше принципы наиболее адекватно и полно, на наш взгляд, отражены в личностно-ориентированной модели обучения. Цель ее — содействовать развитию обучаемого как личности, формировать у него потребности в самообразовании и самоопределении в учебных и жизненных ситуациях с осознанием личной ответственности. Знания, умения и навыки в этой модели рассматриваются не как цель, а как средство развития личности обучаемого. Характер взаимодействия участников учебного процесса отражается в принципах педагогики сотрудничества [6]. К числу основополагающих относятся: демократичность, открытость, альтернативность, диалогичность (полилогичность как форма демократичности), рефлексивность (осознание целей, содержаний, способов деятельности и характера взаимодействия). В таком общении достигается понимание и признание личности ученика, основанное на желании учителя встать на место ученика. Лозунг учителя: «Я не над вами, я вместе с вами!» Обучение строится по формуле: деятельность — рефлексия — теоретическое знание. Вначале учащиеся погружаются в деятельность, где они выступают в роли

** К специфическим функциям компьютера относятся его уникальные возможности управляемого в интерактивном режиме моделирования и конструирования, а также информационная функция, реализуемая через различного характера базы данных и экспертные системы.

ее активного субъекта, а педагог — в роли организатора коммуникации, в ходе которой создаются условия для присвоения учениками предметных содержаний, умений и способов общения. Одно из важнейших условий, которые моделирует педагог, — создание для учащихся затруднений в осуществляемой деятельности (либо в предметном содержании, либо в способе его усвоения). Возникающая потребность в преодолении затруднений выводит его (вначале с помощью педагога, а затем самостоятельно) в рефлексию, где осуществляется анализ деятельности до затруднения, затем поиск причин возникшего затруднения, проблематизация прошлой деятельности и изменение нормы деятельности как депроблематизация. Последнее уже относится к третьей части формулы.

Учащийся в данной модели выступает в качестве субъекта деятельности (в традиционной — объект деятельности).

Результаты реализации данной модели заключаются в следующем: у учеников появляются устойчивый интерес к учебе и познавательные мотивы, формируются потребности в самообучении, саморазвитии, умение самоопределяться в учебной деятельности с осознанием личной ответственности в ней, потребности в коллективной работе, нацеленной на получение общего результата, и т. д.; у педагога меняется позиция, он становится носителем нового педагогического мышления и принципов педагогики сотрудничества, рефлектирующим профессионалом, способным к проектированию и перепроектированию (в зависимости от потребности учебного процесса и каждого отдельного ученика) своей деятельности в соответствии с указанными принципами.

Итак, рассмотренные тезисы могут быть сфокусированы в следующем — новые информационные технологии целесообразно разрабатывать в рамках личностно-ориентированной модели обучения; в этом случае компьютер, как центральное средство этой технологии, сможет проявить свои специфические свойства и тем самым принципиально (по целевому основанию) преобразовать деятельность, в которую он включается.

Обратимся теперь к проблеме содержания обучения в условиях НИТ, преобразив это взглядом на цели обучения.

Доминирующая тенденция традиционного обучения состоит в том, чтобы рассматривать предметы естественнонаучного цикла (ЕНЦ) как дающие сумму знаний, умений и навыков, необходимых для практической жизни, для ориентации в мире физических, химических и других объектов, явлений и процессов. Таким образом, в качестве основной цели преподавания дисциплин ЕНЦ выступает приспособление к реальной жизни, а не саморазвитие личности учащегося. Следствием реализации такой цели является факт, что учащиеся воспринимают и физику, и химию, и математику, как некие внешние по отношению к нему области действительности, подчиненные объективным законам. Чаще всего знание самих этих законов, закономерностей, теорий и т. д. остается отчужденным от ученика, не пропущенным через его личностное, индивидуальное восприятие. Кроме того, естественные науки развиваются настолько быстро, что, например, материал, составляющий содержание химических наук несколько десятилетий назад, сейчас уже считается устаревшим. А именно такой материал все еще остается основным содержанием школьных дисциплин ЕНЦ. Поэтому естественнонаучное образование существенно отстает от развития самой науки. Преодоление этого кризиса возможно при смене целей: от освоения знаний, умений и навыков в вещной форме, т. е. в форме научно-теоретического содержания науки, к цели развития самого учащегося как личности, его способностей, его творческого потенциала. И тогда ученик «сможет овладеть методом мышления современной физики как частным случаем вещного мышления, подлежащим развитию и преобразованию в зависимости от потребностей познания и действия» [5, с. 117].

Указанный взгляд на цели требует и соответствующего отношения к содержанию обучения, а соответственно — и средствам НИТ. Остановимся на этом более детально. Сегодня содержание обучения дисциплин ЕНЦ (фактически, а недекларативно) оказывается безраз-

личным к общественному процессу получения этого содержания, к формированию развивающейся личности. Содержание раскрывается в основном в информационной форме — в форме готового результата. Другими словами, современное содержание представляет собой формализованную систему знаний, оторванных от коллективного процесса их получения, т. е. систему, не ориентированную на отражение социальных отношений в составе этого знания. Цели развития личности обучаемого с объективностью требуют иного подхода к отбору содержания обучения. Речь идет о том, что в содержании предметов ЕНЦ (как, впрочем, и всех других) должно выделяться не усвоение информации, фактов естественнонаучной действительности, не заучивание математических, физических и химических формул, а сам поиск, процесс формирования знания, правил, формул и т. д. Это вовсе не означает, что в содержании обучения не должна входить система знаний, умений и навыков, просто этого не достаточно для развития обучаемого. Знания в этой системе должны быть построены так, чтобы он видел историю происхождения этого знания, и не просто в хронологической последовательности, а как живой механизм разрешения противоречий при возникновении нового его элемента.

Итак, ведущий лозунг данного раздела: содержание обучения предметов ЕНЦ должно быть не объектом, а средством развития обучаемого.

Какие компоненты содержания обучения с объективностью требуют реализации НИТ? — основной вопрос дальнейших размышлений. Чтобы сделать переход к «компьютерному вопросу», обратимся ко всему циклу дисциплин естественно-научного характера и проследим, что общее характерно для них. Исходной общностью предметов ЕНЦ является общность целей — развитие учащегося через формирование научного мировоззрения, знаний о живой и неживой природе, методах ее познания и нравственной позиции по отношению к природе. Другая общность состоит в том, что объекты изучения являют собой сложно организованные системы и представлены в иерархии форм организации вещества:

от субатомного уровня организации до макроформ (атомное ядро, ион, молекула, макромолекула, полимер, кристалл, горная порода и др.); объекты изучения дисциплин рассматриваемого цикла лежат не только в области «видимого» материального мира природы, но и за его пределами — микроуровень организации вещества (атом, ион и т. д.) и макроуровень (планета, звезда, галактика). Другими словами, предметы ЕНЦ отличаются той общностью, что все они изучают вещества и их превращения, отличие же состоит в том, что каждая из дисциплин: физика, химия, астрономия и другие — изучает определенные формы организации вещества и особые формы их превращений.

Третьей объединяющей особенностью является то, что ведущими методами познания их объектов и явлений выступают эксперимент и моделирование.

Выявляя общность в содержании предметов ЕНЦ, мы обратились и к общим тенденциям их развития. Было зафиксировано, во-первых, явное стремление к интеграции естественнонаучного знания; во-вторых, представление объектов изучения как систем, аналогичное взгляду в научных областях знания (химии, биологии и т. д.); в-третьих, усиление связи между теорией и жизнью, объединение проблем науки, техники, общества и природы; в-четвертых, что особенно важно в свете наших предшествующих рассуждений, — это гуманизация содержания предметов ЕНЦ, нацеленная на развитие личности обучаемого.

Обратимся к структурным компонентам содержания образования и с учетом проведенного выше анализа выявим, какое содержание требует реализации НИТ.

Первым компонентом содержания обучения выступает система знаний. Здесь прежде всего требуют «компьютерного вмешательства» циклы интегрального характера, которые одновременно могут вводить учеников в предметный мир ряда дисциплин физики, химии, биологии, математики и других. Примерами таких программ уже сегодня могут стать экологические компьютерные программы, «проживая жизнь» в которой обучаемый сохраняет в равнове-

сии природные ресурсы, чистоту водоема, осуществляет математические расчеты физических процессов, необходимых для стабилизации экологических процессов и т. д.

Вместе с тем существующие средства обучения далеко не всегда позволяют ввести обучаемых в микромиры предметов ЕНЦ. Смоделировать сущность цепной реакции, реакции гидролиза, сущность химического равновесия и т. д., а также дать возможность обучаемым «потрогать» в этом микромире происходящие процессы и явления, изменяя их параметры и характеристики, возможно сегодня только с помощью компьютера.

Компьютер должен внести свой вклад и в формирование таких понятий, законов и теорий в курсе дисциплин ЕНЦ, которые в традиционном обучении не находят достаточного опытного обоснования, например изучение механизмов различных реакций в химии или среды с нулевой гравитацией в физике; формирование таких понятий математики, как бесконечность, неограниченное возрастание (убывание) функции, неограниченное приближение ее к какой-либо асимптоте, параллельность прямых и др.

Вместе с тем объективно необходимо новое полифункциональное средство для формирования понятий, опирающихся на наглядные образы, а в традиционном обучении адекватных средств для этого нет. Примером из физики и химии служит понятие «электрон», объединяющее в себе диалектическую двойственность частицы и волны; механизм электрохимической и химической коррозии, механизм взаимного влияния атомов в молекулах и др. Примерами из математики служат интерпретация числа как точки на числовой прямой; предел числовой последовательности и т. д. «Видимыми» могут стать такие свойства функций, как монотонность, периодичность, ограниченность, асимптотическое приближение. «Зримыми» становятся алгебраические формулы, преобразующиеся с помощью компьютера в графики, причем учащийся становится не только созерцателем таких преобразований, но и активным субъектом деятельности. Так

в программе «Графики функций»*** ученик может найти участки возрастания и убывания функции (перемещая графический курсор по кривой), найти экстремумы функции, определить ее поведение и вычислить значение в любой точке.

Неоценимую помощь может оказать компьютер и при формировании понятий теории вероятности в математике.

Вторым компонентом содержания обучения выступают репродуктивные умения (как специфически предметного, так и общеучебного характера), которыми должен овладеть учащийся в процессе обучения.

Обратимся к первой группе умений. Привлечение компьютера необходимо; во-первых, в ситуациях, связанных с вычислениями: это сокращает время, затрачиваемое учениками на осуществление расчетов, их проверку и обработку результатов; во-вторых, при отработке типовых умений. К ним можно отнести перечень типовых умений по химии (составление формул по валентности, классификация сложных веществ, распределение электронов в атомах химических элементов, составление химических уравнений различных типов, составление изомеров по углеродному скелету, составление изомеров аминокислот и т. д.), типовых умений по физике (умение применять положения молекулярно-кинетической теории для объяснения диффузии в газах и жидкостях, различия между агрегатными состояниями вещества, давления газа, закона Паскаля; определение цены деления измерительных приборов, решение расчетных типовых задач и др.).

Компьютер выступает объективно необходимым и незаменимым средством формирования ряда общеучебных умений, в частности общелогических (систематизации и классификации, анализа и синтеза), рефлексивных (умений планировать эксперимент, обрабатывать экспериментальные данные, осуществлять сбор, упорядочение и анализ информации). Например, на экране

*** Программа разработана в лаборатории информатики НИИ СиМО АПН СССР, авторы Я. Кобринский и Н. Копытов.

компьютера можно моделировать ситуации сравнения и аналогии: видеть аналогию движения жидкости в трубах и электрического тока в проводниках и др.

Третьим компонентом содержания обучения являются умения творческого типа, овладевая которыми обучаемый получает субъективно новое знание путем самостоятельного поиска. Как известно, главным признаком проявления творчества является новизна полученного продукта (в учебном процессе результатом творческой деятельности учащегося является субъективно новый продукт). Непременное условие творческой деятельности — наличие затруднений в ходе познавательного процесса. Таким образом, формирование творческих умений требует специально сформулированных учебных проблем, специально организованной познавательной деятельности. Компьютер может служить эффективным средством формирования творческих умений учащихся. В частности, открываются возможности для решения задач нового типа, называемых оптимизационными. Речь идет о задачах, в которых из ряда возможных вариантов выбирается один — наиболее рациональный с определенной точки зрения. Это могут быть задачи на выбор самого экономичного решения (в задачах по математике или физике); на выбор наиболее оптимального варианта протекания того или иного химического процесса (при производстве аммиака, серной или азотной кислот и др.); на выбор оптимальных условий выращивания растений определенного вида в биологии и т. д. Причем компьютер позволяет находить оптимальное решение не только математически, но и графически. В то же время незаменимы возможности компьютера при постановке и решении задач на проверку возможных последствий выдвигаемых гипотез.

Принципиальные возможности для развития творческих умений, в частности конструктивно-комбинаторных, открывают специальные обучающие среды, так называемые предметные среды или микромиры. Примерами подобных «сред» являются известная система Пейперта ЛОГО по информатике, система

ГИПОТЕЗЕР Шварца и ПОСТРОИТЕЛЬ Б. Сендова по математике (планиметрии). В этих системах целенаправленно продуцируются условия, благоприятные для спонтанного возникновения интеллектуальных моделей в мышлении учащегося, когда по мере «прорастания» одних другие приобретают логическую и эмоциональную форму. Главное назначение таких «предметных сред» («страны математики», «страны физики», «страны химии» и т. д.) — побуждать обучаемого активно участвовать в процессе конструирования «объектов, с помощью которых можно думать». Это совершенно особые объекты, в которых сведены в единое целое признаки культурной среды, встроенное «знание» и возможность персональной идентификации в процессе обучения. Последнее означает, что, задавая поведение объекта и размышляя над ним, ученик осмысливает свои собственные шаги и намерения, и наоборот — осознание своего поведения выступает для него как источник содержательных идей для общения (например, с Черепахой в системе ЛОГО) и как основание для выбора адекватных управляющих воздействий. Общение с компьютером в аналогичных программах требует пространственно-временных средств коммуникации и способствует их развитию у учащихся. Эти «умственные упражнения» позволяют ученикам осваивать особый метаязык, на котором можно рассуждать о формах, их постоянном изменении, о скоростях, процедурах и т. д. Иными словами, работая с подобными «предметными средами», учащиеся осваивают формальный язык и одновременно с этим конструируют новый образ понимания самих себя, в котором естественно слиты рациональные, интуитивные, чувственные и эмоциональные компоненты. В рассматриваемых программах спроектированы логика построения объекта усвоения, его «жизнь» и становление, а становление, в свою очередь, дано через действия педагога с учениками — все это не только расширяет познавательную свободу ученика, но создает условия для его развития, а в целом реализует идею «направляемого обучения через открытие». В качестве при-

мера аналогичной «химической среды» может выступить проект программы «Химический конструктор», с помощью которого на экране дисплея учащиеся могут конструировать различного типа химические структуры, начиная от простейших атомов (типа водорода) и кончая сложными молекулами органических соединений.

Значительный вклад в развитие творческой компоненты обучаемого, как известно, вносит правильно организованный процесс формирования действия моделирования. Компьютер открывает для этого большие возможности. Компьютер может моделировать какой-либо процесс или последовательность событий. Это позволяет ученику делать самостоятельные выводы по поводу факторов, оказывающих влияние на протекание процесса или события. Лабораторный эксперимент, требующий для своего проведения недоступных для школы приборов или чрезвычайно длительного (и наоборот — мгновенного) времени, может быть реализован также с помощью компьютера. Так, компьютерная имитация химических процессов предоставляет возможность изучения эффектов, к которым приводит, например, изменение концентрации взаимодействующих веществ или температуры процесса. Имитационное моделирование позволяет варьировать скорость течения процесса, например замедлять быстрые реакции с целью их обстоятельного анализа. Подобного эффекта можно достичь с изменением скорости киносъемки, однако компьютерное моделирование позволяет ученикам самостоятельно управлять процессами. Аналогично могут быть ускорены длительные события, например выведение особей мух дрозофил в курсе общей биологии путем скрещивания пар с различным набором хромосом; с помощью компьютера этот многодневный процесс может быть проведен за 1 ч. При осуществлении этой работы ученик может принимать решения и проверять их последствия. Большое преимущество компьютерного моделирования состоит также в возможности неоднократного повторения имитации, часто через короткие временные интервалы, повторения до тех пор, пока не будет

достигнут желаемый, с точки зрения обучаемого, результат. Это дает возможность обучаемым глубже усвоить механизм действия и взаимодействия различных факторов, а также приобрести и осмыслить представления о достигнутом.

Возможности компьютера позволяют включать в имитируемый процесс «случайные события», поскольку алгоритм работает на основе математической модели ситуации. Так, при моделировании химического производства (производство стали, чугуна и т. д.) может быть запрограммировано неожиданное падение мощности, отражающееся в потере тепла; при моделировании экологического процесса — неожиданное появление большого количества хищников в изучаемом регионе и т. д. Такие моделируемые ситуации оказываются проблемными для обучаемых, и поиск выхода из возникшего затруднения является творческим поиском.

В заключение рассуждений по проблеме содержания обучения и НИТ обратим внимание на факт, что современное содержание учебных предметов перегружено многочисленной справочной информацией, касающейся объектно-содержательной, а не субъектно-содержательной стороны дела. Следовательно, «знание, которое отчуждено и может быть передано, не должно загружать индивиду. Оно должно загружать соответствующим образом организованные склады информации» [8, с. 117]. Сегодня одним из наиболее совершенных средств хранения информации является компьютер. Упорядоченное хранение информации в базах данных позволяет отыскать необходимую информацию по заданным признакам. Системы такого рода очень удобны, так как имеют большую скорость оперирования данными по сравнению с немашинными способами и позволяют сконцентрировать внимание на творческой работе.

Предметы естественнонаучного цикла требуют особой информации, составляющей как бы фон их изучения. В химии, например, это данные о химических элементах и их соединениях, параметрах протекания тех или иных процессов, составе природного сырья для получения того или другого чистого ве-

щества и т. д. Справочники обычно содержат значительно больше информации, чем это нужно учащимся; они быстро устаревают, а оперативное внесение в них обновленных данных весьма затруднительно — требуется переиздание справочника. Кроме того, работа с подобными изданиями оказывается трудоемкой и длительной. Базы данных по сравнению со справочной литературой обладают рядом преимуществ: они необходимым образом структурированы, а также содержат определенный объем информации. Результаты поиска отображаются на экране или распечатываются принтером. Еще одно преимущество состоит в отсутствии необходимости полной интеграции данных и средств доступа к ним. С помощью соответствующего пакета программ можно получить определенный статистический анализ данных. Например, можно внести в компьютер информацию о свойствах химических элементов периодической системы Д. И. Менделеева и получить от машины сведения о металлах или неметаллах, лантаноидах или актиноидах и др. Компьютер может также произвести сравнение разных групп химических элементов по одним и тем же параметрам, допустим по величине их электроотрицательности и т. д.

Четвертым компонентом содержания обучения выступает личностная компонента. Формирование личностных качеств невозможно без развития у ученика способности видеть другого человека (ученика, педагога и пр.). Личностно-ориентированное содержание способствует развитию нравственности подрастающего человека, призванной стать основным критерием его настоящей и будущей жизни. Наиболее убедительным примером того, что в это дело свой существенный вклад могут и должны внести НИТ, являются проблемы окружающей среды (факторы замедленного действия — загрязнение или кислотные дожди; внезапные катастрофы — разлив нефти на поверхности моря или крупная авария на химическом производстве). Моделирование с помощью компьютера создает неограниченные возможности нравственного вос-

питания обучаемых, в частности через решение указанных выше проблем. Компьютерное моделирование позволяет анализировать возможные последствия тех или иных аварий, последствия применения тех или иных технологий. Можно высказать предположение, что тщательная и правильно организованная работа с компьютерными моделями позволит не только научить учащихся избегать в будущем подобных опасностей, но и воспитать нравственные оценки их возникновения в современном технократическом мире. Работа с компьютерными моделями позволит сформировать у школьников чувство ответственности по отношению к другим людям через отношение к окружающей среде (экономное расходование сырья и энергии, предотвращение ущерба, наносимого окружающей среде в результате неквалифицированных действий); чувство ответственности по отношению к себе, собственному организму (гигиена питания, профилактика заболеваний и др.).

Таков наш взгляд на актуальную проблему НИТ и содержания обучения.

Литература

1. Концепция применения компьютеров в учебном процессе: Психологический аспект / Под ред. В. В. Рубцова. НИИ ОиПП, раб. материалы.
2. *Невуева Л. Ю., Сергеева Т. А.* О перспективных тенденциях развития педагогических программных средств // Информатика и образование. 1990. № 5. С. 5—10.
3. Рекомендации по проектированию педагогических программных средств (психолого-педагогический аспект) / Сост. Т. А. Сергеева, Л. Ю. Невуева // Материалы III совещания рабочей группы стран СЭВ по методологии разработки педагогических программных средств. София, 1988.
4. Учебная и педагогическая деятельность в активных формах обучения / О. С. Анисимов. М.: ВВШУ АПН РСФСР, 1989.
5. Философско-психологические проблемы развития образования / Под ред. В. В. Давыдова. М.: Педагогика, 1981.
6. *Черняевская А. Г., Комраков Е. С., Сергеева Т. А., Сиденко А. С.* Компьютерные технологии в педагогике сотрудничества // Психолого-педагогические вопросы использования компьютеров в средней школе. М., 1989.

Компьютерные коммуникации

Телекоммуникация (от греч. tele — вдаль, далеко и лат. communicatio — связь) — это обмен информацией на расстоянии. Радиопередатчик, телефон, телетайп, факсимильный аппарат, телекс и телеграф — наиболее распространенные и привычные нам сегодня примеры технических средств телекоммуникации.

В прошедшем десятилетии к ним добавилось еще одно средство. Это компьютерная коммуникация, которая получает сейчас все более широкое распространение. Она обещает потеснить факсимильную и телетайпную связь, подобно тому как они вытесняют сегодня телеграф.

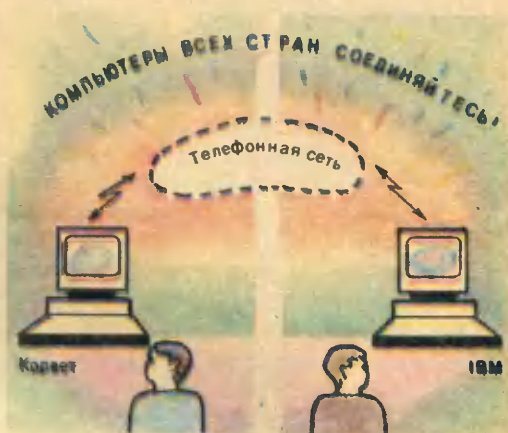
Для пользования компьютерной коммуникацией нужны компьютер с соответствующей программой, модем и линия связи, позволяющая вашему компьютеру соединиться с другой компьютерной системой. Как и все участники проекта «Школьная электронная почта» (ШЭП), вы можете использовать персональный компьютер типа IBM PC, коммуникационную программу «Procom», современный высокоскоростной модем и линию городской коммутируемой телефонной сети.

При использовании компьютерной коммуникации абоненты, как правило, пользуются данными, хранящимися на центральном компьютере, обмениваются сообщениями (текстами), подготовленными с помощью ЭВМ. Все более распространенным становится новый вид обмена сообщениями — электронные письма.

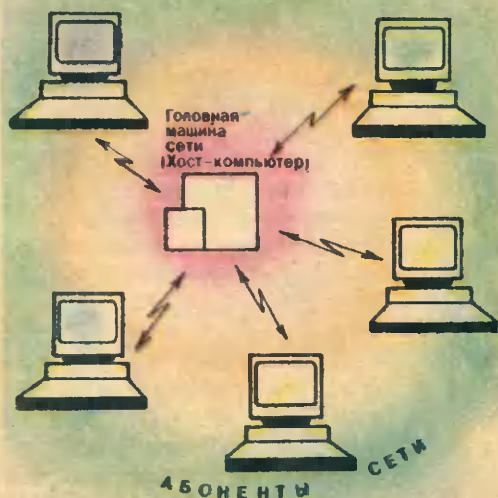
Компьютерные сети

Во многих школах компьютеры в кабинете информатики объединены кабелем в локальную сеть. Это позволяет пересылать данные между ЭВМ, находящимися в учебном классе. Здесь мы будем говорить о сетях, использующих коммутируемые телефонные каналы связи.

В простейшем случае компьютерная

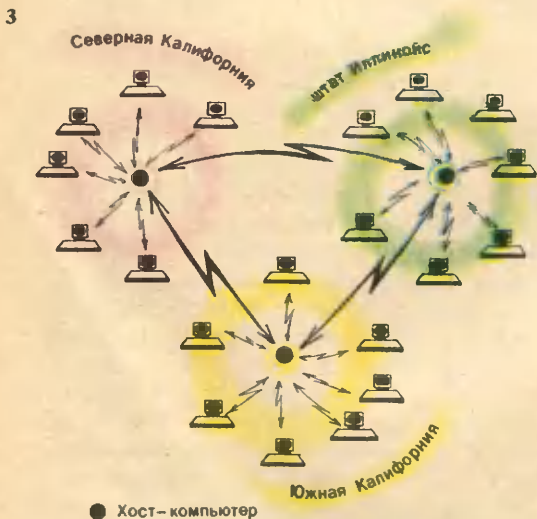


сеть состоит из двух машин. Любые две школы, участвующие в проекте ШЭП, могут сами создать такую, связаться друг с другом (рис. 1). Однако пользоваться такими прямыми связями не очень удобно. Использовать ЭВМ только в качестве устройства связи и держать ее постоянно подключенной к телефонной сети довольно расточительно. Поэтому наибольшее распространение получили сети, включающие много компьютеров, которые по мере необходимости подключаются к центральной ЭВМ. Центральная ЭВМ, через которую связываются



между собой все абоненты (участники, пользователи) сети, называется хост-компьютером (от англ. host — хозяин) или головной машиной сети. Пример такой сети типа «Звезда» приведен на рис. 2.

Хост-машины, в свою очередь, можно связать между собой. В результате образуется разветвленная коммуникационная система, состоящая из нескольких связанных между собой узлов. Каждый узел обслуживает работающих в нем пользователей и обеспечивает обмен информацией с другими узлами. Пример связи узлов в широко известной учебной сети FredMail (США) показан на рис. 3.



Хост-машины и линии связи чаще всего принадлежат коммерческим фирмам, предоставляющим коммуникационные услуги своим абонентам за плату. Обычно абонент платит регистрационный взнос, ежемесячную абонентную плату, а также оплачивает время работы в сети и объем переданной информации. Например, абоненты сети Совам-Телепорт SFMT платят 15 руб. ежемесячной абонентной платы, 45 коп. за каждую минуту работы в сети и 65 коп. за каждый килобайт переданной информации. Высокие цены заставляют экономить. Наиболее естественный путь — разумное сокращение времени пребывания в сети. Отвлечение оператора на посторонние занятия во время сеанса связи

с хост-компьютером может стоить довольно дорого.

Некоторые компьютерные сети принадлежат ассоциации пользователей. Наиболее известный пример — сеть «BITNET», которая объединяет сотни высших учебных заведений в США и других странах мира. Создателями этой системы являются университеты, которые покрывают все расходы на ее эксплуатацию и развитие. Любые сотрудники и студенты этих университетов могут пользоваться сетью «BITNET» бесплатно.

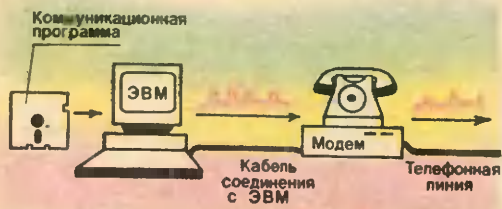
В последние годы появилось несколько сетей, ориентированных на нужды общеобразовательной школы. Одна из наиболее известных — «KIDNET». Эта сеть создана Национальным географическим обществом США и предлагает школам занятия по географии (проекты: Кислотные дожди, Погода и др.). Все школы г. Нью-Йорка могут использовать сеть «NYCENET», созданную и финансируемую городским отделом народного образования. В различных странах мира создателями компьютерных сетей, ориентированных на работу с общеобразовательными школами, выступают телефонные компании (AT&T), производители вычислительной техники Apple и IBM, исследовательские и государственные учреждения.

В Москве в 1989 г. начала действовать Московская школьная телекоммуникационная сеть MoSTNet, созданная Научным советом АН СССР по комплексной проблеме «Кибернетика» при поддержке городского комитета по народному образованию.

Компоненты компьютерной коммуникационной системы

Вне зависимости от того, кем и как организована сеть, все абоненты используют на своих рабочих местах компьютерные коммуникационные системы, состоящие, как правило, из одних и тех же компонентов (рис. 4).

Терминал. Обычно это персональная ЭВМ, используемая для получения и отправки корреспонденции. IBM PC, которые скоро появятся в вашей школе, — мощная многофункциональная вычис-

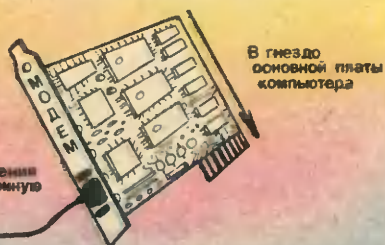


лительная система. На ней легко совместить работу в сети, подготовку электронных посланий, их хранение на машинных носителях, корректировку, автоматическую обработку поступающих материалов, распечатку их на бумаге и многое другое.

Модем. Для того чтобы соединить персональный компьютер с телефонной сетью, необходимо специальное устройство, согласующее их работу. Таким устройством является модем (сокращение от слов модулятор/демодулятор). Модем переводит двоичные сигналы, используемые ЭВМ, в аналоговые, которые обычны для существующих телефонных линий (работают как модулятор), и наоборот (работает как демодулятор).

Обычно модем выполняется в виде отдельного устройства, которое подключается к ЭВМ через последовательный интерфейс (рис. 5, а). В последние годы все чаще используются встроенные модемы. Встроенные модемы выпол-

5а



5б

няются в виде электронной платы, устанавливаемой в компьютере (рис. 5, б).

Существует много марок модемов. Их производят и компьютерные фирмы, и фирмы, специализирующиеся на выпуске коммуникационного оборудования. В школах используются общедоступные дешевые модемы, выпускаемые сегодня в мире десятками и сотнями тысяч. Все эти модемы поддерживают принятые в международной практике стандартные протоколы связи (о протоколах связи см. ниже).

В последнее время у нас начали появляться модемы «оригинальной» конструкции, не поддерживающие стандартных протоколов связи. Такие модемы не совместимы с большинством стандартных сетей, и к ним надо относиться с осторожностью.

Одной из важных характеристик модема является скорость передачи данных. Лет десять назад наиболее распространенными среди дешевых модемов были модемы, передающие данные со скоростью 300 бод. Сегодня обычно применяются модемы, передающие по телефонной сети данные со скоростью 1200—2400 бод.

Слово «бод» обозначает единицу скорости, с которой модем посылает и принимает сигналы. Эта скорость измеряется количеством двоичных сигналов, передаваемых в единицу времени (бит/с). Используемый вами модем во время работы в сети может переслать 1200 бит (или около 120 символов) в с. Если вы пересылаете полную страницу текста (около 2500 знаков), это займет около 20 с. Переключив модем на 2400 бит, вы удвоите скорость передачи. Модем, имеющий высокую скорость, как правило, позволяет работать и с низкой скоростью. Скоростной модем дает возможность быстрее передавать данные и сократить время использования сети.

Чем быстрее может работать модем, тем выше его цена. Но учитывая, что время использования канала связи и хост-компьютера обычно достаточно дорого, предпочтительно пользоваться высокоскоростным модемом, несмотря на их большую стоимость.

13

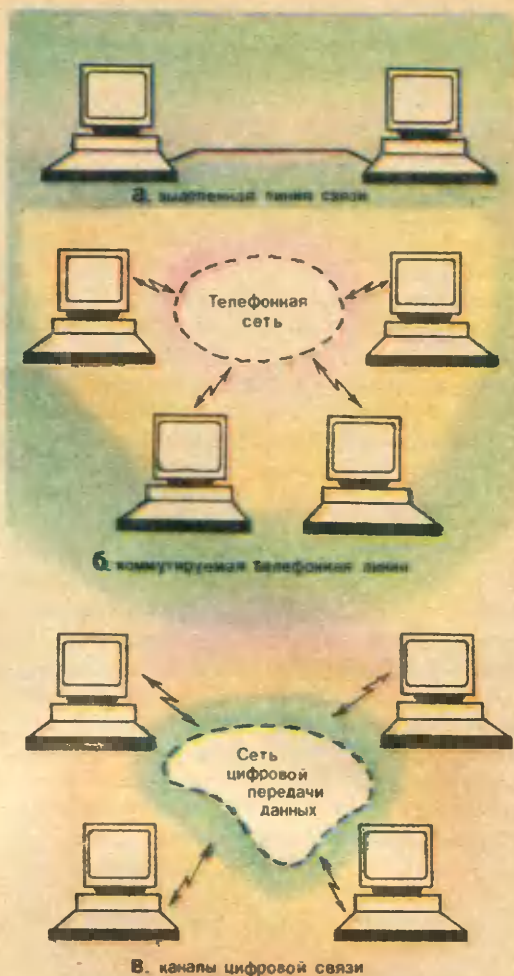
Максимальную скорость передачи данных ограничивает и качество телефонной сети. Лучшие из сегодняшних модемов в состоянии передавать информацию даже по недостаточно качественным телефонным линиям со скоростью 1200 бод. Для этого они имеют специальные средства корректировки ошибок, возникающих в процессе передачи данных. Такие модемы сравнительно дороги, но они обеспечивают связь через практически любую телефонную сеть и выполняют множество вспомогательных функций. Эти модемы часто называют «интеллектуальными».

Как показывает практика, для надежной работы в московской телефонной сети и по междугородным линиям связи предпочтительно пользоваться только «интеллектуальными» модемами. Но даже в этом случае вы не гарантированы от сбоев в телефонной линии во время сеанса связи. Вам не всегда удастся с первого раза переслать файл, размер которого превышает несколько десятков килобайт. Разделив свой файл на части по несколько килобайт каждая, вы значительно увеличите свои шансы быстро передать данные.

Для соединения модема с ЭВМ используется стандартный последовательный порт связи, который обычно имеется у каждого компьютера. Его называют «Стык-2» (в соответствии с отечественным ГОСТом) или RS-232C (международное название). Кабель, присоединяющий модем к этому порту, называется «интерфейсный кабель». Не путайте его с кабелем для принтера, который внешне выглядит очень похоже.

Линия связи. Для компьютерной коммуникации используют коммутируемые телефонные линии, выделенные линии связи и каналы цифровой связи (рис. 6).

Еще не так давно выделенные линии были основными каналами связи между ЭВМ на расстоянии. Они достаточно надежны, а скорость передачи данных по ним составляет от нескольких тысяч до нескольких десятков тысяч бод. С другой стороны, они дороги и, как правило, не очень экономичны. Сегодня они используются в системах, где постоянно передаются большие объемы ин-



формации или требуется срочная надежная передача данных (системы резервирования билетов и т. п.).

С появлением высококачественных модемов коммутируемые телефонные линии становятся наиболее распространенными каналами связи для компьютерной коммуникации. Когда модем не работает, вы можете использовать эту линию для обычного телефонного разговора, так как телефонный аппарат обычно подключен в телефонную сеть параллельно с модемом. Телефонные линии широко распространены, сравнительно дешевы и доступны практически каждому. В то же время коммутируемые телефонные каналы сильно зашумлены,

а скорость передачи данных по ним ограничена и составляет от нескольких сот до нескольких тысяч бод. Появление персональных компьютеров и «интеллектуальных» модемов, способных работать с коммутируемыми телефонными линиями, положило начало массовому проникновению компьютерной коммуникации в сферу управления, в научно-исследовательские учреждения, на производство, в быт. Сегодня она начинает использоваться и в практике работы школ во многих странах мира.

Каналы цифровой связи начали создавать совсем недавно. Они обязаны своим появлением революции в области вычислительной техники за последние десятилетия. Их пропускная способность составляет от десятков тысяч до сотен миллионов килобод. Они используются для быстрой передачи между ЭВМ больших и очень больших объемов информации. Появление и развитие цифровых каналов приводит к революции в технике связи, открывает немислимые еще вчера возможности для абонентов компьютерных сетей, обещает интегрировать в единое целое все существующие сегодня средства связи. В качестве примера на рис. 7 показано использо-

вание цифровых каналов для связи работающих в MoSTNet школ с их партнерами в США через систему «Теленет Сан-Франциско — Москва» (SFMT).

Данные между школами передаются по следующей цепочке каналов связи:

1 — телефонный канал: школа в Москве — MoSTNet — ВНИИ ПАС;

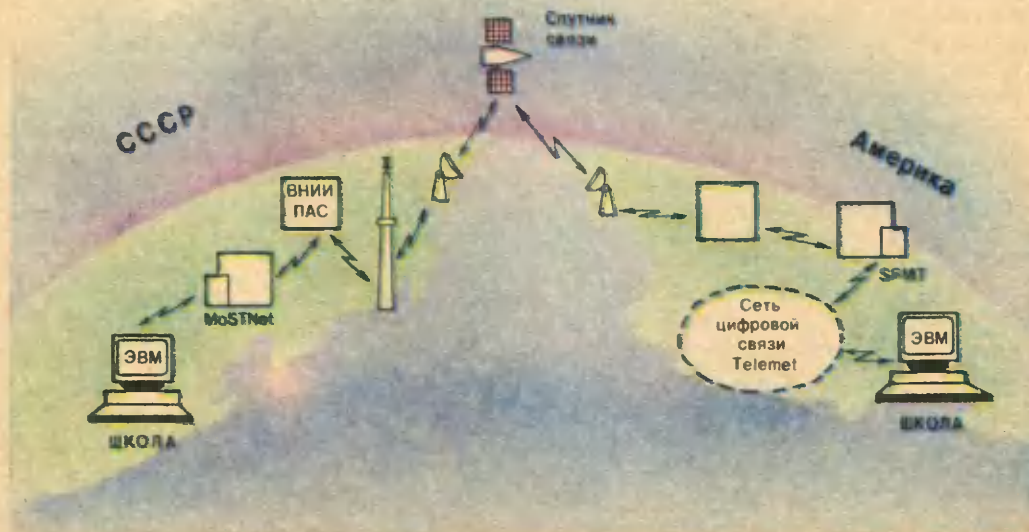
2 — цифровые каналы: ВНИИ ПАС — Останкино — Центр космической связи на Медвежьих озерах — спутник связи — приемная станция на Стейт-Айленде (штат Нью-Йорк) — волоконно-оптическая линия связи до Сан-Франциско — выделенный канал до хост-компьютера SFMT — узел цифровой связи системы «Теленет» в Сан-Франциско — местный узел связи системы «Теленет» в штате Нью-Йорк;

3 — телефонный канал: местный узел связи системы «Теленет» в штате Нью-Йорк — школа.

Коммуникационная программа. Для всякой работы на ЭВМ нужна соответствующая программа. Это универсальное правило справедливо и для компьютерной коммуникации. Имеется довольно много программных систем, обеспечивающих компьютерную коммуникацию на ЭВМ типа IBM PC. Боль-

15

7



шинство из них разработано за рубежом. Одна из широко применяемых систем такого типа называется «PROCOMM». Вы можете использовать для своей работы и многие другие программы. Все они, как правило, обеспечивают «дружественный интерфейс» с ЭВМ (большое количество меню), помогают вести «электронную телефонную книжку», позволяют автоматизировать процедуры соединения с хост-компьютером, отправления писем/получения писем и другие работы в компьютерной сети, и главное — все эти программные системы включают в себя набор стандартизованных процедур передачи данных, которые соответствуют общепринятым протоколам.

Лучшие из имеющихся систем построены так, что позволяют осваивать их многочисленные возможности постепенно, начиная с простых действий по отправке и получению электронных сообщений.

Головная машина сети. Все перечисленные выше технические и программные средства находятся на рабочем месте абонента сети. Для функционирования централизованной сети необходим хост-компьютер, который обслуживает всех абонентов, хранит передаваемую информацию, обеспечивает связь с другими сетями. В качестве головной машины обычно используют ЭВМ высокой производительности, с большим объемом памяти на жестком диске. Хост-компьютер, как правило, обеспечивает одновременную работу нескольких абонентов. Для этого к нему подсоединяют несколько линий связи. Головная машина постоянно включена и готова отвечать на вызовы абонентов сети. Работу головной машины обеспечивает специальная сетевая программа, которая ведет диалог с пользователями и поддерживает все действующие в сети протоколы связи. Сегодня в мире используются десятки сетевых программ, имеющих различный пользовательский интерфейс. Поэтому в каждой сети надо осваивать принятые здесь технические правила работы, соглашения о способах адресации корреспонденции и т. п. Это сдерживает сегодня распространение компьютерной коммуникации. Есть надежда, что в ближайшее де-

сятилетие Международная организация по стандартизации в области связи разработает набор международных стандартов, позволяющих решить эту проблему.

Протокол связи. В компьютерных сетях каждый абонент может использовать различные марки компьютеров, типы модемов, линии связи, коммуникационные программы. Чтобы все это оборудование работало согласованно, специалисты работают над техническими соглашениями (протоколами) об унификации формы представления и способов пересылки сообщений, процедуры их интерпретации, совместной работы различного оборудования в сетях. Эта работа ведется уже не одно десятилетие. Сегодня четко определены функции, которые необходимо выполнять в компьютерных сетях, определена их иерархия (подчиненность) и унифицированы соответствующие протоколы. Международная организация по стандартизации (ISO) подготовила и ввела в действие такую многоуровневую (иерархическую) структуру протоколов.

Полезно знать, что подготавливаемая архитектура получила название OSI (соединение открытых систем) и вклю-

Таблица

Уровень протокола	Функция
7 Прикладной	Передача файлов, заданий, сообщений, доступ к базам данных
6 Представления	Форма представления данных (КОИ-8, ASCIT), сжатие данных, кодирование
5 Сеанса	Соединение с коммуникационной программой, синхронизация передаваемых единиц, обработка прерываний и приоритетов
4 Транспортный	Установление связи по каналу
3 Сетевой	Маршрутизация пересылаемых сегментов данных по адресной информации.
2 Канальный	Установление связи между соседними узлами сети, обработка ошибок передачи данных между соседними узлами
1 Физический	Согласование электрических характеристик и конструктивной формы соединений

чает семь уровней функционирования систем передачи данных (таблица).

Протоколы физического уровня лежат в основании системы. Они определяют электрические характеристики сигналов и конструктивные особенности устройств, обеспечивающие физическую совместимость оборудования.

Протоколы канального уровня определяют процедуры установления связи между соседними узлами сети, способы обнаружения ошибок в передаче данных между ними.

Протоколы сетевого уровня определяют соглашения о способах выбора маршрута пересылки сообщений и использования адресной информации, содержащейся в сообщении.

Протоколы транспортного уровня определяют процедуры установления связи по каналу передачи данных.

Протоколы сеанса рассматривают вопросы установления связи с программой на ЭВМ абонента, синхронизации передаваемых сообщений, обработки прерываний и приоритетов.

Протоколы уровня представления определяют форму представления данных (стандарт ASCII и т. п.), их сжатие, кодирование.

Протоколы прикладного уровня определяют функционирование системы при автоматической передаче файлов, сообщений, доступе к базам данных и т. п.

Каждое достигнутое соглашение оформляется в виде отдельного протокола и в таком виде становится известно разработчикам технических средств, программного обеспечения и абонентам компьютерных сетей.

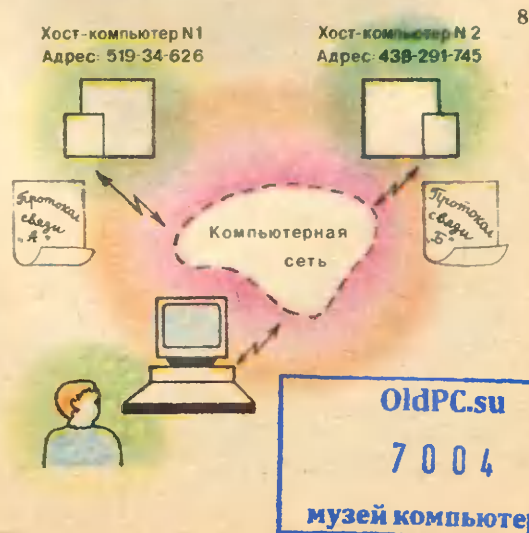
Прогресс в области средств связи идет сегодня необычайно быстро. По мере развития средств вычислительной техники и связи каждые пять — десять лет появляются новые технические решения, ранее утвержденные протоколы пересматриваются и на их основе выпускаются новые. Появление нового протокола не отменяет действие старых. Поэтому большинство систем в процессе эксплуатации поддерживают несколько протоколов. Старые протоколы обычно менее удобны и постепенно вытесняются новыми. Таким образом, каждый из абонентов имеет возможность выбрать, ка-

ким из протоколов (технических соглашений) он будет пользоваться во время своей работы в сети. Этот выбор определяется прежде всего оборудованием и программным обеспечением хост-машины, а также техническими возможностями (ЭВМ, модем, линия связи, коммуникационная программа), которыми располагает абонент, и соображениями эффективности его работы.

Выбор протокола проводится, как правило, лишь один раз — при подготовке оборудования к работе и первом подключении к компьютерной сети. Устанавливаются такие параметры, как режим связи (дуплексный/полудуплексный), скорость передачи данных по каналу связи (300/1200/2400 бод), тип используемого терминала (VT-100), проверка на четность и др. Возможный выбор для абонента MoSTNet описан в разделе «Руководство абонента MoSTNet».

Компьютерные сети и информационные службы, с которыми приходится работать, используют различные протоколы. Поэтому телекоммуникационная программа хранит все принятые значения параметров связи и автоматически устанавливает их каждый раз, когда вы подсоединяетесь к соответствующей системе.

Из рис. 8 видно, что абонент сети может обращаться к различным адресам и использовать много разных протоколов.

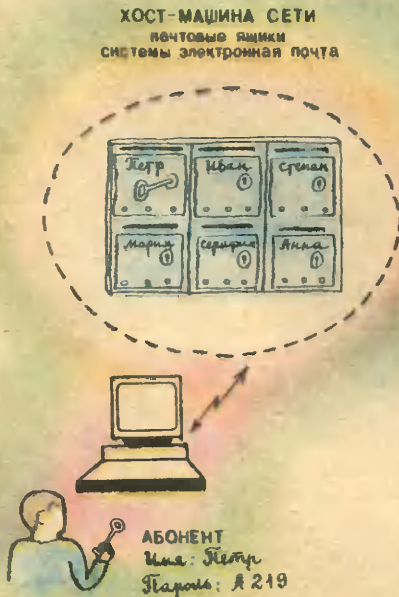


Виды услуг, предоставляемых абонентам компьютерных сетей

Компьютерная коммуникация позволяет легко и быстро отправлять и получать любые данные (тексты, таблицы, программы для ЭВМ, рисунки), подготовленные с помощью вычислительной машины. Все это попадает непосредственно на рабочий стол абонента компьютерной сети. Если абонент путешествует с портативным (переносным) компьютером, он в любой момент может воспользоваться услугами сети, подключившись к ней через ближайшую телефонную розетку. Создатели компьютерных сетей используют эти возможности для того, чтобы предоставлять своим клиентам различные услуги.

18 **Электронная почта.** Наиболее привычным видом услуг, которые предоставляют своим абонентам компьютерные сети, является электронная почта. Каждый абонент при регистрации получает свой собственный «почтовый ящик» — некоторый объем памяти на хост-машине, в который попадают все адресованные ему сообщения (рис. 9).

9

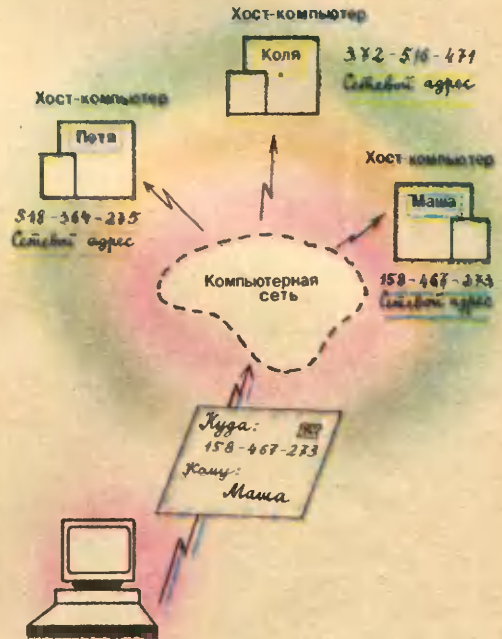


Имена почтовых ящиков и сведения об их владельцах доступны всем абонентам сети. Войдя в сеть, вы можете

послать сообщение по любому адресу, и это сообщение попадет в соответствующий почтовый ящик. Для того чтобы получить поступившие вам сообщения, вы должны сообщить системе имя почтового ящика и пароль, дающий право на получение информации. Каждый владелец почтового ящика сам устанавливает пароль, закрывая тем самым свой почтовый ящик от посторонних.

Если несколько компьютерных сетей связаны друг с другом, абоненты разных сетей также могут пересылать письма друг другу. Для этого, посылая письмо, они должны указать имя сети и почтовый ящик своего адресата (рис. 10).

10



Используя различные протоколы, вы можете положить в почтовый ящик своего корреспондента как текстовые, так и двоичные файлы.

С помощью двоичных файлов передают любые виды данных: программы для ЭВМ в машинных кодах, звук, текстовые файлы с любой кодировкой символов. Единственное требование состоит в том, чтобы ваш адресат сумел правильно интерпретировать (по-

нять) переданное ему сообщение. Если вы передали текст программы, написанной в кодах компьютера типа IBM PC, ее получатель должен иметь такой же компьютер для выполнения поступившей программы. Если вы передаете двоичный файл с картинкой, подготовленной с помощью графического редактора «Paintbrush», ваш адресат должен использовать свою версию этой программы, чтобы распечатать полученное изображение на принтере.

Главным ограничением при передаче данных по сети является объем передаваемых данных. Страница машинописного текста обычно имеет объем около 2,5 Кбайт. Картинка размером со страницу может потребовать файл в десятки раз большего объема. Зная скорость передачи данных по компьютерной сети, вы можете легко подсчитать, сколько времени займет пересылка одной картинки по сети и сколько вы должны будете уплатить за это владельцам SFMT.

Базы данных. Доступ к базам данных — типичный вид услуг, предоставляемых абонентам компьютерной сети. Подключившись к сети через телефонную линию и задав сетевой адрес нужной ему базы данных, абонент подключается к ней и в режиме диалога может получить требуемую ему информацию. Стоимость информационных услуг обычно прямо пропорциональна времени работы с системой.

Первые эксперименты по использованию баз данных учащимися и педагогами школ начались около 15 лет назад. Сейчас в мире абонентам телекоммуникационных сетей доступны сотни баз данных практически по всем областям знаний. Основными потребителями этой информации являются исследовательские центры и учебные заведения. Каждая четвертая школа в Канаде и США имеет сегодня доступ к базам данных. Наиболее популярны справочно-библиографические информационные системы. Они предоставляют абонентам исчерпывающую библиографическую информацию, а иногда — и тексты интересующих их публикаций. Многие библиотекари в школах США используют этот источник для обеспе-

чения информацией учащихся, работающих над своими учебными исследовательскими проектами.

В Москве сегодня действует несколько баз данных. ВНИИ прикладных автоматизированных систем при ГКНТ СССР предоставляет всем желающим доступ в справочную систему «Стоимость международных коммуникационных услуг». ВНИИ патентной экспертизы поддерживает базу данных «Патентная информация». Свои базы данных ведут Институт научной информации по общественным наукам АН СССР, Всесоюзный институт научно-технической информации. Доступ в эти базы данных платный.

Электронные доски. Электронная доска объявлений по своему замыслу аналогична обычной доске объявлений, которая висит на стене в каждой школе. При использовании базы данных любой абонент может прочитать всю хранящуюся там информацию, но не имеет права ее изменять. Пользуясь электронной почтой, абонент может записать информацию в любой почтовый ящик, но прочитать и изменить данные он может только в своем собственном. Электронная доска объявлений позво-

19

11



ляет каждому записать туда любую информацию, представляющую интерес для абонентов системы, и прочитать сообщения, помещенные туда другими абонентами. Такой режим работы дает возможность использовать электронные доски объявлений для проведения компьютерных конференций (рис. 11).

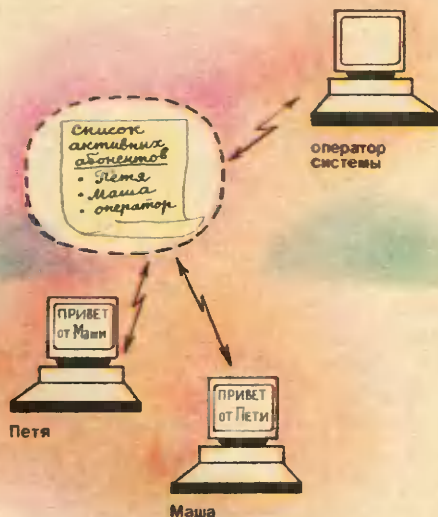
Особую роль в проведении компьютерной конференции играет ее ведущий. Ведущий получает у оператора системы разрешение на проведение конференции, открывает ее, приглашает участников, организует и поддерживает их работу. Ведущий может закрыть конференцию, когда необходимость в ней отпадает. В отличие от остальных абонентов системы ведущий имеет право удалять сообщения, помещенные на доску объявлений.

Участником конференции может стать каждый абонент сети. Если конференция объявлена «открытой», то для участия в ней достаточно просто сообщить об этом системе (зарегистрироваться). Если конференция объявлена «закрытой», то, для того чтобы стать ее участником, вы должны получить разрешение у ведущего.

Электронные доски объявлений позволяют абонентам компьютерной сети публиковать объявления, представляющие интерес для целой группы пользователей, вести коллективное обсуждение интересующих их вопросов. Участие в компьютерных конференциях, использующих электронные доски объявлений, аналогично созданию электронного почтового ящика, к которому в удобное для них время могут обращаться несколько абонентов. Поэтому такие конференции иногда называют «отсроченными».

«Беседа» абонентов в компьютерной сети. Обычно к хост-машине подсоединено несколько телефонных линий и несколько абонентов сети работают с ней одновременно. Благодаря этому абоненты могут «беседовать» в сети, непосредственно пересылая сообщения друг другу (рис. 12).

Такой способ обмена сообщениями обычно называют «общение в реальном времени» (иногда для этого режима работы используют термин «онлайн»



от англ. on-Line). Типичный пример общения в реальном времени (онлайн) — обмен сообщениями с оператором системы во время сеанса связи.

Общение в реальном времени позволяет организовать проведение «компьютерной конференции в реальном времени». Все участники такой конференции должны быть одновременно связаны с хост-машиной. «Компьютерная конференция в реальном времени» позволяет каждому участнику читать все появляющиеся сообщения в процессе их ввода в систему и незамедлительно реагировать на них. В отличие от «отсроченных» компьютерных конференций, которые используют электронные доски объявлений, компьютерная конференция в реальном времени обходится абонентам сети существенно дороже, так как вынуждает их занимать канал связи и хост-машину в течение всей конференции.

Информационные услуги. Большинство существующих сегодня коммерческих компьютерных сетей предоставляет своим абонентам все перечисленные выше виды коммуникационных услуг, а также разнообразные информационные услуги. Вы можете получать и посылать почту, просматривать сообщения о последних новостях в области спорта

и политики, знакомиться с прогнозом погоды и с биржевыми сводками, работать с базами данных, знакомиться с различными электронными изданиями (электронное издание журнала «PC World» выходит на месяц раньше, чем соответствующее печатное издание) и рекламными объявлениями, делать заказы на приобретение различных товаров по каталогу. Многие системы предоставляют своим абонентам возможность играть в компьютерные игры.

Программное обеспечение хост-компьютера ведет постоянный учет объема передаваемой и получаемой вами информации, времени работы в сети и подсчитывает ваши расходы. В любой момент вы можете справиться о стоимости предоставляемых вам услуг и состоянии вашего счета.

О перспективах развития средств компьютерной коммуникации

Сегодня, когда мир движется к «информационному обществу», средства передачи информации развиваются необычайно быстро. Компьютерная коммуникация — одно из средств связи будущего, которые станут повсеместно использовать: цифровое представление передаваемой информации в канале связи, компьютер как средство управления передачей данных, подготовки и интерпретации сообщений. Среди возможных новинок 90-х гг. ожидается:

переход к комплексным цифровым системам передачи данных;

дальнейшее повышение скорости и надежности передачи данных по каналам связи, широкое распространение волоконно-оптических и спутниковых систем связи;

включение в состав передаваемых по электронной почте сообщений «звуковых писем» (новый компьютер Стива Джобса Next имеет для этого все необходимое);

широкий обмен по компьютерной связи аудио- и видеоинформацией, подготавливаемой и отображаемой с помощью систем типа «гипермедия».

Освоение современных средств компьютерной коммуникации — первый шаг на пути к освоению поистине неисчерпаемых возможностей средств компьютерной связи будущего.

Материалы для дополнительного чтения

1. *Хасэгава Х.* Мир компьютеров в вопросах и ответах. Кн. 2. М.: Мир, 1988.

В книге, выполненной в виде комикса, весьма доходчиво и увлекательно описаны принципы построения и работы компьютерных сетей. Она может использоваться как конспект в картинках или сборник опорных сигналов для изучения технических основ вычислительной техники и компьютерных сетей в школе. Настоятельно рекомендуется как для подготовки занятий со школьниками, так и для самостоятельной работы учащихся.

2. В мире ПК. 1989. № 4.

Подборка статей в этом номере популярного журнала по вычислительной технике, выпускаемого издательством «Мир», посвящена компьютерным сетям и компьютерной коммуникации.

3. *Самойленко В. А.* Компьютерные сети. М.: Наука, 1988.

В книге очень сухо, но популярно рассказано о структуре сетей ЭВМ. Много внимания уделено техническим вопросам построения и реализации протоколов связи.

4. *Уэбстер Б.* Что в будущем... Next? // В мире ПК. 1989. № 3.

В этой статье описаны основные характеристики перспективной персональной ЭВМ, разработанной создателем всемирно известных персональных компьютеров Apple и Macintosh Стивом Джобсом. Процессор сигналов, которым оснащена машина Next, позволяет подготавливать, передавать и принимать «голосовые» электронные письма.

На рис. 7 вместо Telemet следует читать Telenet

Информатика в межшкольных УПК

В межшкольных учебно-производственных комбинатах массовое распространение получили два направления, связанные с электронно-вычислительной техникой: преподавание основ информатики и ЭВТ и профессиональное обучение старшеклассников (по профессиям, связанным с эксплуатацией и обслуживанием ЭВТ). Рассмотрим особенности каждого из этих направлений.

22 Сразу отметим, что обучение основам информатики и электронно-вычислительной техники на базе межшкольного УПК — мера временная, в большой степени вынужденная. Она связана с невозможностью пока еще создать соответствующий кабинет в каждой школе. Информатика — не трудовое обучение, и, следовательно, предмет, не свойственный деятельности УПК. Вероятно, отчасти это и повлияло на то, что база информатики оказалась в роли «пасынка» на комбинатах. Так как по Положению о межшкольном УПК комбинаты содержатся в основном за счет создавших их предприятий, в них не предусмотрены доплаты педагогам за заведование кабинетами, обслуживание техники и т. п. Только инструктивным письмом Гособразования СССР от 18 апреля 1990 г. № 16 «О некоторых вопросах работы межшкольных УПК и оплаты труда их работников» установлено, что «...в случае, если на межшкольные учебно-производственные комбинаты возложено преподавание школьного курса «Основы информатики и вычислительной техники» и в связи с этим обору́дованы учебные кабинеты вычислительной техники, переданы из школ соответствующие учебные часы и переведены учителя, то согласно постановлению ЦК КПСС и Совета Министров СССР от 28 марта 1985 г. № 271 учителям УПК, которые ведут занятия по этому предмету, могут производиться доплаты за заведование этими кабинетами и обслуживание вычислительной техники

соответственно в размере 10 рублей в месяц и 5 рублей за каждый работающий компьютер». Аналогичная запись сделана и в новой редакции пункта 66 инструкции об исчислении заработной платы работников просвещения (изменения в инструкцию внесены приказом Гособразования СССР от 8 июня 1990 г. № 400).

Другое новшество, связанное с обучением информатике на УПК, отражено в приказе Гособразования СССР от 26 октября 1990 г. № 675 «Об утверждении примерных штатных нормативов межшкольных учебно-производственных комбинатов» (пункт 2): «Установить, что в случае, если на межшкольные учебно-производственные комбинаты трудового обучения и профессиональной ориентации учащихся дополнительно возложено преподавание школьного курса «Основы информатики и вычислительной техники» и из школ в УПК переданы учебные часы на этот предмет, то учащиеся, проходящие обучение по данному предмету, учитываются дополнительно при определении штата и должностных окладов руководящих работников». Эта запись означает, что ес ли, например, учащийся изучает на УПК только основы информатики, то он теперь также включается в контингент УПК, а если он изучает на комбинате и основы информатики, и какую-либо профессию, то учитывается в контингенте дважды.

Как видим, с точки зрения материальных стимулов положение с обучением основам информатики и вычислительной техники на УПК становится существенно лучше.

Профессиональное обучение по обслуживанию и эксплуатации электронно-вычислительной техники на УПК — предмет особого разговора.

Прежде всего — о профессиях, по которым ведется обучение. Общепринятые наименования рабочих профессий и должностей служащих, на которые раз-

работаны и утверждены квалификационные характеристики, т. е. определено в установленном порядке содержание труда, приведены в Общесоюзном классификаторе профессий рабочих, должностей служащих и тарифных разрядов (Госстандарт СССР, 1988). В нем значатся среди рабочих профессий: оператор электронно-вычислительных машин, оператор вычислительных машин, электромеханик по средствам автоматизации и приборам технологического оборудования, электромеханик по ремонту и обслуживанию счетно-вычислительных машин и др. Среди служащих (специалистов) в классификаторе указаны техник-программист, инженер-программист и др. Вместе с тем в ОКП нет таких профессий, как лаборант-программист, оператор устройств подготовки данных для ЭВМ и т. п. Все это означает следующее. Если целью обучения по профессии является присвоение квалификации, то наименование профессии и содержание труда (обучения) должны определяться в соответствии с общесоюзными документами. В противном случае аттестационный документ не будет иметь юридической силы. Многие даже не поймут, что он означает. С другой стороны ясно, что школьника, выпускника общеобразовательной школы невозможно аттестовать по профессии, требующей среднего специального или даже высшего образования, к каковым и относятся, в частности, программисты.

Что касается новых организационных основ профессионального обучения школьников, в том числе и на УПК, то они подробно изложены в письме Гособразования СССР от 14 сентября 1990 г. № 46 «О профессиональном обучении школьников». В условиях отмены обязательности и массовости обучения школьников рабочим профессиям оно рассматривается как одно из направлений дифференциации школьного образования. Наиболее целесообразный путь — создание специализированных классов и школ, комплектуемых из старшеклассников, изъявивших соответствующее желание. Возможно и обучение в индивидуальной форме под руководством специалистов или квалифицированных рабочих предприятий, но

при условии контроля со стороны школы или межшкольного УПК. Желательна договорная основа с предприятиями в организации обучения профессиям.

В зависимости от сложности профессии, избранного варианта учебного плана школы, типоразмеров учебно-производственного оборудования и других местных условий профессиональное обучение школьников может осуществляться начиная с X или XI, а в отдельных случаях — с VIII класса. Обучение возможно как в два этапа «профиль — профессия» в соответствии с инструктивным письмом Минпроса СССР от 28.07.86 № 70-М, так и в один этап, сразу по конкретной профессии. При любом варианте общее количество учебного времени, отводимого в общеобразовательной школе на профессиональное обучение, не должно быть меньше, чем при подготовке новых рабочих данной профессии на производстве (сроки обучения по каждой профессии указаны в перечне профессий для подготовки рабочих на производстве, утвержденном Госпрофобром СССР в 1987 г., при этом одному месяцу обучения соответствуют 150 часов учебного времени).

Для профессионального обучения старшеклассников наряду с временем, предусмотренным учебными планами школ на трудовое обучение, общественно полезный, производительный труд, может также использоваться время, отводимое на курсы по выбору и факультативные занятия в соответствующих классах.

Если для профессионального обучения и руководства трудом учащихся в УПК невозможно привлечь в качестве мастеров производственного обучения квалифицированных рабочих (колхозников) или специалистов предприятий, то в соответствии с инструктивным письмом Гособразования СССР от 30.01.90 № 4 «О введении должностей мастеров производственного обучения в межшкольных УПК и условиях оплаты их труда» в комбинатах вводятся должности мастеров.

О мастерах УПК следует сказать особо. До введения этой должности вместо недостающего (или отозванного) работ-

ника предприятия на комбинате мог работать только учитель, проводя как теоретические, так и практические занятия в счет своей учебной нагрузки. Это в свою очередь порождало определенную неразбериху в оплате труда, продолжительности отпусков работающих на УПК. Сейчас можно все упорядочить: учителя проводят занятия по теории, мастера (как штатные, так и направленные предприятиями) — практическое (производственное) обучение. Мастерам производственного обучения установлена 41-часовая рабочая неделя и учебная нагрузка 1050 часов в год. В нагрузку включается проведение практических занятий, руководство общественно полезным, производительным трудом и трудовой практикой. В оставшееся время мастер участвует в обеспечении учебно-производственной базы, ведет профориентационную и другую воспитательную работу. За проведение мастером теоретических занятий, совмещение профессий, расширение зоны обслуживания и объема выполняемых работ предусмотрена дополнительная оп-

лата.

Читателям, заинтересованным в более подробной информации по затронутым вопросам, советуем ознакомиться с упомянутыми выше документами полностью.

Л и т е р а т у р а

1. О введении должностей мастеров производственного обучения в межшкольных УПК и условиях оплаты их труда. Инструктивное письмо Гособразования СССР от 30 января 1990 г. № 4 «Школа и производство», 1990, № 4.

2. О некоторых вопросах работы межшкольных УПК и оплаты труда их работников. Инструктивное письмо Гособразования СССР от 18 апреля 1990 года № 16. «Школа и производство», 1990, № 8.

3. О профессиональном обучении школьников. Инструктивное письмо Гособразования СССР от 14 сентября 1990 г. № 46. «Школа и производство», 1990, № 12.

4. Об утверждении примерных штатных нормативов межшкольных учебно-производственных комбинатов. Приказ Гособразования СССР от 26 октября 1990 г. № 675. «Школа и производство», 1991, № 1.



MIT—LOGO

Интегрированная среда программирования, разработанная С. Пейпертом, позволяет использовать компьютер как инструмент, с помощью которого обучение становится более интересным, быстрым и простым, а полученные знания и навыки — более глубокими и обобщенными.

Цена комплекта: в заказе до 10 комплектов — 500 руб.;
от 10 до 100 — 400 руб.;
свыше 100 — 400 руб.

WRITER

Текстовый редактор-документатор. Система подготовки текстов, в функции которой входит редактирование текста, форматирование и печать.

Цена комплекта: в заказе до 10 комплектов — 500 руб.;
от 10 до 100 — 400 руб.;
свыше 100 — 350 руб.

ВAM

Универсальный сетевой монитор, предназначенный для поддержки ряда неспецифицированных и некоторых ориентированных программных средств.

Цена комплекта: в заказе до 10 комплектов — 400 руб.;
от 10 до 100 — 300 руб.;
свыше 100 — 250 руб.

А. АВЕРБУХ

Как от Е-практикума перейти к устройству ЭВМ

Эта статья предназначена для тех учителей, которые в 1989/90 учебном году начали преподавать ОИВТ по учебнику под ред. А. П. Ершова (А. П. Ершов, А. Г. Кушниренко, Г. В. Лебедев, А. Л. Семенов, А. Х. Шень. «Просвещение», 1988).

Во второй четверти этого учебного года вы закончили изучение первой главы, открыли вторую и расстроились. Где содержательный материал, где логика его построения, что объяснять в классе, что задавать на дом?

Хочется надеяться, что эта статья поможет вам.

1. Введение

В отличие от первой главы, где делается попытка формирования структурного алгоритмического мышления довольно нетрадиционным путем, вторая глава содержит более-менее знакомый материал об устройстве ЭВМ: физических и логических основах функционирования, машинных командах, техническом устройстве периферийного оборудования, стандартном программном обеспечении и т. д. По этим вопросам можно найти сравнительно много дополнительной литературы для школьников — от книги, названной в русском переводе «Знакомьтесь: компьютер» (М.: Мир, 1989), до учебного пособия для VII—VIII классов средней школы «Электронно-вычислительная техника» под ред. Я. А. Ваграменко (М.: Просвещение, 1988).

Вторая глава не имеет четкой логики построения. Кроме того, авторы учебника, следуя принципу «если нет возможности охватить весь лес целиком, то займемся изучением отдельных листьев», не делают хотя бы краткого обобщения, и при этом неизбежно теряется общая картина.

Здесь мы постараемся восполнить этот пробел. Кроме того, дадим более подробный разбор тех понятий, которые, по нашему мнению, способствуют дальнейшему развитию логического и алгоритмического мышления учеников.

2. Фиаско коммутативного закона

Приступая к преподаванию второй главы, некоторые учителя, особенно бывшие инженеры и настоящие физики, испытывают облегчение, как заблудившийся путник, выйдя на знакомую тропинку. Возникает иллюзия, будто методика преподавания учебных тем второй главы давно известна и проверена многолетней практикой.

Попробуем показать, что это впечатление обманчиво.

Методика обучения взрослых программистов в свое время была механически перенесена в ПТУ, техникумы, школы. Поэтому, во-первых, материал излагался с опорой на хорошее знание основ физики, математики, на элементарную (хотя бы) техническую грамотность. Во-вторых, предполагалась заинтересованность обучаемого в новых зна-

ниях. Любой учитель знает: у современного школьника обычно нет ни того, что «во-первых», ни того, что «во-вторых».

Кроме того, время, необходимое для изучения устройства ЭВМ по старой методике, значительно превышает возможности курса в общеобразовательной школе.

Но главное, пожалуй, не это. Главное то, что в педагогике от перемены мест слагаемых сумма меняется. Когда-то изучение устройства ЭВМ стояло в самом начале курса информатики. В этом была своя логика: «Вот вам алгебра Буля и двоичная арифметика. А вот электромагнитное реле (триод, транзистор, кристалл кремния), с помощью которого теоретически построения обретают физический смысл. Теперь давайте соберем вентиль, из него — ячейку, затем — блок, наконец — модель вычислительной машины. А теперь попробуем понять, зачем эта машина нам нужна». После этого излагалось программирование — от ассемблера к языку высокого уровня. Логика железная, ничего лишнего.

В некоторых современных учебниках информатики глава об устройстве ЭВМ переместилась в середину (иногда даже в конец) курса. Однако сама глава при этом практически не претерпела изменений. Логика нарушалась: «Вот вам ЭВМ. Она понимает такие-то команды на языке высокого уровня. Видите, она вас слушается! А теперь давайте из вентиля соберем ЭВМ». Зачем ее собирать, если она уже есть?

Непросто убедить школьника в необходимости изучения устройства ЭВМ, если он уже почувствовал, что и без этих знаний можно вполне успешно ею пользоваться. А если и сам учитель считает эти знания лишними, то ситуация патовая.

Таким образом, при отсутствии *мотивации* глава об устройстве ЭВМ кажется многим ученикам второстепенной и скучной, а традиционная методика преподавания не срабатывает.

Поэтому важнейшая задача учителя — убедить ученика в необходимости изучения этого, на первый взгляд ненужного материала. Можно, например,

опереться на тот незаслуженный пиетет, который имеет у нас всякий человек, понимающий машинный язык и знающий устройство этой загадочной ЭВМ. Парадоксально, но «нянька», обучающая ЭВМ выполнять простейшие действия и понимать несколько слов на человеческом языке, окружена ореолом кудесника и чудотворца.

Другой путь: сделать переход от алгоритмического языка ко второй главе максимально плавным и незаметным. Попробуем пойти по второму пути. Задача еще более усложняется из-за того, что в учебнике этот переход настолько незаметен, что приходится признать, что его нет вовсе.

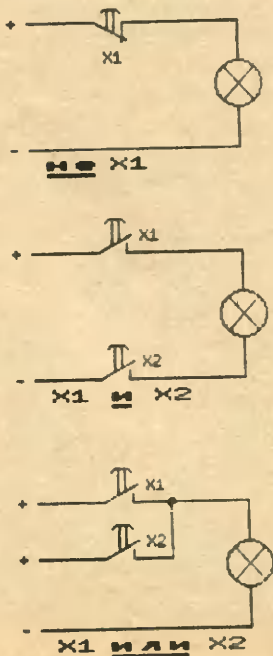
3. Алгебра логики, двоичное кодирование, выключатели и китайцы

Начнем с отдельных параграфов первой главы, вернемся к логическим операциям, выражениям и величинам. Затем напомним запись значений логической величины с помощью нолей и единиц, а в связи с этим повторим двоичное кодирование, которое, кстати, должно оставаться на ознакомительном уровне на манер китайской грамоты: известно, что существует китайский (читай — машинный) язык, что с его помощью можно закодировать любую информацию, что имеются автоматические переводчики с китайского и обратно (трансляторы, преобразователи), что в природе даже есть живые китайцы (системные программисты), владеющие китайским свободно. Но не следует требовать от учащихся зубрить иероглифы, т. е. тренироваться записывать информацию на машинном языке, переводить числа из одной системы счисления в другую.

Итак, двоичная система оказалась удобной в качестве языка логики, так как логические величины могут принимать всего два значения: да («истина») или нет («ложь»). Это поняли спустя почти сто лет после того, как английский математик-самоучка Джордж Буль сформулировал основные положения своей алгебры в работах «Математический анализ логики» (1847), «Исследование законов мышления» (1854) и др.

Булева алгебра была развита американским ученым Чарлзом Сандерсом Пирсом. Хотя Пирс и понимал, что логические выражения можно наполнять физическим смыслом с помощью, например, электрических схем, но развитие практического приложения алгебры логики связано с именем другого американца, основоположника теории информации Клода Шеннона, одного из разработчиков теоретических основ вычислительной техники. В своей диссертации Шеннон, сочетавший в себе качества хорошего математика и отличного специалиста по электротехнике, раскрыл связи между двоичным способом кодирования информации, алгеброй логики и электрическими (в те времена — релейными) схемами. Затем на смену реле пришли электронные лампы, затем — транзисторы, интегральные схемы и т. д.

Но независимо от элементной базы важнейшие принципы работы ЭВМ одни и те же. В роли «элементарной частицы» всегда выступает разновидность выключателя. И если правильно соединить очень много простых бытовых выключателей и нанять очень много людей, которые будут ими в нужный момент щелкать, то получится вычислительная машина.



Покажем это на простейших схемах из кнопок и лампочек. Нажатие на кнопку означает, что некоторая логическая величина принимает значение да. Если логическое выражение тоже равно да, то лампочка загорается (рис. 1).

Из простых выключателей мы собрали действующие модели *вентилей*, т. е. схем, реализующих логические операции. Таким образом, опираясь на материал первой главы, мы плавно перешли к изучению *логических принципов работы вычислительной техники*.

Однако на время прервемся и поговорим о *физических принципах*. Начнем с одной из разновидностей выключателя — МОП-транзистора.

4. МОП-транзисторы, маски, дырки и щепки

27

Основная цель заключается не столько в том, чтобы познакомить аудиторию с физическими основами работы р—п-перехода, сколько в том, чтобы снять с ЭВМ налет таинственности (особенно в нынешнее время проторенессанса мистической культуры в лице домовых и психотерапевтов).

Рамки журнальной статьи не позволяют дать описание работы МОП-транзистора. Этот материал можно легко найти в книгах, например в школьном учебнике физики.

Почему именно МОП-транзистор выбран для изучения? Потому, что внутри современных ЭВМ четвертого поколения происходят точно те же физические процессы, только транзисторы стали очень маленькими. На 1 см² размещается более полумиллиона элементов (транзисторов, конденсаторов, резисторов). Эти элементы составляют *интегральную микросхему* (ИМС).

Интегральные микросхемы часто называют чипами — от английского слова chips, обозначающего щепки или жареный картофель. Основа чипов — кремний, имеющий четыре валентных электрона. Чистый кремний не обладает электрической проводимостью. Но при внесении в него микропримеси пятивалентного элемента (фосфора) возникает лишний электрон, а при добавлении мельчайшей частицы трехвалентного

элемента (обычно бора) возникает так называемая дырка и кремний приобретает соответственную электронную или дырочную проводимость.

Миниатюризация интегральных схем не беспредельна. Причина этого не только в возрастающей трудности проектирования масок для изготовления микросхемы (микросхема имеет слоистое строение; рисунок каждого слоя называют маской). Чем миниатюрнее изделие, тем больше оно подвержено внешним влияниям. Это порождает жесткие требования к технологическому процессу. В помещении цеха микропроцессорного производства запыленность не превышает 4 частиц на литр (в обычном цехе этот показатель равен миллиону!).

28 Кроме того, при плотности свыше 10 млн. элементов на квадратный сантиметр снижается помехоустойчивость микросхемы. Непредсказуемое переключение микроскопического транзистора может произойти под влиянием диффузии атомов в кристалле кремния, при действии космического излучения, слабых полей.

(ПЗУ), перепрограммируемое запоминающее устройство (ППЗУ), оперативное запоминающее устройство (ОЗУ), генератор тактовых импульсов, преобразователи входных-выходных сигналов.

5. Что умеет ЭВМ?

Мы знаем, что метод пошаговой детализации позволяет представить алгоритм как последовательность вспомогательных алгоритмов. Вспомогательные алгоритмы, как нам известно, состоят из команд. Но рассуждения можно продолжить: каждая команда состоит из нескольких операций, которые, в свою очередь, делятся на еще более простые...

Так что же «на самом деле» умеет ЭВМ? Ответ на этот вопрос зависит от того, что именно подразумевается под термином «ЭВМ» и с какой точки зрения на нее смотрят (табл. 1).

Во время изучения первой главы мы рассматривали ЭВМ как универсальный «черный ящик» со сменяемыми алгоритмами обработки информации. Иными словами, мы пребывали в верхней строке табл. 1 и не опускались до изучения внутренностей машины.

Таблица 1

С ТОЧКИ ЗРЕНИЯ...	ЭВМ УМЕЕТ...	ЭВМ СОСТОИТ ИЗ...
пользователя	исполнять алгоритмы	щели для диска, дисплея...
программиста	исполнять команды алгоритмического языка	процессора, ОЗУ, ПЗУ...
системного программиста	исполнять машинные команды	регистров, ячеек памяти...
проектировщика ЭВМ	выполнять логические операции	вентилей, микросхем...
разработчика микросхем	ничего	кристалла кремния с примесями

Однако пора от физических и технологических проблем (т. е. от специальных вопросов) вернуться к общему вопросу курса — к логическим основам устройства ЭВМ. Этот вопрос имеет гораздо большее общеобразовательное значение, поэтому остановимся на нем подробнее. Но перед этим для обобщения (хотя бы минимального) перечислим основные микросхемы, входящие в состав ЭВМ: различные процессоры, постоянное запоминающее устройство

Теперь же мы начали с самой нижней строчки (см. «МОП-транзисторы, маски, дырки и щепки») и кратко познакомились с устройством микросхем. Поднимемся вверх на одну строку и соберем из микросхем процессор.

6. Проектируем процессор

Независимо от принципа работы вентилей договоримся их обозначать в соответствии с принятым у нас стандартом (рис. 2).



Таблица 2

X	Y
0	1
1	0

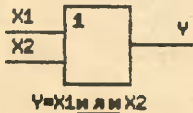


Таблица 3

X1	X2	Y
0	0	0
1	0	1
0	1	1
1	1	1



Таблица 4

X1	X2	Y
0	0	0
1	0	0
0	1	0
1	1	1

ременной и одноразрядным двоичным числом. Следовательно, в процессоре допустимы арифметические операции над логическими величинами. Кроме того, их можно сравнивать.

Разработаем схему, реализующую уравнение:

$$Y = X1 > X2.$$

Начнем с составления таблицы истинности (табл. 5).

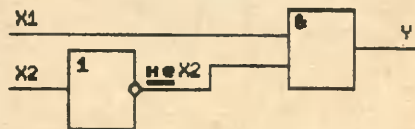
Таблица 5

X1	X2	Y
0	0	0
1	0	1
0	1	0
1	1	0

Видно, что таблица истинности определяется уравнением:

$$Y = X1 \text{ и не } X2.$$

Теперь легко изобразить соответствующую схему (рис. 3).



Каждая схема с логической точки зрения представляет собой выражение. Различные схемы, дающие при одинаковых входных сигналах одинаковые результаты, соответствуют равным логическим выражениям. Например, мы убедились, что:

$$X1 > X2 = X1 \text{ и не } X2.$$

Равенство логических выражений определяется основными законами алгебры логики и следствиями из них.

Если учитель разделяет наше убеждение, что законы логики имеют некоторое общеобразовательное значение, то на работе вентилях и их комбинаций ему следует остановиться подробнее, чем это сделано в учебнике. Чтобы не вынуждать школьников зубрить логические формулы, можно подготовить плакат (табл. 6).

Для удобства разработчиков выпускаются микросхемы, реализующие сложные логические функции, в том числе:

Принятые обозначения можно объяснить. Значок «&» (английское and) используется для записи операции и в логических выражениях. Единица в вентиле «или» — это остаток от устаревшего обозначения « ≥ 1 » (легко убедиться, что этот вентиль действует именно так).

Для пояснения логики работы вентилях можно воспользоваться таблицами истинности. На рис. 2 приведены таблицы истинности трех основных вентилях, реализующих три основные логические операции: не (инвертор), или (дизъюнктор), и (конъюнктор). Заметим, что элемент «и» работает с сигналами «1» так же, как элемент «или» с «0».

При изучении первой главы мы убедились, что любые сложные логические выражения можно свести к комбинации трех логических операций. Следовательно, перечисленные три вентиля обладают функциональной полнотой, т. е. этих трех элементов достаточно для составления любых логических схем.

Так как на машинном языке величины логического типа кодируются нулевым или единичным битом, то ЭВМ «не замечает» разницу между логической пе-

«и-не» («элемент Шеффера»), «или-не» («элемент Пирса»), сумматоры, триггеры, счетчики, шифраторы и т. д. Логическое устройство большинства из них мы рассмотрим чуть позже. Важно, что одно и то же логическое выражение может быть реализовано на различных схемах, т. е. набор микросхем обладает функциональной избыточностью.

и выдает нулевой сигнал в обратном случае (иногда схемой сравнения называют инверсный вариант).

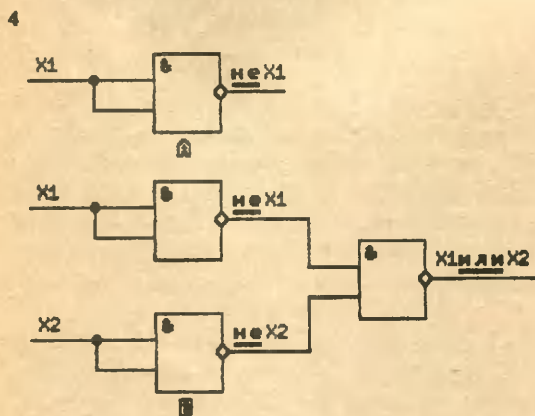
На рис. 5, б приведено стандартное обозначение схемы сравнения, а на рис. 5, в работа схемы сравнения иллюстрируется на примере регулирования потоков некоторой жидкости (один из наглядных способов иллюстрации) при

Таблица 6

ОСНОВНЫЕ ЗАКОНЫ АЛГЕБРЫ ЛОГИКИ		
Закон	Для "или"	Для "и"
переместительный (коммутативный)	$x \text{ или } y = y \text{ или } x$	$x \text{ и } y = y \text{ и } x$
сочетательный (ассоциативный)	$x \text{ или } (y \text{ или } z) = (x \text{ или } y) \text{ или } z$	$(x \text{ и } y) \text{ и } z = x \text{ и } (y \text{ и } z)$
распределительный (дистрибутивный)	$(x \text{ или } y) \text{ и } z = x \text{ и } z \text{ или } y \text{ и } z$	$x \text{ и } y \text{ или } z = (x \text{ или } z) \text{ и } (y \text{ или } z)$
инверсии (формулы де Моргана)	$\text{не } (x \text{ или } y) = \text{не } x \text{ и } \text{не } y$	$\text{не } (x \text{ и } y) = \text{не } x \text{ или } \text{не } y$

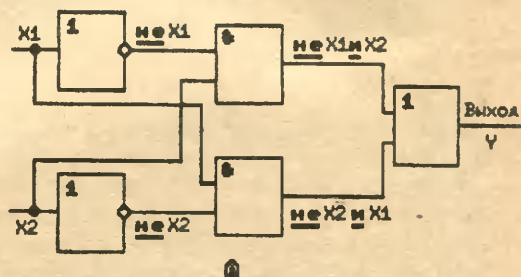
30

Например, можно строить любые схемы только из вентилях «и-не». Из них легко собирается схема «не» и схема «или» (рис. 4). На рис. 4, б реализована одна из формул де Моргана.

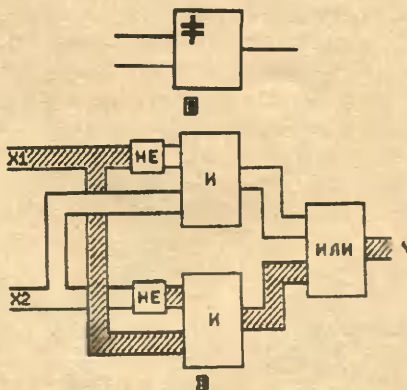


Любое на первый взгляд сложное, почти «разумное» действие ЭВМ можно разбить на элементарные «микродействия». Рассмотрим устройство некоторых логических элементов, выполняющих эти «микродействия».

Начнем с так называемой схемы сравнения (рис. 5), которая вырабатывает единичный сигнал на выходе, если входные сигналы не равны между собой,



5



различных входных сигналах.

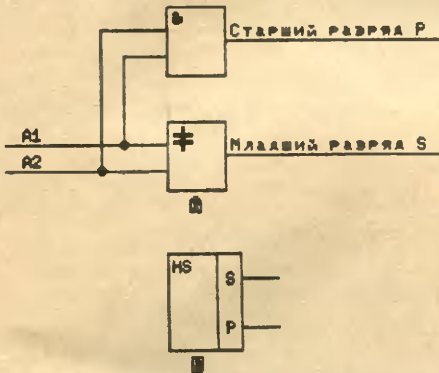
Заметим, что схема сравнения дает на выходе младший разряд суммы двоич-

ных чисел, поданных на входы:

$$\begin{aligned} 0+0 &= 0 \\ 0+1 &= 1 \\ 1+0 &= 1 \\ 1+1 &= 10 \end{aligned}$$

Используя элемент сравнения, можно построить схему, определяющую сумму двух двоичных одноразрядных чисел A1 и A2 (рис. 6, а). Такая схема позволяет складывать два бита. Она называется полусумматором. Обозначение полусумматора приведено на рис. 6, б.

6



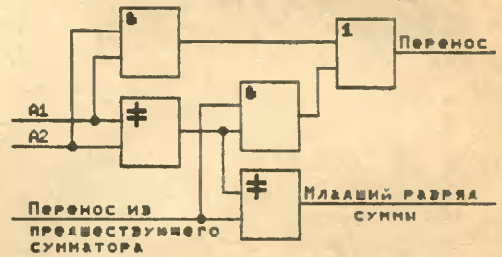
Современные ЭВМ оперируют восьми-, шестнадцати-, тридцатидвух- и шестидесятичетырехразрядными числами. Чтобы понять, как складываются в ЭВМ многоразрядные числа, попробуем сначала «вручную» (на доске, в тетрадь) сложить два восьмиразрядных числа, проделав это по шагам, начиная с младших разрядов. При этом воспользуемся привычным алгоритмом сложения в столбик. Только вместо «один в уме» будем писать «перенос равен 1» (см. схему).

шаг 1	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 1110001 \end{array}$	(0+1=1, перенос равен 0)
шаг 2	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 01 \end{array}$	(1+0=1, перенос равен 1)
шаг 3	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 001 \end{array}$	(1+0+1=10, перенос равен 1)
шаг 4	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 0001 \end{array}$	(0+1+1=10, перенос равен 1)
шаг 5	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 10001 \end{array}$	(1+1+1=11, перенос равен 1)
шаг 6	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 110001 \end{array}$	(0+0+1=1, перенос равен 0)

шаг 7	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 1110001 \end{array}$	(0+1+0=1, перенос равен 0)
шаг 8	$\begin{array}{r} 10010110 \\ + 01011011 \\ \hline 11110001 \end{array}$	(1+0+0=1, перенос равен 0)

Для построения схемы, складывающей многоразрядные числа, потребуется каскад, составленный из полных сумматоров. Каждый полный сумматор делает один шаг «сложения в столбик», т. е. выполняет сложение двух очередных разрядов с учетом переноса (рис. 7).

7



31

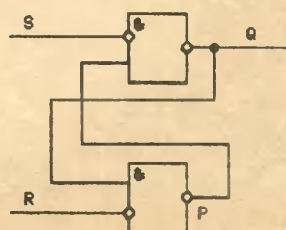
Таким образом, полный сумматор должен складывать три отдельных бита, поэтому в его состав войдут два полусумматора (отсюда префикс «полу-»).

Можно предложить учащимся заполнить таблицу истинности для всех возможных комбинаций трех входных сигналов.

С помощью сумматора можно выполнять любые арифметические операции: сложение, вычитание (сводится к сложению уменьшаемого с числом, обратным вычитаемому), умножение (многократное сложение), деление (многократное вычитание) и т. д.

Для осуществления различных операций процессор обладает небольшой собственной памятью, каждая ячейка которой называется регистром. Чтобы понять принцип работы регистра, рассмотрим RS-триггер (об инверсном выходе

8



R для упрощения умалчиваем). Схема RS-триггера, построенного на вентилях «и-не», приведена на рис. 8. Можно также воспользоваться вентилями «или-не». Попробуйте предложить учащимся составить такую схему самостоятельно.

Составляя таблицу истинности триггера (см. табл. 7), учтем, что его выход является входом в следующий момент времени. Поэтому введем такие обозначения:

Входные сигналы $\left\{ \begin{array}{l} S — установка единицы (set); \\ R — установка нуля (reset); \\ Q_{стар} — выход в предыдущий момент времени. \end{array} \right.$
Выходной сигнал — Q.

Шифратор (рис. 10) преобразует входной сигнал, обозначающий десятичное число от 0 до 7 (например, нажата соответствующая кнопка), в трехразрядное двоичное число. Одновременное нажатие двух кнопок запрещено.

Для того чтобы понять устройство шифратора, разработаем схему шифратора с двумя выходами. Сколько входов будет иметь схема? Понятно, что число входов равно 2^2 , т. е. четырем. Обозначив входы через X_i , а выходы — через Y_i , составим таблицу истинности (табл. 8).

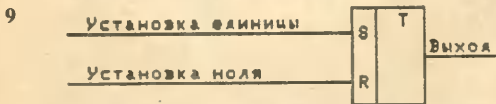
По таблице истинности легко составить принципиальную схему шифратора.

Таблица 7

S	R	Q _{стар}	Q	Примечания
0	0	1	1	Сохраняется предыдущее значение выхода
0	0	0	0	
1	0	0	1	При установке единицы предыдущее состояние не играет роли
1	0	1	1	
0	1	0	0	При установке нуля предыдущее состояние не играет роли
0	1	1	0	
1	1	0	?	Запрещенное состояние входов (S=R=1)
1	1	1	?	

32

Условное обозначение RS-триггера изображено на рис. 9.



В процессоре используются и другие типы триггеров, но они не слишком отличаются от RS-триггера, поэтому не будем их рассматривать.

Чтобы связать процессор с другими устройствами, потребуются различные преобразователи. Сделаем схемы некоторых из них.

Таблица 8

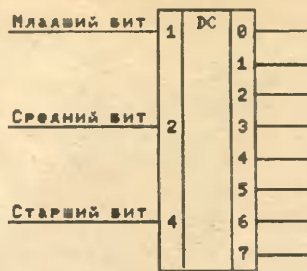
Десятичное число				Двоичное число	
0				00	
1				01	
2				10	
3				11	
X1	X2	X3	X4	Y1	Y2
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Итак, связь с устройствами ввода установлена.

Рассмотрим теперь обозначение дешифратора (рис. 11), преобразующего трехзначное двоичное число в сигнал, соответствующий десятичному числу от 0 до 7. Затем составим схему дешифратора с двумя входами, используя уже известные вентили (решения не приводятся). Появилась связь с устройствами вывода.

10





Изученных элементов вполне достаточно, чтобы построить принципиальную схему процессора (мы фактически уже разработали фрагменты одноразрядного процессора).

Другая важнейшая часть ЭВМ — это память. Поэтому перейдем к изучению логических основ устройства машинной памяти.

Продолжение следует.

Я. МАРГОЛИС, А. ИВАНОВ, Э. БАРАНКИНА
г. Куйбышев

Содержание и методы непрерывного обучения информатике

33

Возраст, с которого дети начинают изучать информатику, неуклонно снижается. Об этом свидетельствует как зарубежный [1—5], так и отечественный опыт [6, 7], показывающий, что школьники младшего и среднего возраста хорошо усваивают основы информатики. В этой связи актуальными становятся принципы построения учебных программ непрерывного обучения информатике с I по XI класс.

Обзор научных работ свидетельствует о том, что содержание первых учебных программ было ориентировано в основном на обучение школьников программированию. В дальнейшем появились различные исполнители, которые берут свое начало от «Черепахи» С. Пейперта [1]. Развитие этих идей привело к осознанию необходимости формирования компьютерной культуры, включающей в себя компьютерную грамотность: умение с помощью компьютера писать, считать, рисовать, работать с другими видами информации. Вместе с тем зарубежный опыт показывает, что использование компьютера даже как инструментального средства обработки различных видов информации не дает должных практических результатов обучения.

Наша концепция использования компьютеров в обучении ориентирована на творчество, самостоятельность и практическую работу школьников в раз-

личных сферах человеческой деятельности. Собственно изучение информатики и компьютера как объекта является основой или фоном, на котором происходит обучение использованию компьютеров.

По содержанию практического применения компьютера в зависимости от возраста учащихся мы выделяем пять этапов (см. таблицу).

Естественно, что возрастное деление на этапы не строгое, а достаточно условное и границы этапов могут перекрывать друг друга. Рассмотрим каждый из них подробнее.

Подготовка к систематическим занятиям по информатике начинается в возрасте 4—5 лет в детском саду на базе использования компьютерно-игрового комплекса (КИК) для дошкольников, так как игра является основной средой обучения детей этого возраста. КИК «Электроник» позволяет создать условия подготовки детей с помощью пакета игровых программ «Знайка» к дальнейшему использованию вычислительной техники в обучении 6—9-летних детей началам информатики [8, 9].

Цели и задачи обучения на этапе 6—9 лет таковы:

использовать вычислительную технику для развития различных видов мышления;

дать представление детям об универсальных возможностях использования

**Структура, содержание и программная поддержка курса непрерывного обучения информатике
в зависимости от возраста учащихся**

№ этапа обучения и возраст учащихся	Цель обучения	Содержание обучения	Программная поддержка курса
I этап, 4—5 лет	Первоначальное знакомство с компьютером	Развивающие компьютерные игры	Пакет программ «Знайка»
II этап, 6—9 лет	Компьютер — средство развития мышления учащихся	Задачи на развитие типов мышления	«Роботландия» «ЛОГО», «LOGOWRITER» и др.
III этап, 10—12 лет	Компьютер — инструмент проектирования, конструирования и управления созданными механизмами	Задачи конструирования и управления объектами	Программная поддержка конструкторов «ЛЕГО» и «ФИШЕР»
IV этап, 13—15 лет	Компьютер — средство исследования природных явлений	Задачи измерения и анализа физических, химических, биологических величин	Программная поддержка «Персональной научной лаборатории» (PSL)
V этап, 16—17 лет	Компьютер — инструмент в сфере делового управления	Задачи автоматизированной обработки информации и телекоммуникаций	Текстовый редактор, электронная таблица, СУБД, электронная почта

34

компьютера как средства обучения, проектирования, музицирования, изображения, вычисления, редактирования, развлечения и др.;

вызвать интерес и создать положительное эмоциональное отношение детей к вычислительной технике;

сформировать у детей знания, умения и навыки работы с компьютером.

Развитие различных видов мышления на втором этапе происходит с помощью безмашинного решения задач: игра Баше, задача о фальшивой монете, задача переливания из сосуда в сосуд, Ханойская башня, перевозчик, волк, коза и капуста и др. Безмашинный вариант необходим для детального разбора алгоритма решения задачи, и в дальнейшем он реализуется на компьютерах. Такие задачи, как Ханойская башня, игра Баше и другие, проходят в своем усвоении целый ряд этапов от эмпирического алгоритма решения (когда ребенок может решать задачу, зная алгоритм, только при данных конкретных условиях) до теоретического обобщенного алгоритма (когда ребенок может решать задачу при любом изменении начальных условий). Наиболее сложным на этом этапе обучения является подбор задач на развитие абстрактного, пространственного, операционного, ассоциативного и образного видов мышления.

В соответствии с практической направ-

ленностью нашей концепции и учетом зарубежного опыта построения учебного курса по информатике [10] обучающие программы делятся условно на следующие группы, подчеркивающие универсальность использования компьютера: знакомство с компьютером; графика; действия с числами; музыка; работа с текстом; поддержка курсов обучения по родному языку, математике, изобразительному искусству, музыке, трудовому обучению.

Третий этап обучения начинается в возрасте 10—12 лет с управления компьютерными исполнителями. Это могут быть «Черепашка» (язык ЛОГО и его развитие LOGOWRITER [11]), «Таракан» (Роботландия), «Муравей» (Школьница) и др. Управление компьютерными исполнителями не является самоцелью. Оно необходимо для перехода к управлению реальными исполнителями, сконструированными самими учащимися. Такими исполнителями могут быть роботы, автомобили, светофоры, станки, плоттеры и другие управляемые с помощью компьютера механизмы (объекты).

Проектирование и конструирование основываются на работе с конструкторами типа «ЛЕГО» и «ФИШЕР», имеющими в своем составе набор деталей и устройств, необходимых для созда-

ния различных механизмов и машин, а также управление ими с помощью IBM — совместимого компьютера [12]. Аналогичные работы ведутся в Болгарии (ПГО БАН) и в нашей стране (проект «Школа»).

Целью программы по информатике в среднем звене (13—15 лет) является использование компьютера в исследовании природных явлений с помощью компьютерной персональной научной лаборатории (PSL) [13]. Возможности PSL определяются набором датчиков, которые входят в нее, позволяя исследовать механические свойства объектов, например жесткость пружин, кислотность растворов pH в диапазоне от 0 pH до 12 pH, температуру в диапазоне от -40°C до $+105^{\circ}\text{C}$ с точностью до одного градуса, освещенность с помощью фотометрических датчиков и другие физико-химические величины.

Использование на уроках физики, химии, биологии PSL способствует формированию исследовательских навыков и является активной формой обучения, так как дети сами добывают знания.

Наиболее актуальным применением компьютерной персональной научной лаборатории является, на наш взгляд, использование ее для решения экологических задач. Примерами могут служить задачи определения содержания нитратов в овощах, собранных детьми в подшефном совхозе, загрязненности водного и воздушного бассейнов реки Волги, кислотности дождей, а также задачи популяционной экологии.

Аналогичной активной формой получения знаний в математике является система «Планиметрия», разработанная вычислительной лабораторией Софийского университета [14]. Она позволяет учащимся выдвигать гипотезу, проводить математический эксперимент с помощью компьютера, анализировать результаты и самим формулировать теоремы, затем доказывая их.

PSL и системы типа «Планиметрия» являются примерами активного получения знаний с помощью новых информационных технологий обучения и показывают учащимся этого возраста роль компьютера как инструмента практической деятельности.

В этом возрасте происходит углубление знаний по инструментальным средствам и наряду с использованием редакторов, появляются работы с учебными базами данных, электронными таблицами, что является подготовкой к использованию компьютера в старших классах.

Для старшеклассников (V этап) целью обучения информатике является использование компьютера в качестве инструмента практической деятельности, в частности в сфере делового управления.

За годы предыдущего обучения учащиеся подготовлены к использованию компьютера в сфере управления организационными системами. На этом этапе обучения учащиеся должны усвоить основные понятия кибернетики, такие, как объект управления, входная и выходная информация, система обратной связи, которая эффективно реализуется с помощью компьютера. Использование компьютера в автоматизации управления предусматривает изучение административно-справочной деятельности (электронные картотека, секретарь, почта), автоматизации административно-управленческой деятельности (автоматизация плановых расчетов, компьютерные системы помощи в управленческой деятельности, математическое и имитационное моделирование), основ создания автоматизированных систем управления на примере своей школы, автоматизации административно-контрольной и административно-отчетной деятельности (задача контроля исполнения приказов и распоряжений, генераторы отчетов), деловой графики (статистико-аналитическое изображение результатов).

Использование компьютеров в системе телекоммуникации является важным элементом в ориентации учащихся на практическую деятельность. Работа в локальной сети комплектов учебной вычислительной техники (КУВТов) на первом этапе позволяет осознать учащимся возможности обмена информацией между разными компьютерами в пределах класса. На следующем этапе связь с помощью телефонного канала позволяет расширить возможности использования компьютера в сфере делового

вого управления и показать учащимся пользу коммуникации в управлении, на примерах электронной почты, сбора различных данных о погоде, кислотности дождей и т. п. Учащиеся также должны иметь представление о возможностях обмена компьютерной информацией через спутники, принимая участие в проведении международных школьных исследований в рамках проекта «Школа» АН СССР (STAR SCHOOL PROJECT) и других аналогичных работах [15].

Коротко расскажем, как осуществляется преподавание информатики с I (6 лет) класса в СШ № 63 г. Куйбышева.

36 Наш опыт показал, что ориентация на самостоятельность, творчество и практическую направленность работ может реализовываться, например, путем выпуска детьми в конце учебного года компьютерного журнала. Этот журнал содержит результаты работы каждого учащегося, выполненные с помощью компьютера. В него входят стихи и проза, написанные на свободную тему, компьютерные картинки, ее иллюстрирующие, а также музыкальные фрагменты в виде распечаток



партитуры на эту тему. Выбор собственной темы, подготовка сценария (включающего текст, картинку и музыкальный фрагмент) позволяют учащимся уже в раннем возрасте осознать, что компьютер не может заменить человека в его творчестве, но является его помощником.

Практическая работа в течение учебного года проводилась на КУВТ «Ямаха». Например, детьми набиралась мелодия «В лесу родилась елочка» и распечатывалась ее партитура для школьного

новогоднего вечера; готовились заметки для классной стенной газеты с помощью текстового редактора; поздравительные открытки для мам на 8 Марта; набиралась и исполнялась на компьютере популярная мелодия «Ламбада» и т. п.



Компьютерный журнал, содержащий работы учащихся, позволяет им увидеть результаты своего труда. Наличие такого журнала в школьной библиотеке является стимулом творческой деятельности детей и служит повышению интереса к учебе. Для учащихся старшего возраста компьютерный журнал содержит результаты их собственных исследований по литературе и истории (рефераты, обзоры), в области точных наук (результаты экспериментов). Результаты таких работ обсуждались учащимися, педагогами школы и вузов города на итоговой общешкольной конференции. Лучшие выступления учащихся были записаны на видеокассету, что положило начало созданию школьного видеожурнала по информатике. Создание таких журналов способствует формированию и сохранению традиций школы и передач их от поколения к поколению учащихся.

Результаты работы, полученные на компьютере, могут быть промежуточным этапом при изготовлении детьми каких-либо предметов. Подобный вид деятельности обязательно должен заканчиваться изготовлением детьми своими руками этих предметов: города, построенного из кубиков, предварительно спроектированного на компьютере; аппликации, нарисованной на дисплее и вырезанной учениками из разноцветной бумаги; глиняных горшков, форма которых от-

работана на компьютере, изготовленных в гончарной мастерской; шкатулки, вазы, орнаменты которых подобраны на дисплее и затем выпилены учащимися из фанеры. Компьютерный журнал, о котором говорилось выше, переплетается самими учащимися на уроках труда.

Такая деятельность позволяет учащимся совместить работу «рук и головы», что очень важно для их практической работы в будущем.

Одним из элементов реализации нашей концепции развития творчества и самостоятельности является создание учащимися (совместно с учителями) компьютерных программ по различным учебным предметам для КУВТ «Корвет». Хотелось бы подчеркнуть, что реализовывались такие темы школьного курса, которые сложно изучать без компьютера, например газовые законы, атмосферные явления, исследование функций, понятие алгоритма и т. п. Для учащихся создание демонстрационных, моделирующих, обучающих, контролирующих и других программ является результатом их творческого отчета на итоговой школьной конференции.

Разработка таких программ для учащихся, увлеченных программированием, является еще одним способом более глубокого изучения выбранного ими учебного предмета.

Школьную IBM PC/AT предполагается использовать для конструкторов «ЛЕГО» и «ФИЩЕР» и для компьютерной персональной научной лаборатории.

Наличие модема позволяет включить школьную IBM PC в систему телекоммуникации проекта «Школа» АН СССР. Это даст возможность учащимся нашей школы проводить телеконференции, обмениваться электронными письмами с ребятами из других городов.

АСУ «Школа», созданная на базе IBM PC/AT, включает в себя подсистемы кадров учителей и учеников, контроль исполнения приказов и распоряжений, составления расписания и т. п. При этом подготовка базы данных и ее ведение осуществляются учащимися старших классов.

К сожалению, наличие одной, даже

мощной, IBM PC/AT не может удовлетворить потребности школы в реализации указанных выше направлений обучения.

В заключение отметим важность психолого-педагогического обеспечения и оценки эффективности обучения.

Наш опыт показывает, что дети плохо воспринимают программы, связанные с цейтнотом времени на принятие решения или ответа, поэтому необходимо использовать программы, дающие возможность ребенку свободно размышлять. Это существенный психологический фактор, влияющий на качество работы детей за компьютером.

Следующим важным моментом является индивидуальная работа 6—7-летних детей за компьютером. Дети этого возраста, на наш взгляд, с трудом работают парами за одним дисплеем. Психология их такова, что каждый борется за право первым нажать на клавишу. Поэтому в этом возрасте работать на компьютере дети должны индивидуально, однако это не исключает совместной творческой работы группы детей, особенно на этапе изготовления предметов, что способствует активизации вербального взаимодействия учащихся.

Оценка эффективности обучения проводилась по тестам, разработанным в Институте общей и педагогической психологии АПН СССР [16].

В результате непрерывного обучения информатике выпускник школы должен уметь использовать компьютер как инструмент практической деятельности при проектировании, конструировании, управлении машинами и механизмами, исследовании природных явлений, а также в сфере делового управления и телекоммуникации. Овладение этими навыками и умениями позволяет выпускать из школы учащихся, обладающих компьютерной грамотностью и способных к восприятию информационной культуры.

Литература

1. *Peunept С.* Переворот в сознании: Дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989.
2. *Николов Р., Сендова Е.* Начала информатики. Язык ЛОГО. М.: Наука, 1989.
3. *Wiederhold K., Mitzlaff H.* Computerein-

satz in der Grundschule? // Bildung und Computer, Hagen und Dortmund, Heft 2. Dezember. 1989.

4. Хантер Б. Мои дети работают на компьютерах. М.: Просвещение, 1989.

5. Нокс Дж. Что могут дать компьютеры педагогике: взгляд из американской школы // Информатика и образование. 1990. № 1.

6. Бешенков С. Экспериментальная программа преподавания начал информатики в младших классах // Информатика и образование. 1987. № 5.

7. Дуванов А., Зайдельман Я., Первин Ю., Гольцман М. Роботландия — курс информатики для младших школьников // Информатика и образование. 1989. № 5.

8. Компьютерно-игровой комплекс для дошкольников. ЦНИИ «Электроника», НИИ дошкольного воспитания АПН СССР, 1989.

9. Иорки Т. ЭВМ в детском саду //

Перспективы: Вопросы образования. ЮНЕСКО. 1987. № 4.

10. Примерны програми за обучение по информетика в специализиранте младежки клубове «Компютър» // А. А. Марчев, Н. В. Марчева. София, 1981.

11. LogoWriter // Reference Guide, Logo Computer System Inc., 1986.

12. Ocko S., Papert S., Resnik M. LEGO, Logo and Science // Technology and Learning. 1989.

13. Personale Science Laboratory. Explore the questions to learn the answers. IBM., 1989.

14. Система «Планиметрия». София, 1989.

15. Зайцев А. Спутники, компьютеры, образование // Информатика и образование. 1990. № 1.

16. Зак А. З. Развитие теоретического мышления младших школьников. М.: Педагогика, 1984.

В. ГУЗЕЕВ

Работа группами с компьютерной поддержкой

Организации работы класса в кабинете ВТ, когда машин в 2,5—3 раза меньше числа учеников, пока уделяется недостаточно внимания. Эта проблема весьма актуальна, поскольку ожидать, что в обозримом будущем число учеников в наших классах снизится до 12—15, не приходится. Но может быть, 12 машин на 30 человек достаточно? Именно об этом мы и хотим поговорить в этой статье. Итак, групповая работа с компьютером. Какие же проблемы здесь возникают?

Эффективные способы организации компьютерных учебных групп сегодня только начинают прорабатываться. О их формировании известно немного: оптимальный состав — от трех до семи человек; наличие «генератора идей», «скептика», «эрудита»; взаимная дополнителность личностных черт, наличие лидера. Кроме того, известны факторы, отрицательно влияющие на эффективность работы группы: негативное отношение к самой групповой работе, неадекватность задачи, особенности отношений между участниками, трудности включения в интеллектуальную групповую деятельность.

Работа группы имеет два аспекта: объективный (эффективность решения задачи) и субъективный (продуктивность деятельности каждого ученика для его индивидуального обучения и развития). Однако все сказанное практически ничего не говорит об эффек-

тивности работы в группе для каждого из партнеров, что больше всего интересует педагога.

Далее, далеко не каждая задача пригодна для группового решения. Проведенные психологами эксперименты показывают, что совместное решение имело место в задачах, требующих общего мнения, логического рассуждения, а в переборных или задачах на «понимание» продуктивного диалога не получалось.

Требуются серьезные психологические и педагогические исследования, способные дать рекомендации по подбору учеников в группы и отбору задач для таких групп, нацеленных на индивидуальное развитие личности при коллективной групповой форме деятельности.

В качестве компьютерной поддержки урока в условиях групповой работы целесообразно использование имитационных, моделирующих программ, программ-исполнителей, баз данных. Типичный пример таких программ — болгарская система «Планиметрия», реализованная на языке ЛОГО.

Система является программой-исполнителем и представляет пользователю возможности экспериментировать с конкретными геометрическими объектами, выдвигать гипотезы, пытаться построить синтетические доказательства. К сожалению, работа с этой

системой требует от учеников умения программировать. Сходным по назначению и не требующим умения программировать является пакет программ «Мотор». Он представляет ученику средства для исследования некоторого класса функций и применения их к решению задач. Мощным инструментальным средством является решатель задач Solver, который одновременно может использоваться для проведения инженерных расчетов. Одним из примеров применения экспертных систем и баз данных можно считать систему Zgeom — тренажер для обучения логике решения геометрических задач.

Как включить все это в ткань урока?

В качестве примера возьмем урок математики «Функция $y=\sqrt{x}$ » с применением ЭВМ, описанный болгарскими авторами В. Стефановой и М. Русановой. В болгарском варианте урока компьютер применяется только для фронтальной работы как демонстрационное средство. Мы же включили в него групповую работу, составив силами старшеклассников аналогичные программы.

Проведенный урок выглядит так.

Две группы (по три человека) работают с компьютером, одна (четыре человека) без него. Вспомнив все вместе свойства квадратного корня, учащиеся переходят к геометрическим представлениям: если b — сторона квадрата, S — его площадь, то зависимость $S(b)$ выражается как $S=b^2$. Формулы $S=b^2$ и $b=\sqrt{S}$ задают две функции. В привычных обозначениях — $y=x^2$ и $y=\sqrt{x}$. При $x \geq 0$ $y=x^2$ — ветвь параболы. Она демонстрируется на большом видеомониторе (цветном учительском).

Эмпирическим путем учащиеся выясняют простейшие свойства функции $y=\sqrt{x}$. Для выяснения закономерности изменения функции при возрастании аргумента учащиеся, используя ЭВМ, строят таблицу корней из целых от (0 до 8) и дробных (120, 120.5, ..., 161, 161.5) чисел. Анализируя полученную таблицу, доказывают возрастание функции. После нескольких упражнений на сравнение числа строится график функции $y=\sqrt{x}$. На демонстрационном мониторе теперь оба графика и проводится луч $y=x$ для подготовки изучения свойств обратных функций.

Далее приступают к работе группы. Одна исследует закон пути при равноускоренном

движении $S = \frac{at^2}{2}$ (отсюда $t = \sqrt{\frac{2S}{a}} = \sqrt{\frac{2}{a}} \cdot \sqrt{S}$). Рассматриваются случаи $\frac{2}{a} = 1$, $\frac{2}{a} > 1$, $\frac{2}{a} < 1$. При разных значениях a ($a > 0$) компьютер строит в одной сис-

теме координат графики $t = k\sqrt{S}$, где $k = \sqrt{\frac{2}{a}}$. По окончании работы группы экран

их компьютера отображается на демонстрационный монитор, чтобы все ученики в классе представляли вид графика при различных значениях k и могли его построить.

Другая группа исследует в это же время изменение концентраций веществ и полученного продукта в реакции вида $A+B \rightarrow AB$ в зависимости от времени и строит с помощью компьютера три графика. Один из них — график концентрации продукта — получается из графика $c = \sqrt{t}$ простым преобразованием. Результаты исследования группа доложит в конце урока.

Третья группа, получив отпечатанный на принтере график $y = \sqrt{x}$, строит в тетрадах графики $y = \sqrt{-x}$ при $x \leq 0$, $y = -\sqrt{x}$ при $x \geq 0$, $y = -\sqrt{-x}$ при $x \leq 0$. По окончании их работы эти графики для сравнения будут показаны с учительского монитора. Пока работают группы, учитель с оставшейся частью класса решает задачи на нахождение значений x по y и y по x с последующей компьютерной проверкой. В конце урока фронтально решаются задачи вида: «Принадлежат ли графику функции $y = \sqrt{x}$ точки $(-4; 2)$, $(2.25; 1.5)$, $(81; -9)$, $(-16; -4)$ и т. д.?»

Как видим, групповая работа с программным обеспечением подобного рода организационно укладывается в структуру семинара-практикума, не исключая работу с программами разных типов на одном уроке. Скажем, ученики, не достигшие минимального уровня, индивидуально работают с тренажерами. Группы общего и продвинутого уровней на свободных компьютерах решают задачи исследовательского характера. Часть учеников, хотя и объединены в группы, не работают с машинами на этом уроке.

Представляется, что в настоящее время семинар-практикум — наиболее приемлемая форма организации урока группами с компьютерной поддержкой.

Литература

1. Анисимов В., Соловьев Л. Учебный пакет прикладных программ «МОТОР» // Информатика и образование. 1988. № 3.
2. Атанасова С., Трендафилова Т. Компьютерно изследване на правата на Ойлер за четириъгълници с помощта на система «Планиметрия» // Обучение по математика и информатика. 1988. Кн. 6.
3. Гершунский Б. С. Компьютеризация в сфере образования: Проблемы и перспективы. М.: Педагогика, 1987.
4. Гузев В. В. Одна из форм урока-семинара // Математика в школе. 1987. № 2.

5. *Зенкин А., Зенкин Г.* Обучающая система-тренажер // Информатика и образование. 1988. № 5.

6. *Касък К., Яаксоо К.* Решатель задач Solver // Информатика и образование. 1988. № 4.

7. *Клейман Г. М.* Школы будущего: компьютеры в процессе обучения: Пер. с англ. М.: Радио и связь, 1987.

8. *Колчева М., Марчева К.* Этапы в обучении по математика със система «Планиметрия» // Обучението по математика и информатика. 1989. Кн. 3.

9. *Стефанова В., Русанова М.* Темата «Функцията $y=\sqrt{x}$ » с използване на персонален

компютър // Обучението по математика и информатика. 1989. Кн. 5.

10. *Шмелев А.* Детская болезнь компьютерного всеобуча // Информатика и образование. 1987. № 1.

11. *Щербо Н. П.* Особенности индивидуального и группового решения задач в условиях совместной деятельности // Вопросы психологии. 1984. № 2.

12. *Johnson D. W., Johnson R. T.* Cooperative Learning: One Key to Computer Assisted Learning // The Computing Teacher. 1985. October.

Хозрасчетный центр «Прогресс-Х» предлагает

Владельцам IBM-совместимых персональных ЭВМ

Описание языка FORTRAN 5.0 MS. Данная версия FORTRANa 1989 г. выгодно отличается от предыдущих расширенных набором операторов, позволяет создавать оптимальные командные коды, имеет мощную графическую библиотеку. Описание на русском языке. Поставка — на дискете с возможностью распечатки.

Пакет программ по бухгалтерскому учету и расчету заработной платы для предприятий и организаций.

Школам и владельцам персональных ЭВМ БК-0010

Пакет обучающих программ по математике для КУВТ-86 и БК-0010. Программы обеспечат 100 % занятость на уроке и повышенный интерес к предмету. Поставка — на кассетах.

Эти и многие другие программные продукты вы можете приобрести по безналичному и наличному расчету.

Мы приглашаем временные творческие коллективы и отдельных исполнителей для работ на договорной основе. Гарантируем заработную плату до 90 % от суммы договора.

Наш адрес: 119899, Москва, Ленинские горы, МГУ им. М. В. Ломоносова, комитет ВЛКСМ, молодежный экспериментальный центр «Прогресс-Х».

Телефон: 939-31-01 с 11 до 18 часов.

С. ГРИГОРЬЕВ

Обработка списков на Прологе-Д

На практике часто встречаются задачи, связанные с перечислением объектов. Для их описания используются списки. Пример — список учеников I класса:

[Саша, Петя, Дина, Кснша, Лена, Оля, Катя]

Элементами списка могут быть любые элементы, даже списки. Пример — состоящий из функций список остановок пригородного поезда с указанием времени стоянки:

[Казань (0), Юдино (2), Зеленый-Дол (5), Помары (30), Илеть (3), Шелангер (3), Суслонгер (5), Кундыш (2), Сурок (3), Пемба (2), Куяр (3), Йошкар-Ола (0)]

Примером списка, состоящего из списков, может служить прямоугольная таблица (матрица), представляющая собой список строк, каждая из которых есть список элементов в данной строке. Например, таблица

```
23 45 56 2 78 89 66 45
56 12 3 75 2 3 6 5
2 1 56 2 5 8 9 22
23 22 33 5 6 9 1 33
```

может быть представлена следующим списком списков:

```
[ [23, 45, 56, 2, 78, 89, 66, 45],
  [56, 12, 3, 75, 2, 3, 6, 5],
  [2, 1, 56, 2, 5, 8, 9, 22],
  [2, 1, 56, 2, 5, 8, 9, 22],
  [23, 22, 33, 5, 6, 9, 1, 33] ]
```

Во всех примерах квадратные скобки означают, что данный объект представляет собой список.

В системе Пролог-Д MSX список может быть определен двумя способами: перечислением элементов списка, т. е. так, как это

было сделано выше, и определением головы и хвоста списка.

Например, список [X:Y] определен именно таким путем. X — это голова списка, а Y — его хвост. На вопрос

? РАВНО(X, [a:[b,c,d]]);

будет получен ответ

X=[a,b,c,d]

Второй способ представления списка можно использовать для определения головы и хвоста. Например, с помощью вопроса

?равно([X..Y], [a,b,c]);

получим:

X=a
Y=[b,c]

Вторая форма представления списка в сочетании с рекурсией позволяет написать несколько полезных программ.

1. Принадлежность элемента списку.

А. Основной случай можно сформулировать следующим образом: терм Р принадлежит списку [X:Y], если P=X.

Б. Рекурсивная ситуация описывается с помощью высказывания: терм Р принадлежит списку [X..Y], если Р принадлежит Y.

На Прологе эти два высказывания можно записать в виде предложений

```
принадлежит(P,L) ← РАВНО(L, [X:Y]),
                    РАВНО(P,X);
принадлежит(P,L) ← РАВНО(L, [X:Y]),
                    принадлежит(P,Y);
```


Эту программу можно усовершенствовать, если вспомнить, что Пролог-Д унифицирует сначала голову правила, а затем его цели. Усовершенствованная программа будет выглядеть так:

```
элемент(F, [P|T]);
элемент(F, [X:Y]) ← элемент(F, Y);
```

2. «Склеивание» двух списков.

А. Основная ситуация состоит в том, что если к пустому списку [] добавить список P, то в результате получится P.

Б. Рекурсивная ситуация состоит в том, что можно список P добавить к концу списка С, если P будет добавлен к хвосту С и затем присоединен к голове В.

На Прологе-Д эти предикаты можно записать так:

```
добавить([], F, P);
добавить(C, P, B) ← равно(C, [X:Y]),
добавить(Y, P, Z), равно(B, [T:Z]);
```

42

Пользуясь вышеупомянутым свойством Пролога, эту программу тоже можно упростить:

```
присоединить([], P, P);
присоединить([X:Y], P,
[X:T]) ← присоединить(Y, P, T);
```

Например, если к этой базе знаний задан вопрос

```
?присоединить(L, [Дина:R], [Ваня,
Кеша, Дина, Тина, Дима, Боря]),
```

то будут получены ответы

```
L=[Ваня, Кеша]
R=[Тина, Дима, Боря]
L=[Ваня, Кеша, Дима, Тина]
R=[Боря]
```

Во входном языке системы Пролог-Д БК-0010 не предусмотрена синтаксическая конструкция «список». Синтаксис не допускает использования квадратных скобок. Следовательно, для применения списков в программах необходима имитация списков синтаксическими средствами, допустимыми во входном языке. Для имитации списка, заданного перечислением,

```
[<аргумент1>, <аргумент2>, ..., <аргументn>]
```

предлагается использование функциональных термов вида

```
ff(<аргумент1>, ff(<аргумент2>, ...,
ff(<аргументn>, пусто) ...)
```

а для имитации списка, заданного головой и хвостом,

```
[x:y]
```

использовать функциональный терм

```
ff(x, y)
```

Заметим, что списки в более старших версиях системы Пролог-Д записаны именно в таком виде.

Если определить в базе знаний предикат *равно*

```
равно(x, x);
```

и затем обратиться к системе с вопросом

```
?равно(X, ff(a, ff(b, ff(c, ff(d, пусто)))));
```

то будет получен ответ

```
X=ff(a, ff(b, ff(c, ff(d, пусто))))
```

Двойное представление списка можно использовать для определения головы и хвоста. Например, с помощью вопроса

```
?равно(ff(X, Y), ff(a, ff(b, ff(c, пусто))));
```

получим

```
X=a
Y=ff(b, ff(c, пусто))
```

Запишем на Прологе-Д БК-0010 решения приведенных выше задач.

1. Принадлежность элемента списку.

```
принадлежит(F, L) ← равно(L, ff(X, Y)),
равно(F, X);
принадлежит(F, L) ← равно(L, ff(X, Y)),
принадлежит(F, Y);
```

Усовершенствованная программа будет выглядеть так:

```
элемент(F, ff(F, T));
элемент(F, ff(X, Y)) ← элемент(F, Y);
```

2. «Склеивание» двух списков.

```
добавить(ff(пусто), ff(P, пусто),
ff(F, пусто));
добавить(C, P, B) ← равно(C, ff(X, Y)),
добавить(Y, P, Z), равно(B, ff(T, Z));
```

Эту программу тоже можно упростить:

```
присоединить(ff(пусто), ff(F, пусто),
ff(F, пусто));
присоединить(ff(X, Y), P,
ff(X, T)) ← присоединить(Y, P, T);
```

Упражнения.

1. Напишите базу знаний, описывающую сортировку списка, элементы которого суть целые, по возрастанию.

2. Напишите базу знаний, описывающую обращение списка (первый элемент становится последним).

3. Напишите базу знаний, описывающую нахождение элемента списка с номером N.

4. Напишите базу знаний, описывающую удаление N-го элемента списка.

УКНЦ: Бейсик с магнитофона

Представим *невероятное*: в классе УКНЦ вышел из строя дисковод, и нет способа оживить 12 ставших мертвыми машин.

Это не трагедия, если вы заранее подготовитесь к такой неожиданности и обеспечите возможность загрузки Бейсика с магнитофона.

Когда в главном меню загрузки выбирается режим «Загрузка с магнитофона», в ОЗУ центрального процессора размещается с адреса 1000 (здесь и далее все числа восьмеричные) и запускается программа, обеспечивающая загрузку (с адреса 0) первых 400 слов любой программы, записанной на магнитофон в формате УКНЦ. Управление последней передается (т. е. обнуляется счетчик адреса команд), только если ее первой командой будет NOP (код 000240).

Пусть это будет загрузчик, обеспечивающий загрузку самого Бейсика. Напишем его по аналогии с имеющейся программой (изменив лишь начальный адрес и длину загружаемого файла). Он может выглядеть, например, так.

```

012701 176674 110027 000403 105011 012702 000400 012703
000002 105711 100376 110261 000002 000302 077306 012703
000002 105711 100376 112761 000377 000002 077306 105711
100376 105727 000400 001403 105027 000400 000750 000240
000240 000240 012707 003356

```

Загрузчик передает в периферийный процессор приведенный ниже блок параметров, в соответствии с которым осуществляется считывание с МЛ соответствующей информации

и размещение ее в ОЗУ ЦП (см. «Программное обеспечение комплекса «Электроника МС 0202». 00031—01 90 01).

```

004777 157011 000000 000000 000000 000000 000000 000000
000000 000000 001000 064576

```

Последнее число в загрузчике — стартовый адрес, последнее число в блоке параметров — длина загружаемого файла.

Загрузчик и блок параметров размещаются в первых 1000 байтов ОЗУ ЦП. В этом же адресном пространстве должны располагаться и адреса прерываний. Чтобы не потерять их, необходимо сделать следующее: работая в Бейсике, с помощью операторов РОКЕ разместить с адреса 120 загрузчик, с 400 — блок параметров, а затем записать первые 1000 байтов на МЛ командой BSAVE. Таким образом на МЛ будет записан загрузчик.

Сразу после него на МЛ записывается Бейсик командой

```
BSAVE "CAS: BASIC", &01000, &065600
```

Верхний адрес действителен для Бейсика «Вильнюс 1987. 11. 26». В более поздних версиях возможно изменение этого адреса.

Если что-то не получается, обращайтесь по адресу: 644005 Омск, Добролюбова, 15, Омский индустриально-педагогический техникум, лаборатория вычислительной техники, Ю. Д. Плетнев. Тел. 41-04-17.

Пользователям КУВТ-86, версия Бейсик.

Территориальное межотраслевое производственно-техническое
управление Куйбышевского исполкома г. Москвы
ПРЕДЛАГАЕТ

вам пакет прикладных поурочных программ
по всему курсу «Информатика в X и XI классах».
Пакет разработан совместно с «ЭХТО» и насчитывает
свыше 100 программ на 11 магнитных дисках 5'25.
Стоимость пакета 1400 руб.

Гарантийные письма направлять по адресу:
105023, Москва, Суворовская, 27. ТМПТУ.

Химические шашки на ЭВМ

Во многих кабинетах химии средних школ и профессионально-технических училищ используется игра «Химические шашки». Ее правила легли в основу одноименной компьютерной программы по химии, реализованной на ДВК-2М и КУВТ «Ямаха MSX-2», предназначенной для изучения одной из основных тем курса неорганической химии для учащихся восьмых классов средней школы «Обобщение сведений о важнейших классах неорганических соединений». Работа на ЭВМ с данной программой позволяет учащимся в процессе игры пополнить и привести в систему свои знания.

Основу всякой игры составляют правила, определяющие множество допустимых ходов. В «Химических шашках» фигурами являются химические соединения: кислоты, соли, оксиды, вода. По правилам игры фигуру можно снять лишь тогда, когда вещества реагируют друг с другом. Далее будем называть это основным правилом.

В данной учебной программе правила игры в русские шашки дополнены некоторыми незначительными ограничениями, которые направлены на упрощение записи хода и достижение учебных целей. В других разработках эти ограничения могут быть исключены.

Обучение осуществляется в процессе игры учащегося с компьютером в шашки. На экране дисплея изображается игровое поле с фигурами. Учащийся делает свой ход первым. На экране высвечивается новая позиция, затем ЭВМ делает ход и комментирует его и т. д. При совершении хода учащимся возможны следующие ситуации.

Передвижение фигуры. При этом обычная шашка может передвигаться на свободную позицию только вперед, а дамка может двигаться вперед и назад.

Ошибочная запись хода. Ситуация возможна, если учащийся недостаточно хорошо усвоил правила игры и записи ходов. Компьютер указывает на допущенную ошибку и предлагает совершить другой ход.

Ошибочное снятие фигур. Учащийся совершил ход, выполнение которого требует снятия фигур противника с нарушением основного правила. Ситуация возникает вследствие незнания некоторых свойств веществ. Компьютер высвечивает вещества, которые, по предположению учащегося, должны вступить в реакцию, и сообщает, незнание каких свойств или правил привело к ошибочным действиям. После этого компьютер предлагает сделать другой ход.

Нарушение правила «боя». Ситуация может возникнуть, если имеется возможность снять фигуры противника, но учащийся этого не заметил и записал ход без «боя» или если учащийся снял несколько фигур и не заметил, что в данном направлении можно еще снимать фигуры противника. Компьютер расценивает такую ситуацию как незнание свойств веществ и правил, по которым они вступают в реакцию, осуществляет снятие фигур и высвечивает соответствующие уравнения химических реакций.

Снятие фигур. Сделан ход, приводящий к снятию фигур противника. Фигуры снимаются по очереди в направлении, заданном ходом. Перед снятием очередной фигуры высвечивается уравнение соответствующей химической реакции.

Если машина снимает фигуры учащегося, правильность ее действий подтверждается записью уравнений химических реакций на экране. Информационное обеспечение игры содержит два набора химических соединений. Первый набор (HCl , H_2S , KOH , $\text{Cu}(\text{OH})_2$, NaCl , K_2CO_3 , CuCl_2 , CO_2 , CaO , K_2O , Fe , H_2CO_3) используется для обозначения белых фигур, а второй ($\text{Ca}(\text{OH})_2$, HNO_3 , H_2SO_4 , AgNO_3 , MgSO_4 , Mg , NaCl , MgO , SO_3 , P_2O_5 , NaOH , $\text{Mg}(\text{OH})_2$) обозначает черные фигуры. Чтобы можно было отличить белые фигуры от черных, последние спереди и сзади помечаются звездочками. Например, запись $*\text{H}_2\text{SO}_4*$, $*\text{P}_2\text{O}_5*$ указывает, что данные соединения принадлежат черным фигурам. Учащийся всегда играет белыми фигурами и делает ход первым. Расположение химических соединений на шашечной доске в каждой партии задается с помощью датчика случайных чисел, и вероятность появления двух одинаковых начальных положений очень мала. Практически это означает, что каждая партия неповторима. Таким образом исключается возможность быстрой адаптации учащегося к игре, поддерживается постоянное желание играть с машиной и побеждать ее и, что самое важное, активизируются знания по химии в достаточной широкой предметной области.

Отношения и связи между наборами химических соединений из информационного обеспечения игры отражены в табл. 1. Плюсы, стоящие на пересечении строк и столбцов табл. 1, означают, что принадлежащие данным строкам и столбцам вещества взаимодействуют друг с другом. Если же на пересечении строки и столбца стоит целое

Таблица 1

	NaOH	Ca(OH) ₂	HNO ₃	H ₂ SO ₄	AgNO ₃	MgSO ₄	NaCl	MgO	SO ₂	P ₂ O ₅	Mg	Mg(OH) ₂
HCl	+	+	2	2	+	3	9	+	13	13	+	+
H ₂ S	+	+	11	12	+	3	+	+	13	13	+	+
KOH	4	4	+	+	+	+	6	4	+	+	4	4
Cu(OH) ₂	4	4	+	+	10	10	10	4	5	5	4	4
NaCl	6	10	3	3	+	8	1	1	1	1	7	10
K ₂ CO ₃	6	+	+	+	+	+	+	+	13	13	7	+
CuCl ₂	+	+	3	3	+	8	8	1	13	13	+	+
CO ₂	+	+	2	2	+	1	1	+	1	1	1	+
CaO	4	4	+	+	13	13	1	1	+	+	1	4
K ₂ O	4	4	+	+	13	13	1	1	+	+	1	4
Fe	4	4	+	+	+	7	7	1	1	1	1	4
H ₂ CO ₃	+	+	2	2	+	+	+	+	13	13	+	+

число из диапазона 1—13, то данные вещества не взаимодействуют друг с другом.

Знания по химии, заложенные в табл. 1, не выходят за пределы общего школьного курса и факультативных занятий. Химические соединения подобраны таким образом, чтобы можно было наиболее полно представить и обобщить сведения о свойствах неорганических соединений. При этом предполагается, что условия проведения реакций обычные, комнатные, а кислоты, основания и соли представлены в виде водных растворов.

Обратим внимание на то, что кислотные и основные оксиды могут взаимодействовать с водой даже в растворах кислот, оснований, солей, однако по правилам игры учитываются уравнения реакций только между основными веществами.

В табл. 2 приводится классификация ошибок, которые могут быть допущены школьником во время игры. В первом столбце указан номер ошибки, а во втором приводятся комментарии или правила взаимодействия химических соединений, незнание которых может повлечь за собой ошибки в процессе игры. Поэтому перед началом игры во вводных кадрах указанные правила выводятся на экран дисплея, и школьнику на первом этапе обучения следует хорошо познакомиться с ними.

Целые числа из табл. 1 — это номера ошибок из табл. 2. Пусть на пересечении *i*-й строки и *j*-го столбца ($1 \leq i, j \leq 12$) табл. 1 стоит число *n* ($1 \leq n \leq 13$). Если в процессе игры школьник попытается «заставить» прореагировать вещества, стоящие в *i*-й строке и *j*-м столбце, на экран дисплея будут выведены сообщение о допущенной ошибке с номером *n* и соответствующие данной ошибке комментарии и правила, которые следует запомнить.

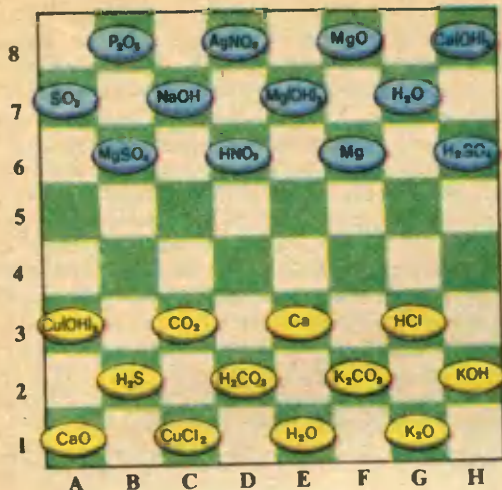
Табл. 3 содержит уравнения всех допустимых реакций для химических соединений из информационного обеспечения игры. Их ровно столько, сколько плюсов в табл. 1. Это те уравнения реакций химических соеди-

нений, которые школьник должен научиться составлять в процессе игры.

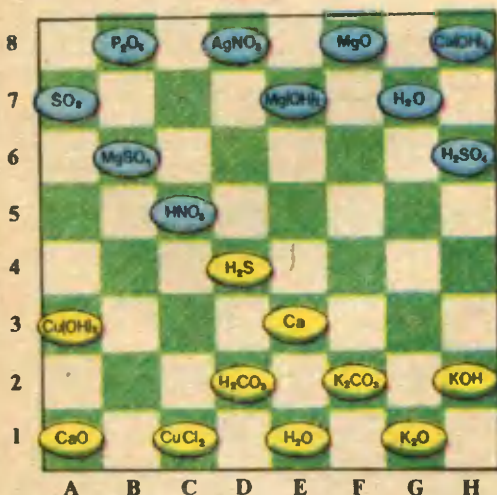
Игра начинается с ознакомления учащегося с правилами игры в шашки и правилами записи ходов. Затем учащийся имеет возможность ознакомиться с основными правилами взаимодействия химических соединений, выводимыми на экран дисплея. Темп листания страниц задает сам учащийся клавишей ВК. Для записи ходов выбрана обычная нотация (например, А3—b4, с7—b6, d4:h8, e3:g1). В качестве примера рассмотрим фрагмент игры. Начальное положение приведено на рис. 1.

Таблица 2

Номер ошибки	Правила и комментарии
1	Указанные вещества не реагируют между собой в обычных условиях
2	Кислоты не реагируют с кислотами и кислотными оксидами
3	Растворимые кислоты вступают в реакцию обмена с солями, если хотя бы один из продуктов реакции нерастворим или выделяется в виде газа
4	Основания не реагируют с основаниями, основными оксидами и металлами
5	С кислотными оксидами взаимодействуют растворимые основания
6	Щелочи реагируют с растворами солей с образованием новой соли и нового основания, если хотя бы один из продуктов реакции нерастворим в воде
7	В водных растворах солей металл, входящий в их состав, может замещаться только более активным металлом
8	В водных растворах соли реагируют между собой с образованием двух солей, если хотя бы одна из образующих солей нерастворима в воде
9	Реакция обмена не происходит
10	Соли реагируют с растворимыми основаниями только в тех случаях, когда ожидается новое основание или новая соль нерастворимы в воде
11	Сероводородная кислота не реагирует с раствором азотной кислоты, но с концентрированной азотной кислотой идет окислительно-восстановительный процесс: $\text{H}_2\text{S} + 2\text{HNO}_3 = \text{SO}_2 + 2\text{NO} + 2\text{H}_2\text{O}$
12	Сероводородная кислота не реагирует с раствором серной кислоты, но с концентрированной серной кислотой идет окислительно-восстановительный процесс: $3\text{H}_2\text{S} + \text{H}_2\text{SO}_4 = 4\text{S} + 4\text{H}_2\text{O}$
13	Оксиды не реагируют с данным веществом. Взаимодействие некоторых оксидов с водой из растворов кислот, оснований, солей не учитывается по правилам игры

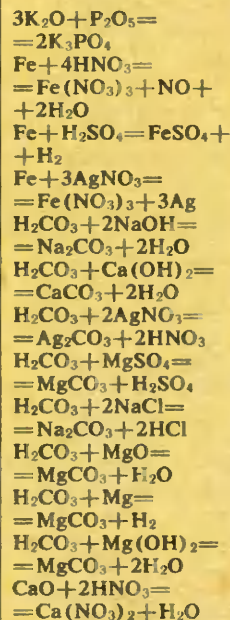
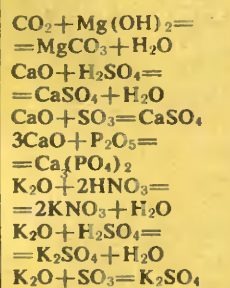
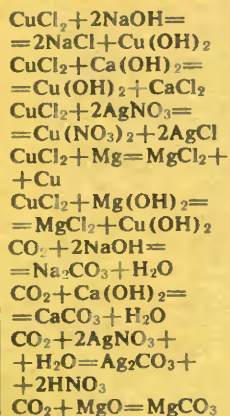
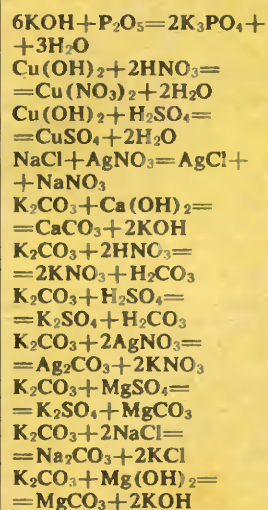
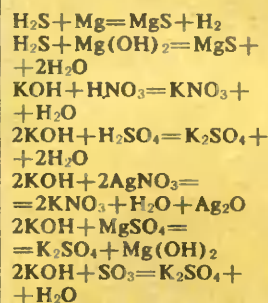
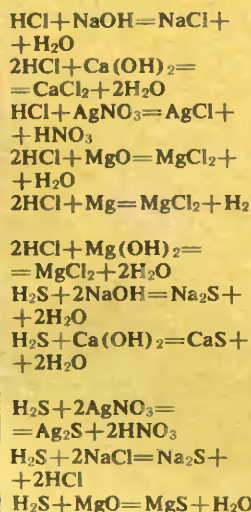


46 После выполнения серии ходов 1. g3—f4 d6—c5 2. c3—d4 f6—e5 3. f4:d6 c7:c3 4. b2:d4 получаем положение, приведенное на рис. 2.



При выполнении третьего и четвертого ходов произошел обмен фигур: компьютер и учащийся сняли по две шашки. На экран дисплея были выведены следующие уравнения реакций: $2\text{HCl} + \text{Mg} = \text{MgCl}_2 + \text{H}_2$, $\text{HCl} + \text{NaOH} = \text{NaCl} + \text{H}_2\text{O}$, $2\text{NaOH} + \text{CO}_2 = \text{Na}_2\text{CO}_3 + \text{H}_2\text{O}$, $\text{H}_2\text{S} + 2\text{NaOH} = \text{Na}_2\text{S} + 2\text{H}_2\text{O}$.

Игра может завершиться, когда у противника не останется ни одной фигуры либо когда фигуры есть, но нет свободного хода. Иногда целесообразно заканчивать игру по истечении некоторого времени, например через 15—20 мин. В любом случае при завершении игры на экран выводится статистическая информация: количество набранных баллов, количество побитых фигур, общее количество допущенных ошибок и оши-



бок с каждым номером в отдельности, а также оценка по пятибалльной системе. По желанию преподавателя критерий опре-

деления оценки может изменяться или оценка может вовсе не выставляться. На основе статистической информации учитель имеет возможность определить, какой материал наиболее труден для усвоения, и учесть это в своей работе.

Хотя настоящая версия компьютерной программы рассчитана на изучение темы «Обобщение сведений о важнейших классах неорганических соединений», она может легко быть адаптирована и для изучения других тем по курсам органической и неорганической химии. Для этого достаточно изменить информационное обеспечение игры и на его основе в табл. 1, 2, 3 внести соответствующие изменения. С такой работой под руководством учителя информатики могут справиться школьники, посещающие занятия факультативов и кружков по химии и знакомые с основами программирования на Бейсике.

В заключение заметим, что компьютерная программа «Химические шашки» может быть использована учащимися школ и профессионально-технических училищ не только на

уроках химии, но и во внеурочное время на факультативах, кружках или во время самостоятельной работы в кабинете информатики. Первый опыт использования ее в школах и профтехучилищах Минска показал, что она позволяет значительно повысить уровень знаний обучаемых по химии. Учащиеся достаточно быстро запоминают правила игры и хорошо усваивают химические свойства солей, кислот и оснований. В непринужденной игре с компьютером они доводят свои знания и умения до уровня навыков. Особенно полезной игра «Химические шашки» оказалась для тех учащихся, которые проявляют интерес к химии и желают избрать ее в качестве своей будущей профессии.

Дополнительную информацию по вопросам приобретения и использования программного средства «Химические шашки» можно получить по телефонам 26-81-02, 26-86-36. Адрес для переписки: 220080, Минск, Ленинский просп., 4, БГУ им. В. И. Ленина, научно-исследовательская лаборатория информатики и вычислительной техники.

47

В. ТИЦЕНКО

Работа с простейшим компьютером

Эффективно работать с компьютером можно, только если понимаешь принцип его действия и знаешь, какие операции (команды) он способен выполнять.

Поэтому прежде всего необходимо познакомиться с его устройством и особенностями. Рассказ об этом будет вестись в расчете на неподготовленного читателя.

Чтобы понять, как устроена и как работает с информацией «разумная» машина компьютер, вспомним, как выполняет эту же работу человек.

Сначала человек воспринимает информацию при помощи органов чувств (зрение, слух). Затем запоминает ее в своей памяти. После чего начинает обрабатывать путем вычислений, сопоставлений и логических рассуждений. После обработки человек выдает информацию в виде ответа на поставленную задачу, плана действий, нарисованного чертежа и др.

Точно так же поступает с информацией и компьютер. Он воспринимает ее при помощи технических средств, называемых устройством ввода, затем запоминает при помощи устройства памяти, обрабатывает, используя микропроцессор. И наконец, он вы-

дает обработанную информацию при помощи устройства вывода.

Рассмотрим наиболее важные особенности работы этих устройств.

Первая особенность сводится к тому, что любая информация (числовая, текстовая, графическая и др.), поступающая в компьютер, преобразуется с помощью устройства ввода в условное число, называемое числовым кодом. Для примера в табл. 1 приведены кодовые числа для некоторых символов (букв, цифр и знаков) в соответствии с международным стандартом.

Таблица 1

Символ	Десятичный код	Двоичный код
Русская буква А	225	1 1 1 0 0 0 0 1
Латинская буква А	65	0 1 0 0 0 0 0 1
Знак сложения +	43	0 0 1 0 1 0 1 1
Цифра 5	53	0 0 1 1 0 1 0 1
Знак «больше»	62	0 0 1 1 1 1 1 0
Точка	7	0 0 0 0 0 1 1 1
Вопросительный знак ?	63	0 0 1 1 1 1 1 1

Подобным образом кодируются и все другие виды информации.

Для упрощения запоминания кодовых чисел в памяти и обработки в микропроцессоре кодовые числа представляются в компьютере не в обычном десятичном виде, а в двоичном.

Двоичная система счисления является одной из важнейших особенностей работы не только компьютера, но и многих других современных электронных устройств (высококачественного телевидения, цифровой магнитной записи и др.). Поэтому понятие о двоичном способе представления чисел является очень важным и совершенно необходимым для тех, кто решил понять устройство компьютера и желает собрать его своими руками.

Коротко напомним, что в общепринятой десятичной системе счисления числа записываются при помощи 10 цифр, причем цифра, расположенная в числе слева, оказывается больше такой же цифры, расположенной справа, в 10 раз. Для примера рассмотрим число 333. Здесь справа стоит тройка. Такая же тройка, находящаяся левее (в середине числа), означает уже 30 (т. е. в 10 раз больше, чем 3), а написанная еще левее — 300 (т. е. снова в 10 раз больше).

В двоичной же системе счисления числа записывают с помощью двух цифр (1 и 0). При этом цифра 1, будучи расположена в числе слева, больше соседней справа в два раза. Например, число 111, написанное в двоичной системе, расшифровывается так: самая правая единица означает 1, написанная левее (в середине) означает 2 (в два раза больше), написанная еще левее — 4 (снова в два раза больше). Таким образом, 111 в двоичной системе соответствует $4 + 2 + 1 = 7$ в десятичной. Еще пример: десятичное число 5 запишется в двоичной системе, как 101.

В табл. 2 приведена запись первых шестнадцати чисел в десятичной и двоичной системах счисления. Внимательно рассматривая ее, можно понять, как выполняется запись чисел в двоичном виде. Если же из сказанного выше и из табл. 2 не удастся понять способ записи двоичных чисел, обязательно обратитесь к дополнительной литературе, где двоичная система рассмотрена более подробно, так как пробел в знаниях двоичного исчисления не позволит понять работу компьютера.

Второй особенностью компьютера является электронная природа его устройств. Все они собраны из электронных элементов (транзисторов, диодов, резисторов и конденсаторов), подобных тем, из которых сделаны

Число в десятичной системе счисления	Число в двоичной системе счисления
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

радиоприемники и телевизоры. Только в компьютере их несравнимо больше. Используя уникальную технологию, удастся в крошечном объеме кристалла микропроцессора (размером со спичечную головку) разместить сотни тысяч транзисторов и соединить их друг с другом по нужной схеме.

Электронный характер устройств компьютера требует некоторых знаний в области электроники. Особенно это относится к процессу налаживания компьютера. Необходимо знать, что такое электрический ток, напряжение, сопротивление; понять принцип работы электронных элементов, а также научиться рисовать и читать простейшие электрические схемы.

Тому, кто не имеет достаточно сил и времени для освоения электроники, необходимо как можно больше пользоваться аналогиями и житейскими сравнениями. Для примера укажем, что электрический ток в популярной литературе часто сравнивается с потоком воды в трубах, а диод сравнивают с водяным клапаном, пропускающим воду в одну сторону.

Весьма полезно бывает пользоваться понятием «черного ящика», иначе говоря рассматривать сложные микросхемы только со стороны входных и выходных выводов, определяя, согласно заводской таблице истинности, что будет на выходе микросхемы, если совершать те или иные действия на входе, и пренебрегать при этом знанием очень сложного, непонятного («черного») внутреннего устройства.

Третьей особенностью компьютера является его программная работа. Он выполняет только те действия и в той последователь-

ности, которые указаны в программе, заложены в память.

Программу для компьютера пишут на особом языке, называемом языком программирования.

Известно, что лучшим средством выражения мыслей и способом описания всего того, что происходит в мире, является наш разговорный язык. Однако существует ряд понятий и предметов, рассказывать о которых более удобно на специальных языках. Примерами таких языков являются нотный язык музыки, язык расположения шахматных фигур, телеграфный язык азбуки Морзе, язык химических формул и др.

Теперь расскажем об устройстве четырех основных частей компьютера.

Модуль ввода

В качестве простейшего устройства ввода двоичных чисел в компьютер применяются обычные переключатели (тумблеры, кнопки). В одном положении переключателя вводится высокий уровень электрического напряжения, изображающий собой единицу двоичного числа, а во втором — низкий уровень напряжения, изображающий ноль.

В более сложных конструкциях ввод двоичного кода производится путем нажатия клавиш на специальной клавиатуре типа пишущей машинки. При нажатии клавиши особое электронное устройство преобразует замыкание контактов клавиши в двоичный код, соответствующий значению данной клавиши, и посылает этот код в компьютер.

Кроме того, двоичные числа (двоичный код) часто вводят при помощи бумажной перфоленты, магнитной ленты или магнитного диска.

В этих случаях пробитое в перфоленте отверстие или намагниченный участок ленты (диска) изображают единицу, а размагниченный участок или «непробитое отверстие» — ноль.

Если компьютер используется для программного управления какой-либо машиной или технологическим процессом, то в качестве устройств ввода применяют различные датчики (температуры, давления, перемещения и др.), преобразующие информацию о происходящем процессе в двоичный код; используются и речераспознающие устройства.

В нашем простейшем компьютере для ввода двоичных чисел мы использовали обычные тумблеры.

Модуль памяти

В популярной литературе запоминающее устройство (память) часто сравнивают со шкафом, имеющим пронумерованные ящики,

в которых хранятся двоичные числа. При этом программисты называют номера таких ящиков адресами, а ящики — ячейками, говоря, что в ячейке, имеющей такой-то адрес, содержится такое-то двоичное число.

Каждый разряд этого числа запоминается в своем отделении ящика-ячейки. Такие отделения (запоминающие элементы) выполняются в простейшем случае в виде конденсаторов (заряженный конденсатор изображает единицу двоичного числа, а разряженный — ноль).

Используемая в нашем компьютере микросхема памяти типа К537РУ10 (см. рис. 4 на с. 61, «ИНФО» № 5 за 1990 г.) содержит 2048 ячеек памяти, а каждая ячейка имеет восемь запоминающих элементов, позволяющих хранить восьмиразрядное двоичное число.

Чтобы обратиться к нужной ячейке, указывают ее адрес путем подключения к адресным выводам А0—А10 уровней высокого (1) или низкого (0) электрического напряжения в соответствии с единицами или нулями двоичного числа, равного этому адресу. Например, чтобы вызвать к работе начальную (нулевую) ячейку памяти, на все выводы А0—А10 нужно подать нулевой (низкий) уровень напряжения, а чтобы вызвать самую последнюю ячейку (№ 2047), на выводы А0—А10 следует подать «единичный» (высокий) уровень напряжения. Ячейку памяти № 5 можно вызвать, подавая на выводы А0—А10 такие уровни напряжений:

Вывод	А10	А9	А8	А7	А6	А5	А4	А3	А2	А1	А0
Уровень напряжения	0	0	0	0	0	0	0	0	1	0	1

Если при этом на вывод 21 (Ч/З) микросхемы К537РУ10 подать единичный уровень напряжения, то выходы всех восьми запоминающих элементов (их называют разрядами) вызываемой ячейки памяти подключаются к выводам Д0—Д7; теперь с них можно считать восьмиразрядное двоичное число, находящееся в ячейке. Если же на вывод 21 подать нулевой уровень, то к выводам Д0—Д7 подключаются входы восьми запоминающих элементов и можно записать в ячейку (заставить ее запомнить) двоичное число, цифры которого поданы на выводы Д0—Д7 в виде уровней (1 или 0) электрических напряжений.

Микросхема памяти работает в режиме чтения или записи при условии, что на ее вывод 18 (ВК) подан нулевой уровень напря-

жения. В случае подключения к выводу 18 единичного уровня ячейка памяти делается неработоспособной. Ее входы и выходы отключаются от выводов D0—D7. При работе компьютера выходы A0—A10 подключают к адресной шине, выходы D0—D7 — к шине данных, а выходы 18 и 21 — к шине управления.

Микросхема K537PY10 имеет два вывода питания (24 и 12), подключаемые к шине питания. И еще: вывод 20 (DE) не используется, его нужно присоединить к общему выводу питания.

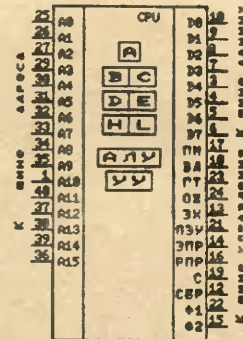
Применяемая в нашем компьютере память называется оперативным запоминающим устройством (ОЗУ). Этот тип памяти имеет свои достоинства и недостатки. Достоинством является оперативность в работе, позволяющая микропроцессору за доли секунды обратиться к любой ячейке памяти ОЗУ, чтобы записать или прочесть двоичное число. Недостаток — разрушение информации (двоичных чисел) при выключении питания; электрические элементы, не получая питания, теряют свое состояние высокого или низкого электрических уровней. Поэтому наряду с ОЗУ применяют память типа ПЗУ (постоянное запоминающее устройство), сконструированную так, что выключение питания не изменяет состояние запоминающих элементов. К сожалению, ПЗУ позволяет оперативно (с той же скоростью, что и ОЗУ) только считать числа, хранящиеся в ячейках памяти. Запись же чисел производится очень долго (в течение десятков секунд и даже минут), а в некоторых конструкциях вообще возможна только один раз. Поэтому такой тип памяти используется в компьютере исключительно для чтения микропроцессором двоичных чисел, записанных в ПЗУ где-то в другом месте.

В микросхемах ОЗУ и ПЗУ, выпускаемых промышленностью, можно хранить несколько тысяч (в последних образцах — десятки тысяч) байтов двоичных чисел. В то же время некоторые программы насчитывают миллионы байтов. Такие программы размещают во внешних запоминающих устройствах (ВЗУ), хранящих информацию чаще всего на магнитных лентах или дисках. Так как поиск нужного места ленты (диска) занимает много времени, программу, записанную в ВЗУ, частями (порциями) переписывают в ОЗУ и последовательно обрабатывают эти части.

Модуль микропроцессора

Микропроцессор является самым главным элементом компьютера. Он выполняет обработку информации, представленной в виде двоичных чисел.

Обработка сводится к вычислению, сравнению, упорядочению, преобразованию и пересылке двоичных чисел, которые вводятся в микропроцессор, обрабатываются там и выводятся в виде электрических сигналов высокого или низкого уровня. Эти сигналы создаются в микропроцессоре с помощью многочисленных электронных элементов — транзисторов, диодов, резисторов и др. Современный микропроцессор насчитывает сотни тысяч таких элементов, соединенных между собой по сложной схеме и образующих большое количество взаимосвязанных узлов, блоков и устройств, выполняющих различные действия. Среди множества этих устройств все же удается выделить несколько главных. К ним относятся регистры, арифметическо-логическое устройство (сокращенно АЛУ) и устройство управления (УУ).



На рис. 1 изображена упрощенная схема микропроцессора КР5801ИК80А, применяемого в нашем простейшем компьютере. На этой схеме латинскими буквами А, В, С, D, E, H и L обозначены регистры, представляющие собой ячейки памяти, служащие для кратковременного запоминания восьмиразрядных (однобайтовых) двоичных чисел, поступающих на входы D0—D7 микропроцессора, или пересылаемых из одного регистра в другой, или пересылаемых из регистра в АЛУ и обратно.

АЛУ выполняет арифметические вычисления (сложение, вычитание) и логические операции, сводящиеся зачастую к проверке какого-либо условия. Если условие выполняется, то АЛУ выдает 1 — условие истинно, а если нет, то 0 — условие ложно.

УУ вырабатывает серию управляющих сигналов в виде низких и высоких электрических уровней. Сигналы поступают в АЛУ, регистры и другие узлы микропроцессора для координации их работы, а также передаются на управляющие выходы (шину управления) для согласования действий микропроцессора с действиями внешних устройств, например ОЗУ.

Примечание.

Символ "#" — курсор, который перемещается с помощью клавиш "Стрелка вверх" и "Стрелка вниз"; служит для выбора нужной операции.

На этом этапе преподаватель может работать с меню на РМР, выбирая и выполняя те операции, которые ему требуются.

Подготовленный пользователь может выйти из меню в ОС, нажав одновременно клавиши "УРР" и "С".

2. Просмотреть содержимое рабочего диска (ГМД), вставленный в нижний карман накопителя). Для этого достаточно нажать на клавишу "ВВОД", предварительно установив курсор "#" напротив 1-позиции меню (п. 2 выполнять необязательно).

3. Подготовить РМУ к загрузке ОС, для чего включить питание и убедиться, что на экране появилось сообщение: «Загрузка из сети».

Примечание.

Для повышения надежности работы КУВТ во время загрузки ОС рекомендуется готовить РМУ к загрузке последовательно, переходя к следующему РМУ после того, как завершена загрузка на предыдущем.

Желательно не работать с ОС до окончания загрузки во все РМУ (время последовательной загрузки двенадцати РМУ — 3—5 мин). В дальнейшем работы могут вестись параллельно.

4. Загрузить ОС в РМУ, выполнив операцию 9 меню. При успешной загрузке ОС на экране монитора РМУ должна появиться точка ".", после чего на клавиатуре РМУ нужно набрать и исполнить команды:

ASS VK1 DK "ВВОД"
С RМУ # "ВВОД"

На экране появится новое меню:

```

.....
;
;          РАБОТА С КОМПЬЮТЕРОМ
;
;  1  Просмотр рабочего диска
;  2  Загрузка Бейсика с рабочего диска
;  3  Загрузка Фокала с рабочего диска
;  4  Текстовый редактор
;  5  Графический редактор
;  6  Создание новой базы данных
;  7  Ввод и исправление информации в базе данных
;  8  Просмотр и анализ информации в базе данных
;
;
.....

```

5. Работать с необходимыми прикладными программами в соответствии с инструкциями на них.

Примечание.

Для работы с программами, не вошедшими в перечень меню, можно выйти из меню в ОС. Для этого требуется одновременно нажать клавиши "УРР" и "С", а далее, после приглашения (появления на экране точки "."), работать с нужными программами непосредственно в ОС.

Для повышения надежности работы нажимать клавиши ТОЛЬКО после появления соответствующего

запроса системы! Учитывая увеличение времени реакции на команду ОС, загруженной в РМУ, нужно быть внимательным и проявлять терпение!

После загрузки ОС в РМУ в режиме функционирования сети РМР работает только на обслуживание запросов от РМУ и обеспечивает поддержку сети, время от времени выдавая на экран монитора служебную информацию. Для перехода РМР в автономный режим нужно нажать клавишу "ВВОД" и на запрос системы, какую команду исполнить, нажать клавишу "К" ("Конец"). На экране появится меню.

Кроме интерфейса предлагаемый пакет включает в общей сложности более ста учебно-прикладных программ, написанных на разных алгоритмических языках.

В пакет входят обучающие программы, позволяющие самостоятельно освоить основы Бейсика и Фокала; учебные программы по общеобразовательным предметам (математика, физика, русский и иностранные языки, химия, биология, астрономия и др.) для учеников V—X классов. Кроме того, имеются игровые программы. Записаны необходимые для работы инструкции.

При работе учеников на РМУ с учебно-прикладными программами, написанными на языках Бейсик или Фокал, нужно предварительно выполнить операции 2 или 3 меню по загрузке соответствующих интерпретаторов. Для этого в рабочем кармане накопителя должен находиться ГМД, на котором записаны эти интерпретаторы (BASIC.SAV или DBAS.SAV, FOCAL.SAV — имеются в пакете).

Далее необходимо загружать нужный файл и запустить программу на исполнение, ведя диалог с программой на русском языке.

С командами текстового и графического редакторов, используемыми в пакете, знакомят помещенные там же «Инструкции пользователя».

Для ознакомления с основными режимами работы СУБД (РБД-Микро) рекомендуем обратиться к книге А. А. Осадчиева «Персональный компьютер» (М.: Московский рабочий, 1988).

При выводе файла на печать нужно, выполнив операцию 10 меню, указать имя распечатываемого файла с расширением и в случае появления запроса системы о наличии бумаги в принтере (Page ready?) дать подтверждение нажатием клавиши "ВВОД".

На ГМД, входящих в пакет с системой меню, записаны следующие эксплуатационные документы:

MENU.HLP — интерфейс пользователя для работы с КВУ «Электроника МС 0202» (инструкция оператора);

EDITOR.HLP — инструкция оператора по работе с текстовым редактором;

Неисправность	Что нужно сделать
А. Исчезло меню. На экране появился символ "°"	После "°" набрать на клавиатуре команду CRMU (для РМУ) CRMР (для РМП) и нажать "ВВОД". Меню восстановится
Б. На РМУ исчезло меню и появился символ "°"	Перезагрузить ОС в это РМУ
В. На РМП исчезло меню и появился символ "°"	Загрузить меню с системного диска

GRED.HLP — инструкция оператора по работе с графическим редактором;

NET.HLP — описание кольцевой локальной вычислительной сети.

Приступая к работе на вычислительном комплексе, не забудьте убедиться в исправности физической линии связи между РМП и РМУ! Для этого нужно воспользоваться соответствующей штатной методикой проверки работоспособности КУВТ.

Характерные неисправности и способы их устранения приводятся в таблице.

Более подробную информацию по данному пакету учебно-прикладных программных средств можно получить по адресу: 121002, Москва, Серебряный переулок, 2, НЦПСО. Телефон: 272-11-25.

Ю. КОЗЛОВ, Л. ЛЕПИН

Две утилиты для «Ямахи» MSX-2

Система NETWORK предназначена для организации учебной работы на КУВТ «Ямаха» MSX-2, в котором ученические ЭВМ не оборудованы собственными дисководами. Отсутствие дисководов не позволяет запускать на РМУ программы, содержащие обращение к нему. На занятиях по обучению программированию на Бейсике много учительского времени занимает пересылка по сети ученических программ, а также прием их для записи на диск. Система NETWORK позволяет справиться с этими трудностями. Она обеспечивает каждого учащегося как бы собственным дисководом, позволяя выполнять такие операторы, как LOAD, RUN, SAVE, BLOAD, FILES, KILL. Все обращения при этом переадресуются на учительский дисковод. Если несколько учеников одновременно пытаются работать с учительским дисководом, то они обслуживаются поочередно. На экране учительской ПЭВМ будут выдаваться сообщения об обращении к дисководу каждого из учеников. В то же время от учителя при этом не требуется каких-либо действий, и освободившееся время он может посвятить более углубленной индивидуальной работе с учениками. Все перечисленные выше операторы могут быть использованы как в командном, так и в программном режиме, хотя, вероятно, операторы LOAD, SAVE, FILES и KILL в программном



режиме использовать не стоит. Приведенный ниже пример показывает, как можно использовать в программном режиме оператор BLOAD.

```
10 SCREEN 5
20 BLOAD"PICTURE.SCR",S
30 GOTO 30
```

Эта небольшая программа, выполненная на ученической ПЭВМ, выведет на экран дисплея ученика картинку, содержащуюся в файле PICTURE.SCR, которая могла быть создана и записана на диск с помощью, например, программы PAINTER, и сохранит ее до нажатия клавиш CTRL+STOP.



С помощью оператора RUN, используемого в программном режиме, можно из одной программы вызвать и запустить другую, что позволяет выполнять на ученических ПЭВМ пакеты, состоящие из нескольких взаимосвязанных программ.

Программа PRSCR позволяет значительно расширить возможности использования принтера. После загрузки программа находится в редко используемой области машинной памяти, может быть вызвана во время работы других программ путем нажатия клавиши SELECT и распечатывает на принтере то, что в момент нажатия клавиши SELECT находилось на экране. Программа сама определяет режим экрана и в зависимости от этого выбирает текстовый или графический режим распечатки. Если на экране имеются спрайты, программа распечатает и их, определив предварительно их тип. При распечатке программа различает 64 оттенка цвета, ставя им в соответствие различные матрицы распечатки. Это позволяет достаточно хорошо распечатывать даже такой многоцветный экран, как SCREEN 8. К ее достоинствам можно отнести также то, что окружность, нарисованная на экране оператором, скажем, CIRCLE (100, 100, 80), будет выглядеть на распечатке именно окружностью, а не эллипсом. Обычно распечатка графического экрана производится в формате 9×7,5 см, но если в момент нажатия клавиши SELECT была нажата клавиша SHIFT, то распечатка будет произведена в увеличенном формате 18×15 см. В увеличенном формате качество передачи мелких деталей рисунка значительно улучшается. Среднее время распечатки графического экрана: малоформатного — около 1,5 мин, крупноформатного — 5—6 мин.

Адрес для справки: 226250, ГСП, Рига, бульвар Райниса, 29, Институт математики и информатики Латвийского университета, отдел ОДУ. Тел. (0132) 213206.

Ю. КРАСНОВ

Вопросы с однозначным ответом

Машинный контроль знаний освобождает преподавателей от рутинной работы по опросу студентов, отнимающей много сил и времени. Устное собеседование также необходимо, но не в форме зачета, а в виде свободного обмена мнениями в часы, предусмотренные для консультаций. Устный зачет или экзамен не способствуют свободе обмена мнениями, так как сказывается психологическое напряжение студентов.

Машинный контроль повышает самостоятельность и уверенность студентов, позволяет им защищать лабораторные работы в

любое удобное для них время, а не только по расписанию. Оценки при машинном контроле может записать лаборант, если это не предусмотрено в программе контроля. Ответственность студентов повышается, ибо при получении неудовлетворительной оценки студенту обижаться не на кого, кроме как на самого себя.

Широта дидактических возможностей машинного контроля зависит от качества контролирующего процесса. Наиболее распространенный в настоящее время метод основывается на предъявлении нескольких отве-

тов, один из которых является верным (множественный выбор). Такой подход должен быть решительно отвергнут, так как он несет ложную информацию. Например, при пяти вариантах ответа 80 % находящейся на экране информации ложны, являются «информационной помехой» и снижают качество обучения.

Необходимо стремиться к вопросам, поставленным так, чтобы ответы на них были единственными. Ниже излагаются некоторые возможные варианты и примеры реализации такой методики.

1. Ответ непосредственно вытекает из вопроса. *Пример.* Сколько спутников у Марса? *Ответ:* 2.

2. Ответ получается в результате решения какой-либо задачи. *Пример.* К трансформатору подведено напряжение 220 В, коэффициент трансформации равен 4. Какое напряжение будет на вторичной обмотке трансформатора? *Ответ:* $U_2 = 220/4 = 55$ В.

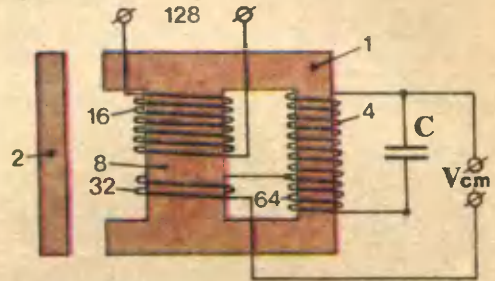
Эти варианты хорошо известны, но они не исчерпывают возможных подходов. Перспективным представляется метод использования двоичных индексов, суть которого заключается в индексировании элементов описания ситуации, к которой относится вопрос, числами, являющимися степенями двойки.

3. Использование индексированного рисунка. *Примеры.* Указать элементы схемы (рис. 1) феррорезонансного стабилизатора

напряжения, относящиеся к линейному дросселю, и обмотку, компенсирующую нестабильность напряжения на контуре из-за негоризонтальности его характеристики $U = f(I)$. Соответствующие индексы сложить. *Ответ:* $2 + 8 + 16 + 32 = 58$.

На схеме магнитного пускателя ПМЕ-223 (рис. 2) указать цепь прохождения тока от фазы А к фазе В при запуске асинхронного электродвигателя после отпущания кнопки «Вперед». Соответствующие индексы сложить. *Ответ:* $1 + 8192 + 512 + 64 + 128 + 2048 + 2 = 10\ 949$.

2



55

4. Использование индексированного текста. *Пример.* Холостой режим работы трансформатора.

Токи, возникающие в сердечниках трансформаторов, замыкаясь в толще сердечника, нагревают их, создавая потери энергии (1); в результате циклического перемагничивания сердечника трансформатора ток в катушке оказывается сдвинутым по фазе относительно потока на некоторый небольшой угол потерь (2);

токи, возникающие в толще сердечника, протекают в плоскостях, перпендикулярных направлению магнитного потока (4);

для уменьшения потери энергии, идущей на нагрев сердечников трансформаторов, их изготавливают из специальной трансформаторной стали (8);

для уменьшения потери энергии, идущей на нагрев сердечников трансформаторов, их набирают из отдельных изолированных друг от друга пластин (16);

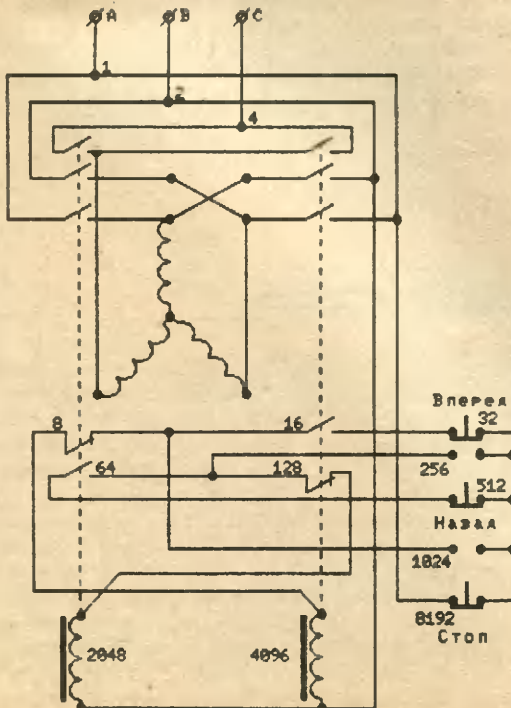
на перемагничивание сердечника затрачивается энергия, пропорциональная площади фигуры, изображенной на рис. 3 (32).

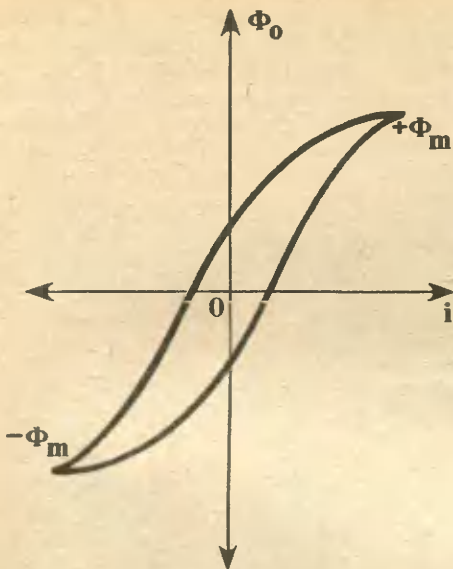
Задание 1. Сложить индексы утверждений, говорящих о вихревых токах и их влиянии на потери энергии. *Ответ:* $1 + 4 + 16 = 21$.

Задание 2. Сложить индексы утверждений, говорящие о гистерезисе и его влиянии на потери энергии. *Ответ:* $2 + 8 + 32 = 42$.

В тех случаях, когда очередность элементов описания ситуации, к которой относится вопрос, однозначно определена для правильного

1





руется значение первичного тока в зависимости от изменения тока нагрузки;

- 9 — создается магнитный поток Φ_2 ;
- 10 — наводятся ЭДС ε_1 и ε_2 в обмотках;
- 11 — появляется ток холостого хода I_{10} ;
- 12 — увеличивается магнитный поток Φ_1 ;
- 13 — магнитный поток снова станет равным Φ_0 .

Вопрос 1. Перечислить процессы, происходящие в трансформаторе при холостом режиме. Индексы ввести в ПЭВМ подряд, без пробелов. **Ответ:** 411210.

Вопрос 2. Перечислить процессы, происходящие в трансформаторе после включения нагрузки до возникновения неравенства $\varepsilon_1 < U_1$. **Ответ:** 19635.

Возможны, естественно, и другие варианты использования предложенных методов. По всей вероятности, могут формироваться вопросы с однозначными ответами не только по техническому и естественнонаучному дисциплинам, но и по гуманитарным.

В случае недостатка ПЭВМ для группового контрольного опроса может использоваться только одна машина. В этом случае учащимся должны быть заранее розданы соответствующие тексты и рисунки, а ПЭВМ используется только для регистрации ответов, на что требуется 10—20 с для каждого учащегося.

Справки по применению метода можно получить у Ю. С. Краснова (428000, Чебоксары, ул. Карла Маркса, 38, ЧГПИ, физмат) или В. В. Шишкина (428015 Чебоксары, Московский просп., 15, Чувашский государственный университет. Тел. 24-83-41). Будем признательны за сообщения о новых вариантах применения описанных методов.

56
ответа, можно использовать другую систему индексирования.

5. Описание детерминированных процессов. **Пример.** Процессы, происходящие в трансформаторе (здесь их последовательность нарушена).

- 1 — при подключении нагрузки появляется ток I_2 ;
- 2 — создается магнитный поток Φ_0 ;
- 3 — уменьшается ЭДС ε_1 ;
- 4 — включается напряжение U_1 ;
- 5 — возникает неравенство $\varepsilon_1 < U_1$;
- 6 — уменьшается общий магнитный поток;
- 7 — увеличивается ток I_1 ;
- 8 — таким образом автоматически регули-

Ю. КОЛОМИЕЦ

Функции пользователя на «Агате»

В арсенале Бейсика «Агата» есть оператор EXEC, назначение которого, судя по документации, «исполнение текстовых файлов». Немногие знают, как им пользоваться, а для него можно найти интересные приложения.

Взять хотя бы определение функций пользователем в режиме диалога. Вообще-то для описания функций существует оператор DEF FNF(X), но в диалоге он не работает. Чтобы изменить вид функции в какой-либо программе, необходимо прервать ее выполнение, набрать с клавиатуры соответствующую программную строку и снова запустить программу. А можно поступить более оригинально — запросить вид функции в диалоге с пользователем, записать определение функции в виде программной строки в текстовый файл, а затем оператором EXEC ввести строку в программу и передать на нее управление.

Ниже приводится часть текста программы, реализующей этот метод. В строке 20 пользователь вводит с клавиатуры вид интересующей его функции. Затем управление передается на строку 640. Во время работы программы в НГМД должна быть вставлена дискета. На диске создается текстовый файл со строкой, в которой определяется функция,

а также командой передачи управления GOTO 40. После закрытия файла FUNK его содержимое оператором EXEC вводится в программу.

```

10 HOME : PRINT "ЗАДАЙТЕ ВИД ФУНКЦИИ" : PRINT
20 INPUT "F(X)=";F$
30 GOTO 640 'ОПРЕДЕЛЕНИЕ ФУНКЦИИ ЧЕРЕЗ ФАЙЛ
40 HOME
.
.
.
640 REM СОЗДАНИЕ ПРОГРАММНОЙ СТРОКИ
      С ОПРЕДЕЛЯЕМОЙ ФУНКЦИЕЙ
650 PRINT : PRINT CHR$(4);"OPENFUNK"
660 PRINT : PRINT CHR$(4);"WRITEFUNK"
670 PRINT "50 DEF FNF(X)=";F$
680 PRINT "GOTO 40"
690 PRINT : PRINT CHR$(4);"CLOSEFUNK"
700 PRINT : PRINT CHR$(4);"EXECFUNK"

```

Оператору EXEC можно найти еще одно ценное применение. Пусть не печалятся пользователи «Агата», что они не имеют

оператора MERGE для объединения двух Бейсик-программ. Объединение можно осуществить с помощью несложной процедуры.

Программа, подлежащая присоединению, записывается в текстовый файл. Сделать это можно так: в начале программы открывается на запись файл с каким-либо именем, затем командой LIST содержимое программы записывается в файл, который после этого закрывается. Вот пример.

```

10 ?? CHR$(4);"OPENFILE"
20 ?? CHR$(4)WRITEFILE"
30 LIST 40,50
40 HOME
50 ?"ПРИВЕТ!"
60 ?? CHR$(4);"CLOSEFILE"

```

Теперь достаточно исполнить эту программу, загрузить в ОЗУ другую программу, подлежащую объединению, и дать команду EXECFILE. Необходимо только следить, чтобы номера строк двух программ не перекрывались.



ИнфоМир

Пакет состоит из пяти «миров»: АльфаМир, Клави́р, КуМир, АсМир и МикроМир и обеспечивает программную поддержку школьного курса информатики А. Г. Кушниренко и др.

Цена комплекта: в заказе до 10 комплектов — 1000 руб.;
от 10 до 100 — 600 руб.;
свыше 100 — 400 руб.

Роботландия

Программно-методический комплекс для обучения информатике в раннем возрасте.

Цена комплекта: в заказе до 10 комплектов — 1000 руб.;
от 10 до 100 — 750 руб.;
свыше 100 — 600 руб.

Информатика-91

Пакет обучающих, контролирующих программ и моделей поддержки любого курса информатики в общеобразовательной школе.

Цена комплекта: в заказе до 10 комплектов — 600 руб.;
от 10 до 100 — 500 руб.;
свыше 100 — 400 руб.



Научно-производственный кооператив «ЦИТРОН» предлагает

оптовому заказчику партию персональных компьютеров на базе процессора Z-80.

Персональный компьютер на базе процессора Z-80 предназначен для обучения школьников основам программирования на языке Бейсик, а также для сбора и обработки информации, систематизации и ведения документации, решения практических задач бухгалтерского и кадрового учета на небольших предприятиях, а также для проведения досуга.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

центральный процессор Z-80;

разрядность — 8;

такты частота — 3,5 МГц;

объем ОЗУ — 48К байт;

объем ПЗУ — 16К байт;

формат изображения — 192×256;

8 цветов, 2 градации яркости;

полная программная совместимость с «ZX-Spectrum».

Желательно оформление заказа с предварительным авансированием.

НПК «Цитрон». Адрес: 193124, Ленинград, пр. Маршала Говорова, д. 34. Телефоны: (8-812) 142-99-11, 538-14-65.

Пользователям IBM-совместимых компьютеров

Если вы еще не овладели техникой скорочтения и десятипальцевой методикой работы на клавиатуре, вам необходимо приобрести обучающие системы

«ТЕМП» и «ЗИГЗАГ».

Система обучения скорочтению «Темп» повышает скорость чтения в несколько раз, развивает память, адаптируется к индивидуальным особенностям пользователя. Обучение состоит в прочтении с экрана монитора нескольких произведений русских писателей. *Стоимость 210 руб.*

Система обучения машинописи «Зигзаг» позволит получить основные навыки работы на русской и латинской клавиатуре «вслепую» десятипальцевым методом за 3—4 ч обучения. После 10—15 ч обучения скорость работы повышается до 100 знаков в минуту и растет до достижения «потолка» возможностей данного человека! *Стоимость 560 руб.*

Поставка каждой из программ на одной дискете 5,25.

Заявки и гарантийные письма направлять по адресу: 125015, Москва, а/я № 6. Тел.: 148-71-98.

Архитектура и ассемблер БК

59

Ю. ЗАЛЬЦМАН

Архитектура БК

Как работает ЭВМ?

Не волнуйтесь, автор не собирается рассказывать вам о триггерах, логических элементах, регистрах сдвига, счетчиках, источниках питания и прочей электронике, это не входит в его задачу. Речь пойдет о том, как ЭВМ работает с точки зрения программиста.

Как уже было сказано, в памяти ЭВМ хранится программа — последовательность двоичных кодов. При ее запуске процессор запоминает адрес ее начала в специальном регистре, называемом *счетчиком команд*, и читает первое слово по этому адресу, причем счетчик команд сразу же увеличивает свое содержимое на 2. В зависимости от того, какая команда прочитана, процессор может или сразу ее выполнить (если в ней заключены все необходимые данные), или заняться «добыванием» указанных в команде данных. Эти данные могут либо располагаться где-то в памяти (как уже говорилось, для программы и данных используется одно и то же ОЗУ), либо входить в состав самой команды, но не в первое ее слово. Команды БК-0010 могут иметь длину от одного до трех слов, причем первое слово — всегда код самой команды, а два последующих, если они есть, — данные (это могут быть как сами числа для выполнения над ними каких-то операций, так и указания на адреса, по которым нужны числа расположены). Если команда состоит более чем из одного слова, содержимое счетчика команд в процессе ее выполнения увеличивается не на 2, а на 4 или на 6. Таким образом, счетчик команд

всегда содержит адрес следующей команды программы.

После окончания исполнения очередной команды процессор опять читает слово по адресу, указанному в счетчике команд, и выполняет следующую команду и т. д. Ясно, что таким образом может исполняться так называемая линейная последовательность команд, когда команды следуют строго одна за другой. Но линейная последовательность не является достаточной для решения любых задач, необходимы еще, по крайней мере, две структуры — условие и цикл. Кто занимался программированием, знает, что исполнение этих структур связано с изменением естественной последовательности команд. Делается это в ЭВМ очень просто — достаточно прибавить к счетчику команд какое-либо число (положительное или отрицательное), и процессор послушно начнет выполнять команды уже в другом месте программы в соответствии с адресом, указанным счетчиком. Таким образом реализуются все необходимые структуры программы.

Как же устроена машинная команда, позволяющая столь гибко управлять работой процессора? Не вникая в подробности, можно сказать, что в ее 16 разрядах (если команда состоит из одного слова) есть сам код команды и указание на так называемые *операнды*, т. е. числа, с которыми нужно иметь дело. Если эти числа входят в состав команды, то, как уже сказано, они записаны в ее втором и третьем словах. Кроме того, команда еще содержит указание на то, нужно ли оперировать со словом или с байтом. Все эти указания располагаются в специально отведенных для этого разрядах слова команды, так называемых *полях*. Этим кратким

описанием мы пока и ограничимся.

А что представляют собой числа, с которыми оперирует ЭВМ?

Дополнительный код

ЭВМ, как легко догадаться, оперирует с двоичными числами. Положительные двоично-восьмеричные числа от 0 до 7 мы уже приводили. Но это еще не все. Для полноценной арифметики нужны и числа отрицательные, а они изображаются в ЭВМ не совсем обычным образом. Чтобы понять, как это делается, проделаем несколько действий с двоичными 4-разрядными числами. Сначала сложение:

$$\begin{array}{r} +0001 \\ +0011 \\ \hline 0100 \end{array} \quad \begin{array}{r} +1 \\ +3 \\ \hline 4 \end{array}$$

А теперь вычитание:

$$\begin{array}{r} -0001 \\ -0011 \\ \hline 1110 \end{array} \quad \begin{array}{r} -1 \\ -3 \\ \hline -2 \end{array}$$

Вам что-нибудь понятно? Почему у нас получилось 1110 и почему это —2? При вычитании мы столкнулись с тем, что нам потребовалось сделать «заем» в старших разрядах уменьшаемого, а поскольку «цена» каждого разряда двоичного числа 2, то у нас и получилось то, что получилось. При этом мы считали, что число у нас как бы неограниченной разрядности, и мы «заняли» единицу в пятом разряде, который мы не изображаем. Теперь — внимание! Старший разряд двоичного числа со знаком (в данном случае четвертый) называется *знаковым разрядом*. Если число положительное, он равен 0, а если отрицательное — 1. Значит, раз у нас число 1110 (старший разряд равен 1), то число это отрицательное. А как понимать остальные три разряда — 110? Почему это 2? Вот это и есть *дополнительный код*. Какой в нем смысл? Попробуем выполнить еще два действия:

$$\begin{array}{r} +1110 \\ +1110 \\ \hline 1100 \end{array} \quad \begin{array}{r} +(-2) \\ +(-2) \\ \hline -4 \end{array}$$

$$\begin{array}{r} +0101 \\ +1110 \\ \hline 0011 \end{array} \quad \begin{array}{r} +5 \\ +(-2) \\ \hline 3 \end{array}$$

А правда ли мы получили в первом случае —4? Проверим:

$$\begin{array}{r} -0000 \\ -0100 \\ \hline 1100 \end{array} \quad \begin{array}{r} -0 \\ -4 \\ \hline -4 \end{array}$$

Итак, 1100 — это действительно —4! Следовательно, вот в чем смысл дополнительного кода: мы можем не задумываться о знаках при операциях с положительными или отрицательными числами, знаки получаются автоматически! Очевидно, что во втором сложении результат тоже верен, 0011 — это 3. Теперь вам ясно, как получается дополнительный код? Чтобы получить отрицательное число в дополнительном коде, нужно вычесть соответствующее по абсолютной величине положительное число из нуля! Потому код и называется дополнительным, что отрицательное и положительное числа, равные по абсолютной величине, до пол н я ю т друг друга до нуля, или, как говорят, *до переноса* в разряд за пределы числа. Есть простое правило получения отрицательного числа в дополнительном коде, которое позволяет делать перевод в дополнительный код в уме. Если вам нужно получить отрицательное число, возьмите положительное, равное ему по абсолютной величине, замените в нем все единицы нулями, а нули единицами и прибавьте единицу. Попробуем? Берем 4 — 0100. Заменяем: 1011. Прибавим единицу: 1011 + 0001 = 1100. Мы получили —4, правило работает! Теперь очень легко получить любое отрицательное число в дополнительном коде. Например, возьмем число 01101100 (154). Заменяем нули и единицы (это называется *поразрядное инвертирование*): 10010011. Прибавим единицу: 10010100. Это — минус 154. Можете проверить?

$$\begin{array}{r} -00000000 \\ -01101100 \\ \hline 10010100 \end{array} \quad \begin{array}{r} -000 \\ -154 \\ \hline -154 \end{array}$$

Если же нам нужно перевести число из дополнительного кода обратно, делаем все наоборот: вычитаем единицу и поразрядно инвертируем полученное двоичное число. Попробуйте сами и получите абсолютную величину отрицательного числа. И так для вычисления любой разрядности.

А теперь попробуем выполнить еще одно действие:

$$\begin{array}{r} +0110 \\ +0011 \\ \hline 1001 \end{array} \quad \begin{array}{r} +6 \\ +3 \\ \hline -7 \end{array}$$

Вот это фокус! Дополнительный код почему-то «отказал»!

А дело в том, что произошло так называемое *переполнение*, т. е. перенос единицы в знаковый разряд. Просто-напросто такая разрядность (четыре бита) мала для сложения.

ния двух предложенных чисел. Все станет на место, если мы добавим еще один разряд слева (и будем уже его считать знаковым):

$$\begin{array}{r} + 00110 \\ + 00011 \\ \hline 01001 \end{array} \begin{array}{r} + 6 \\ + 3 \\ - 9 \end{array}$$

Но наша ЭВМ оперирует не любимыми числами, а только 8- и 16-разрядными, т. е. байтами и словами. Поэтому и знаковый разряд фиксирован — это разряд 07 для байта и 15 — для слова. Как же быть, если у нас фиксированный формат числа и добавить разряд мы не можем? Это предусмотрено — ЭВМ отмечает переполнение и дает о нем знать специальным *флагом результата*, о чем мы поговорим чуть позже. Обнаружив этот флаг, мы уже знаем, что результат в дополнительном коде неверный и нуждается в коррекции.

Есть в ЭВМ и еще один формат чисел — так называемые *числа без знака*. При этом мы включаем знаковый разряд в состав самого числа, которое в этом случае может быть только положительным. Ясно, что тогда число 1001 — это 9.

Зачем еще и такое представление числа? Просто для того, чтобы число могло быть больше на один разряд. Если для выражения числа 9 в дополнительном коде нам потребовались пять разрядов, то как числа без знака — только четыре. Если в один байт можно записать числа в дополнительном коде от -128 до $+127$, то без знака — от 0 до $+255$. А нередкий знак числа нас вовсе не интересует, потому не нужен и дополнительный код. Например, если нужно сравнить коды символов, то при чем тут знак? Иное дело — арифметика. Тут ради знака приходится жертвовать разрядностью числа. Но самое главное, что обычный двоичный код и код дополнительный — это, по сути, одно и то же и мы в принципе можем оперировать числами независимо от того, в дополнительном коде они или нет. А если нам вдруг нужен знак числа — посмотрим на его старший разряд, и все! Может быть, вам тут еще не все понятно, тогда попробуйте поупражняться в сложениях и вычитаниях двоичных чисел со знаком и без него, и вы поймете, что правила все время одни и те же. И еще вы заметите одну деталь: вместо того чтобы вычест число из другого, можно прибавить к первому числу второе в дополнительном коде, ведь это есть отрицательное число! Вот оно, главное преимущество дополнительного кода, вот зачем он введен на ЭВМ: мы можем не иметь отдельного устройства — «вычитателя», достаточно одного сумматора и перевода вычитаемого в

дополнительный код. Это намного упрощает устройство процессора и увеличивает скорость вычислений.

А может ли ЭВМ так же просто умножать и делить? Да. Попробуем «сдвинуть» какое-либо число влево. 00010 сдвигаем влево: 00100; еще раз: 01000.

У нас было число 2, получили число 4, а затем 8. Но ведь это умножение на 2! Да, так называемый *арифметический сдвиг влево* есть умножение на 2. При этом мы считаем, что на место младших разрядов числа пишется 0. Совершенно очевидно, что, сдвигая такое число вправо по тому же правилу, мы получим деление на 2. А если число со знаком «минус»? Попробуем его умножить на 2, причем правило то же самое. 10011 ($-13d$) сдвинем влево: 00110.

Получили... 6! Разве это правильно?! Мы опять столкнулись с тем же явлением — не хватает разрядности. Но на этот раз произошло не переполнение, а перенос, т. е. выход за пределы числа. В ЭВМ на этот случай тоже есть флаг, который отметит, что результат ошибочный и нуждается в коррекции. Ну а как с делением чисел со знаком на 2? Тут все в порядке, но теперь мы должны не вписывать слева нули, а *расширять знаковый разряд*, т. е. переписывать его значение в соседний разряд справа. 10011 (-13) сдвинем вправо: 11001 (-7).

Все верно, ведь результат округленный! Причем округление производится отбрасыванием младшего разряда. Но что, если нам нельзя округлять число? Ничего страшного, ведь у нас снова произошел перенос за пределы слова и флаг переноса отметит, что результат нуждается в коррекции.

Итак, мы в общих чертах разобрались, как ЭВМ выполняет арифметические действия. Нетрудно теперь придумать, как можно умножить или разделить на число, не кратное двум (вспомните, что умножение — это не что иное, как многократное сложение, деление же — многократное вычитание, а чтобы экономить время, вспомните, как умножают в столбик и делают «уголком»). Мы не будем разбирать соответствующие примеры, так как это заняло бы слишком много места, а пойдем дальше в изучении архитектуры нашей ЭВМ.

Продолжение следует.

Ассемблер БК

Оператор прямого присваивания. Выражения

После ознакомления со способами адресации рассмотрим еще два тесно примыкающих к ним вопроса. Мы знаем уже, что метками в тексте программ помечаются не-

которые строки, а в процессе трансляции и компоновки метке присваивается текущий адрес, т. е. адрес начала той строки, в которой она стоит. Этот способ задания адресов меток естествен и хорош, пока метка находится в тексте программы. Но как быть, если метка должна стоять «в стороне» от программы, например в области адресов ПЗУ или в системной области? Ограничиваться в таких случаях абсолютной адресацией, прямо указывая адрес, не всегда удобно, поэтому в ассемблере предусмотрен специальный механизм для *определения меток* — так называемый *оператор прямого присваивания*. Он записывается как $МЕТ=X$, где X — восьмеричное число без знака. Этот оператор не транслируемый, т. е. при трансляции он не преобразуется в машинный код, а только присваивает заданной метке значение адреса X . Оно заносится в таблицу меток и используется при трансляции и компоновке. Операторы прямого присваивания принято размещать в начале каждого отдельного модуля программы. Что же можно определить с помощью этого оператора и для чего?

Прежде всего, особенности работы ассемблера таковы (мы не будем вдаваться в подробные объяснения), что обращение по адресам системной области (0—777) и ПЗУ (100000—177777) возможно только с применением абсолютной адресации либо через регистр, косвенно. Относительная адресация этих ячеек невозможна. Вот эти-то значения адресов и могут быть присвоены меткам с помощью оператора прямого присваивания, причем при обращении по адресу метки, определенной таким образом, адресация все равно получается не относительная, а абсолютная. С другой стороны, оператор прямого присваивания не может приписать метке адрес, лежащий в пределах ОЗУ пользователя — при трансляции и компоновке это обернется ошибкой. Приведем пример (отрывки листинга реальной программы — драйвера принтера):

```

REG=57           Признак регистра
CPR=256          Копия порта
POZ=374          Позиция печати
POR=177714       Порт ввода-вывода
Выход в Фокал или MSD
;.....
CLRB REG         Рег. код сбросить
;.....
CLR POZ          Сброс позиции
;.....
CLR POR          Сброс порта
;.....
MOV R0,CPR       Запомнить символ

```

Точки здесь означают пропуски в тексте программы-оригинала (привести весь текст, конечно, невозможно). Сохранен также авторский текст комментариев. Здесь показано

как определение меток в начале текста с помощью операторов прямого присваивания, так и дальнейшее использование определенных меток. Отметим, что определенная таким образом метка может быть использована в составе команды с любым способом адресации, а не только относительным, и ведет себя, как обычная метка. Например, если бы мы записали $MOV\#POR, R4$, то тем самым число 177714, регистраемое метке POR, было бы записано в регистр R4.

Рассмотрим второй вопрос. Если мы обращаемся к операнду по метке, то подразумеваем, что она имеет фиксированный адрес. Но бывают случаи, когда нужно изменить его, например если меткой помечено начало текста и в одних случаях нам нужно обратиться к началу, а в других — к середине текста. Развитые версии ассемблеров допускают адресацию операндов с помощью не только меток, но и выражений, состоящих из *термов*. Термами называются входящие в состав выражения числа или символы, связанные знаками арифметических операций «плюс» или «минус». Число термов в выражении для рассматриваемых версий ассемблеров не должно превышать двух. Выражение записывается как $МЕТ+X$ или $МЕТ-X$ и в процессе трансляции вычисляется как одно целое. Приведем примеры. Пусть адрес метки TEX равен 1100. Тогда следующие команды выполнят действия:

```

CLR TEX         Очистка ячейки 1100
CLR TEX+100     Очистка ячейки 1200
CLR TEX-100     Очистка ячейки 1000

```

Совершенно аналогично можно задать операнды с помощью выражений и в других случаях, например:

```

CLR MET+10(R2)
MOV #MET-122, (R2)
CLRB #MET+43

```

При трансляции и компоновке вторым словом данных команд записывается результат вычисления выражений.

Операторы ассемблера

Операторы, или собственно команды, определяют, что программа должна делать и в какой последовательности. Операторов в составе ассемблера много, классифицировать их можно по-разному. Самой простой и удобной для изложения, видимо, является классификация по функциям:

- операторы для работы с операндами (одно- и двухоперандные);
- операторы управления программой;
- операторы управления машиной;
- прочие.

При описании операторов мы будем приводить форму их записи следующим обра-

зом. Если оператор может работать и с байтом, и со словом, то дополнение его имени для работы с байтом будет приводиться в скобках, например MOV(B). Если в составе команды допустимы любые способы адресации, то операнды в общем виде будут обозначаться как «А, В», причем символом А будет обозначаться обычно *операнд-источник*, в процессе исполнения команды не меняющийся, а символом В — *операнд-приемник*, куда заносится результат операции. Если один и тот же операнд является и источником и приемником, то он будет обозначаться «АВ». Если в процессе выполнения команды содержимое операндов не меняется (оба они как бы являются источниками), будем обозначать их «А1, А2». Если операнд в составе команды используется только как адрес перехода, в общем виде он будет обозначаться «N». Если оператор допускает не любые методы адресации, а лишь некоторые, это будет отмечаться особо; для обозначения этих методов будут использоваться уже знакомые символы RN, X, MET.

Перед тем как перейти к описанию конкретных операторов, кратко напомним, что кроме восьми регистров общего назначения в составе процессора имеется еще один служебный регистр PS, или регистр *слова состояния процессора* (ССП). При описании операторов нас будут интересовать четыре младших разряда (бита) ССП. Это так называемые *биты, или флаги, условий*. Они обозначаются буквами N, Z, V, C; значение каждого из них устанавливается равным нулю или единице в зависимости от результата выполнения очередной команды. Эти биты используются для выполнения условных переходов и некоторых других целей. Их значения устанавливаются равными единице в следующих случаях:

N — если результат операции является отрицательным числом;

Z — если результат операции — ноль;

V — если при операции произошло перенос (т. е. перенос в знаковый разряд);

C — если при операции произошел перенос (т. е. выход единицы за пределы слова или байта).

В противных случаях (если результат операции противоположный) соответствующие биты ССП равны нулю.

Биты условий устанавливаются в результате операции со словом или байтом в зависимости от формы оператора (обычной или байтовой). Более подробно этот вопрос рассматривается в разделе, посвященном центральному процессору ЭВМ.

Операторы для работы с операндами

С одним операндом

1. CLR(B) В — очистка. Обнуляет слово (байт) операнда. Примеров записи этого оператора было рассмотрено уже много. Записывает в операнд ноль независимо от исходного содержимого. Используется для обнуления (сброса) программных счетчиков, признаков, сумматоров, подготовки ячеек и т. п.

2. COM(B) АВ — поразрядное инвертирование. Все биты операнда независимо друг от друга заменяются на противоположные — ноль на единицу, а единица на ноль. Пусть в регистре R4 число 1 011 101 100 111 000 (восьмеричное 135470). После выполнения команды COM R4 в R4 будет 0 100 010 011 000 111 (восьмеричное 042307).

Таким образом, оператор выполняет операцию *дополнения до единицы*: $AB + COM(AB) = \langle \text{все единицы} \rangle$.

3. INC(B) АВ — инкремент. К операнду прибавляется 1.

4. DEC(B) АВ — декремент. Из операнда вычитается 1.

5. NEG(B) АВ — инверсия знака. Число заменяется на имеющее противоположный знак и равное по абсолютной величине, в дополнительном коде. Если выполнить команду NEG R4 (R4 тот же, что и в предыдущем примере), получим 0 100 010 011 001 000 (восьмеричное 042310).

Таким образом, оператор выполняет операцию *дополнения до двух*: $AB + NEG(AB) = \langle \text{все нули} \rangle + \text{перенос}$.

6. TST(B) А — проверка содержимого операнда. Операнд при этом не меняется, а только устанавливаются биты условий ССП в соответствии с его содержимым. Если операнд равен нулю, то устанавливается равным единице бит Z, если операнд отрицателен, то N. Иногда этот оператор применяется для инкремента (или декремента) содержимого регистра на 2; например, команда $TST(R2) +$ увеличивает на 2 содержимое R2, причем делает это быстрее, чем две команды INC R2, и занимает в ОЗУ всего одно слово, однако работает только в том случае, если содержимое регистра указывает на доступную для чтения память (иначе произойдет прерывание по вектору 4).

7. ASL(B) АВ — арифметический сдвиг влево. Слово (или байт) операнда сдвигается поразрядно влево, старший разряд пересылается в бит C ССП, а вместо младшего разряда заносится ноль. Пусть, например, в регистре R2 слово 1 011 101 100 111 000 (восьмеричное 135470). После выполнения команды ASL R2 в R2 будет 0 111 011 001 110 000 (восьмеричное 073160), разряд C=1.

Оператор применяется для умножения на 2 и других целей.

8. **ASR(B) AB** — арифметический сдвиг вправо. Слово (или байт) операнда сдвигается поразрядно вправо, младший разряд переписывается в бит С ССП, старший разряд не меняется. Если содержимое R2 то же, что и в предыдущем примере, то после выполнения команды **ASR R2** получим 1 101 110 110 011 100 (восьмеричное 156634), разряд С=0.

Оператор применяется для деления на 2 и других целей.

9. **ROL(B) AB** — циклический сдвиг влево. Слово (или байт) операнда сдвигается поразрядно влево, старший разряд переходит в разряд С ССП, а предшествующее содержимое С переходит в младший разряд операнда. Слово (или байт) как бы замыкается при этом в кольцо, причем замыкающим звеном является бит С ССП, и происходит вращение кольца на один разряд.

64

10. **ROR(B) AB** — циклический сдвиг вправо. Слово (или байт) операнда сдвигается поразрядно вправо, младший разряд переходит в разряд С ССП, а предшествующее содержимое С переходит в старший разряд операнда. Аналогично предыдущему оператору, но с вращением в другую сторону.

Два последних оператора могут применяться помимо сдвига содержимого операндов для поразрядной перезаписи одного операнда в другой через разряд С. Например, если выполнить команды,

```
1) ROL R2      2) ROL R2      3) ROR R2      4) ROR R2
   ROR R4      ROL R4      ROL R4      ROR R4
```

то в примере 1 старший бит содержимого R2 окажется переписан в старший бит R4, в примере 2 — старший бит R2 в младший бит R4, в примере 3 — младший бит R2 в младший бит R4 и, наконец, в примере 4 — младший бит R2 в старший бит R4. Выполняя такие сочетания команд несколько раз подряд, можно как угодно «вывертывать наизнанку» слово или его часть, выделять нужное число разрядов в другое слово и т. д.

11. **ADC(B) AB** — прибавление разряда переноса. К операнду прибавляется содержимое бита С ССП. Применяется для учета переноса при сложении, умножении, вычислении контрольных сумм, а также для других целей.

12. **SBC(B) AB** — вычитание разряда переноса. Из операнда вычитается содержимое бита С ССП. Применяется для учета переноса при вычитании, делении, а также для других целей.

13. **SWAB AB** — перестановка байтов. Старший и младший байты операнда меняются местами.

14. **SXT AB** — расширение знака. Всем

разрядам операнда (слова) присваивается значение бита N, следовательно, знакового разряда слова, если N установлен корректно. В результате операции слово может стать равным или 0 (если исходное число было положительным, точнее, если N был равен 0), или 177777 (исходное число — отрицательное, N=1).

15. **MFPS B** — чтение ССП. Младший байт ССП записывается по адресу операнда. О структуре ССП подробно сказано в разделе, посвященном описанию центрального процессора. Применяется очень редко, в основном для сохранения текущего ССП с целью его последующего восстановления.

16. **MTPS A** — запись ССП. Младший байт слова-операнда записывается в регистр PS процессора. Применяется чаще всего для задания необходимого уровня приоритета процессора, реже — для восстановления ССП. Не может устанавливать бит Т.

С двумя операндами

17. **MOV(B) A,B** — пересылка. Содержимое операнда А записывается (копируется) по адресу операнда В. Исходное содержимое А не меняется, исходное содержимое В теряется. Этот оператор, особенности и примеры его работы уже были подробно рассмотрены. **MOV** — наиболее часто применяемый из всех операторов ассемблера.

18. **BIC(B) A,B** — очистка разрядов по маске. Во втором операнде обнуляются биты (разряды) по маске первого. Смысл этого действия следующий: во втором операнде (слове или байте) устанавливаются равными нулю те разряды, которым соответствуют единицы в первом операнде. Приведем пример. Пусть в регистре R2 слово

```
0 100 001 111 010 100,
```

а в регистре R4 слово

```
0 110 100 000 100 111.
```

Выполним команду **BIC R2, R4**, получим в регистре R4 слово

```
0 010 100 000 100 011.
```

Те биты R4, которым соответствуют единицы R2, обнулились, остальные не изменились.

Приведем еще один пример. После выполнения команд

```
MOV    #177777,R4
BIC    #177400,R4
```

содержимое R4 станет равно 377 (выделен младший байт R4).

19. **BIS(B) A, B** — установка разрядов по маске. Это предыдущий оператор «наоборот» — во втором операнде устанавливаются равными единице те разряды, которым соответствуют единицы в первом операнде. Например, с теми же исходными данными, что

в первом примере, относящемся к предыдущему оператору, результатом выполнения команды **BIS R2,R4** будет слово

0 110 101 111 110 111,

а последовательность команд

CLR R4
BIS #377,R4

даст содержимое R4, равное 377.

20. BIT(B) A1,A2 — проверка битов. Проверяются биты (разряды) второго операнда по маске первого. Оба операнда не меняются, устанавливаются только биты условий ССП. Смысл операции следующий. Если хотя бы одному разряду первого операнда, равному единице, соответствует единица в том же разряде второго, результат не нулевой. Если же каждой единице первого операнда соответствует во втором ноль, результат нулевой. Чаще всего оператор используется для проверки отдельных битов, для чего в первом операнде задается маска, в которой присутствует только одна единица (в соответствии с проверяемым разрядом второго). Например, если нажата любая клавиша, в системном регистре 177716 разряд 06 будет равен 0, в противном случае — 1. Для выявления нажатия клавиши можно применить команду **BIT#100,@#177716**. Если клавиша нажата, результат исполнения команды равен нулю, и наоборот.

21. CMP(B) A1,A2 — сравнение. Операнды сравниваются друг с другом, и по результатам сравнения устанавливаются биты условий ССП. Сами операнды не меняются. Сравнение осуществляется псевдовычитанием второго операнда из первого, т. е. как бы осуществляется действие (A1—A2). Если, например, операнды равны, то результат операции — ноль (Z=1), если второй операнд больше первого, то результат отрицательный (N=1) и т. д. Оператор применяется очень широко как для сравнения операндов, так и для инкремента (или декремента) регистров общего назначения (если в них записано число, указывающее на доступную для чтения память), причем позволяет модифицировать содержимое регистра сразу на 4. Например, **CMP (R2) +,(R2) +** увеличивает R2 на 4, меняет биты условий и занимает всего одно слово. Но при таком действии необходимо помнить, что биты условий не имеют никакого отношения к содержимому регистра, и обнаружить после такой команды, например, произошло ли переполнение R2, невозможно.

22. ADD A,B — сложение. Складывается содержимое операндов (слов), результат за-

писывается во второй операнд, первый не меняется, т. е. выполняется присваивание $V=B+A$. Операция широко применяется для сложения и умножения.

23. SUB A,B — вычитание. Из второго операнда вычитается первый, результат записывается во второй операнд, первый не меняется, т. е. выполняется присваивание $V=B-A$. Необходимо всегда помнить, что порядок вычитания оператора **SUB** (из второго — первый) обратен псевдовычитанию оператора **CMR** (из первого — второй). Широко применяется для вычитания и деления, иногда для сравнения.

24. XOR RN,B — поразрядное «исключающее ИЛИ». Выполняется логическая функция «исключающее ИЛИ» между соответствующими разрядами (битами) первого и второго операндов; результат заносится во второй операнд. Первый операнд может быть только регистром общего назначения, второй адресуется любым способом. Что такое «исключающее ИЛИ»? Это функция, фиксирующая несовпадение битов, так называемая «функция *неравнозначности*». Поясним это таблицей значений функции:

65

Первый бит	Второй бит	Результат
0	0	0
0	1	1
1	0	1
1	1	0

Такая операция оператором **XOR** выполняется между всеми соответствующими разрядами операндов, в результате чего образуется число, дающее не разность, а как бы «разницу» между операндами. Одинаковые операнды дают в результате ноль, взаимно инверсные — число 177777.

Зачем нужна такая функция? Она дает возможность «наслаивать» одно число на другое, например формировать изображение на экране на фоне другого изображения с сохранением последнего. Если вывести то же изображение на то же место вторично с помощью функции **XOR**, то оно исчезнет, а фон восстановится. Аналогично можно «наслаивать» звуки в многоголосных мелодиях, воспроизводимых одним динамиком, и т. п.

Приведем пример. Пусть в R2 записано число 0 101 110 111 001 011, а в R4 — число 1 001 100 111 101 011. Тогда после выполнения команды **XOR R2,R4** в R4 будет число 1 100 010 000 100 000.

Продолжение следует.

Принципы работы БК-0010

Бытовой микрокомпьютер БК-0010 пользуется заслуженной популярностью. К сожалению, отсутствие литературы по его устройству и принципам работы не позволяет владельцам полностью использовать все его скрытые возможности. Он особенно пригоден для управления различными внешними устройствами, автоматизации производственных и технологических процессов. В его конструкции заложены богатые возможности расширения памяти, подключения различных периферийных устройств. Цель настоящей

статьи — помочь любознательным владельцам БК-0010, желающим разобраться в работе этого микрокомпьютера, использовать его не только для игры.

БК-0010, как и любой современный микрокомпьютер, состоит из следующих основных узлов: микропроцессора, запоминающего устройства и устройств ввода-вывода информации. В БК-0010 использован однокристалльный микропроцессор типа K1801BM1A. Это вполне современный микропроцессор, имеющий 16-разрядную мультиплексирован-

Назначение выводов микропроцессора K1801BM1A

№	Обозначение	Функциональное назначение
1	CLC	Вход тактовых импульсов (от 100 кГц до 5 МГц)
2	SACK	Вход поддержания прямого доступа к памяти (ПДП)
3	SP	Вход резервный
4	DMGO	Выход: разрешение ПДП (процессор отключен)
5	DMR	Вход: запрос внешнего устройства на ПДП
6	SP	Вход резервный
7	SEL1	Выход: низкий уровень при обращении по адресу 177716
8	SEL2	Выход: низкий уровень при обращении по адресу 177714
9	ADO	Вход/выход: адрес-данные 0
10	AD1	Вход/выход: адрес-данные 1
11	AD2	Вход/выход: адрес-данные 2
12	AD3	Вход/выход: адрес-данные 3
13	AD4	Вход/выход: адрес-данные 4
14	AD5	Вход/выход: адрес-данные 5
15	AD6	Вход/выход: адрес-данные 6
16	AD7	Вход/выход: адрес-данные 7
17	AD8	Вход/выход: адрес-данные 8
18	AD9	Вход/выход: адрес-данные 9
19	AD10	Вход/выход: адрес-данные 10
20	AD11	Вход/выход: адрес-данные 11
21	OV	Общий вывод
22	AD12	Вход/выход: адрес-данные 12
23	AD13	Вход/выход: адрес-данные 13
24	AD14	Вход/выход: адрес-данные 14
25	AD15	Вход/выход: адрес-данные 15
26	SP	Вход резервный
27	SP	Вход резервный
28	BSY	Выход: признак занятости канала
29	DCLO	Вход: авария источника питания
30	ACLO	Вход: пропадание сетевого напряжения
31	IRQ1	Вход: запрос на прерывание
32	IRQ2	Вход: запрос на прерывание по таймеру (адрес 100)
33	IRQ3	Вход: запрос на прерывание (адрес 270)
34	INIT	Вход/выход: установка исходного состояния
35	VIRQ	Вход: запрос на прерывание
36	IAKO	Выход: подтверждение прерывания
37	DOUT	Выход: вывод данных из процессора
38	DIN	Выход: признак ввода данных в процессор
39	RPLY	Вход: ответ внешнего устройства на DOUT и DIN
40	WTBT	Выход: признак вывода байта
41	SYNC	Выход: синхронизация адреса на шине A
42	+VCC	Питание +5В, 240 мА

Примечание: низкий уровень — активный (логическая 1).

ную шину «адрес-данные», производительность порядка 500 000 элементарных операций в секунду, развитую систему команд. Он предоставляет пользователю широчайшие возможности создания на его основе систем для решения самых различных задач — от телеигр до автоматизированного управления небольшими промышленными и сельскохозяйственными предприятиями.

Системная шина БК-0010 (разъем МПИ)

Системной шиной, или каналом, компьютера называется совокупность линий (проводов или дорожек), несущих информационные и управляющие сигналы. Она предназначена для связи различных узлов компьютера между собой, а также с внешними устройствами.

Микропроцессор К1801ВМ1А, используемый в БК-0010, формирует системную шину с протоколом обмена, соответствующим стандартному межмодульному параллельному интерфейсу (МПИ).

Контакт	Обозначение	Функциональное назначение
A31	AD0	Адрес-данные 0
B31	AD1	» 1
B29	AD2	» 2
B30	AD3	» 3
B28	AD4	» 4
A28	AD5	» 5
B27	AD6	» 6
B32	AD7	» 7
B26	AD8	» 8
A27	AD9	» 9
B25	AD10	» 10
A26	AD11	» 11
B24	AD12	» 12
A25	AD13	» 13
B23	AD14	» 14
B07	AD15	» 15
B22	SYNC	Синхронизация адреса
A23	DIN	Ввод данных
B21	DOUT	Вывод данных
B20	RPLY	Подтверждение ввода/вывода
B11	WTBT	Запись байта
A4, A, B12	+5B	Питание +5В, 150 мА
A, B2, A, B3	GND	Общий вывод

Запуск микропроцессора К1801ВМ1А

В момент включения питания на входах DCLO и ACLO должен быть низкий уровень. На выходе INIT сразу после подачи +5В устанавливается низкий уровень. Не менее чем через 5 мс после включения питания на входе DCLO должен быть уста-

новлен внешним устройством высокий уровень, и еще через 70 мс или более высокий уровень должен быть установлен на входе ACLO. В этот момент на выходе INIT устанавливается низкий уровень, при этом происходит установка внешних устройств в исходное состояние. После перевода DCLO в неактивное состояние разряды 0—7 счетчика команд микропроцессора очищаются, а разряды 8—15 загружаются содержимым соответствующих разрядов регистра 177716. В слово состояния (PS) заносится константа 340. Далее анализируются запросы на прерывания. Если их нет, происходит выполнение команды в соответствии с содержимым счетчика команд.

Обмен данными

Возможны три вида обмена: ввод, вывод, ввод-пауза-вывод (считывание-модификация-запись).

1. Ввод. Процессор устанавливает BSY и выдает адрес на шину AD. После установки адреса процессор выдает отрицательный перепад напряжения (скачкообразный переход от высокого уровня к низкому) на выход SYNC. По этому перепаду внешнее устройство должно защелкнуть адрес. Затем процессор снимает адрес* и устанавливает DIN. По этому сигналу внешнее устройство должно установить данные на AD и после этого установить RPLY. Если RPLY не будет установлен в течение 64 тактовых импульсов после установки DIN, произойдет прерывание по зависимости (адрес 4). Приняв RPLY, процессор вводит данные и снимает DIN. В ответ на снятие DIN внешнее устройство должно снять RPLY. После снятия RPLY процессор снимает SYNC и BSY. WTBT не вырабатывается.

2. Вывод. Аналогично, только вместо DIN устанавливается DOUT после предварительной выдачи данных на AD. Внешнее устройство после приема данных должно установить RPLY. Приняв RPLY, процессор снимает данные и DOUT; далее — как в цикле «ввод». Сигнал WTBT вырабатывается при записи байта.

3. Ввод-пауза-вывод. Производятся ввод, обработка данных и их вывод без изменения адреса. Адрес и SYNC выдаются только в начале цикла, при вводе данных.

Прямой доступ к памяти

Для достижения высокой скорости обмена данными между внешним устройством и оперативной памятью компьютера используется

* Т. е. отключает шину, в результате чего на ней оказывается некий неопределенный сигнал.

режим прямого доступа к памяти (ПДП). При этом микропроцессор отключается от системной шины компьютера и обмен данными производится под управлением внешнего устройства, обычно контроллера ПДП.

В начале цикла ПДП внешнее устройство устанавливает низкий уровень на входе DMR (требование ПДП), в ответ микропроцессор устанавливает низкий уровень на выходе DMGO (предоставление ПДП). В ответ на DMGO внешнее устройство должно установить SACK и снять DMR. После этого внешнее устройство производит обмен данными, самостоятельно генерируя управляющие сигналы SYNC, DIN, DOUT. По окончании цикла ПДП внешнее устройство снимает SACK, после чего управление системной шиной возвращается к микропроцессору.

68 Прерывания от внешних устройств

Ввод в компьютер данных от внешних устройств может производиться методом программного периодического опроса этих устройств, а также по прерыванию. Второй способ эффективнее, поскольку при этом микропроцессор не «отвлекается» от выполнения основной задачи на опрос внешних устройств. После получения команды (запроса) прерывания выполнение основной программы прекращается и микропроцессор переходит к выполнению программы, адрес которой находится в специальной ячейке памяти, называемой адресом вектора прерывания (см. таблицу назначения выводов K1801BM1). Микропроцессор K1801BM1 имеет пять входов для запросов прерываний от внешних устройств: IRQ1, IRQ2, IRQ3, VIRQ, ACLO. Установка седьмого разряда регистра состояния (PS) запрещает (маскирует) прерывания по VIRQ, IRQ2, IRQ3.

На входы IRQ2, IRQ3 запросы на прерывания подаются путем подачи на них отрицательного перепада (динамические входы), на остальные — путем установления на них низкого уровня.

Прерывание по VIRQ производится следующим образом: внешнее устройство устанавливает низкий уровень на этом входе. В ответ микропроцессор запоминает в стеке содержимое счетчика команд и регистра состояния, затем последовательно устанавливает низкий уровень на выходах DIN и IAKO. Внешнее устройство в ответ на IAKO снимает запрос на входе VIRQ, устанавливает на шине «адрес-данные» адрес вектора прерывания (по выбору пользователя), устанавливает RPLY. В ответ на RPLY микропроцессор снимает DIN и IAKO, после чего внешнее устройство освобождает шину «ад-

рес-данные» и снимает RPLY. Микропроцессор переходит к обработке прерывания.

Вход IRQ1 использован в БК-0010 для прерывания по клавише СТОП. Адрес вектора прерывания — 4.

Вход IRQ2 имеет специальное назначение — это вход прерывания по таймеру. Пользователи БК-0010 могут использовать этот вход для привязки выполняемой программы к реальному времени, например вести часы в программу. Для этого следует подать прямоугольные импульсы TTL-уровня на контакт В1 разъема внешнего порта от внешнего генератора секундных импульсов. В ячейку 100 предварительно нужно занести адрес программы обработки прерывания, которую лучше составить на языке ассемблера.

Вход IRQ3 в БК-0010 не используется и соединен с шиной питания.

Запоминающее устройство БК-0010

Запоминающее устройство БК-0010 состоит из оперативного запоминающего устройства (ОЗУ) и постоянного запоминающего устройства (ПЗУ). ОЗУ доступно как по записи, так и по чтению, а ПЗУ — только по чтению. ОЗУ БК-0010 имеет информационную емкость 32К байт, при этом область от 0 до 377 используется для системных программ, от 400 до 1000 — для стека, от 1000 до 40000 — для программ пользователя. Область ОЗУ от 40000 до 100000 называется экранной областью, каждый ее бит отображается на экране в виде точки. В области от 100000 до 120000 расположено системное ПЗУ с драйвер-мониторной системой, от 120000 до 177600 — ПЗУ с Бейсиком. Область от 177600 до 177777 включительно предназначена для системных регистров. Здесь следует отметить, что возможности БК-0010 можно значительно расширить, подключив внешнее ОЗУ в область от 120000 до 177600 вместо имеющегося ПЗУ.

ОЗУ БК-0010 реализовано на 16 микросхемах типа КР565РУ6Г. Эти микросхемы являются динамическими, т. е. могут сохранять записанную в них информацию только в течение двух миллисекунд, после чего необходимо восстановление (регенерация) информации. В БК-0010 регенерация информации производится во время вывода изображения на экран. Эту сложную функцию выполняют специальная большая интегральная микросхема (БИС) типа K1801ВП1-037, называемая контроллером ОЗУ и видеомонитором, и 16-разрядный регистр, составленный из двух микросхем типа K589ИР12.

ПЗУ БК-0010 реализовано на БИС КР1801РЕ2Б. На корпусе БИС, содержащей драйвер-мониторную систему, стоит номер 017, Фокал — 018, мониторную систему диагностики — 019, Бейсик размещен в трех микросхемах с номерами 106, 107, 108. ПЗУ с номерами 018 и 019 расположены в подключаемом блоке МСТД.

Устройства ввода-вывода (УВВ)

К УВВ БК-0010 относятся: клавиатура (ввод), видеомонитор (вывод), магнитофон (ввод-вывод). Кроме того, конструкцией предусмотрен внешний порт, позволяющий производить обмен информацией с другими внешними устройствами, например с принтером, джойстиком, АЦП и ЦАП, дисководом, бытовым радиокомплексом и др.

Клавиатура БК-0010 представляет собой матрицу 10×7 линий, на пересечении которых стоят замыкающие кнопки. Имеются также кнопки, подключенные отдельно (соответствуют клавишам ЗАГЛ, СТР, ПРОБЕЛ, СУ, ПР, НР, ИНД СУ, СТОП). В БК-0010 имеется специальная БИС типа К1801ВП1-014, называемая контроллером клавиатуры. Эта БИС в ответ на нажатие любой клавиши формирует семиразрядный код КОИ-7, соответствующий нажатой клавише, и помещает его в свой регистр с адресом 177662, называемый регистром данных клавиатуры. БИС содержит еще один адресуемый регистр с адресом 177660, называемый регистром команд клавиатуры. Одновременно с записью кода клавиши в регистр данных клавиатуры БИС устанавливает запрос на прерывание на входе VIRQ микропроцессора. Адрес вектора прерывания — 60 (274 по AP2). Прерывания от клавиатуры можно программно запрещать. Внимание: БИС К1801ВП1-014 чувствительна к статическому электричеству, поэтому, садясь за компьютер, прежде чем прикоснуться к клавишам, необходимо разря-

даться об трубу отопления или водопроводную трубу!

Вывод изображения на экран видеомонитора осуществляется при помощи контроллера видеомонитора, содержащего БИС К1801ВП1-037 и 16-разрядный регистр сдвига, реализованный на двух микросхемах типа К155ИР13. Содержимое ячеек экранного ОЗУ поочередно записывается в регистр в параллельном коде, затем происходит сдвиг этой информации с частотой 6 МГц и она побитно поступает на вход видеосуилителя видеомонитора. Вывод этой информации синхронизируется строчными и кадровыми синхрои импульсами, генерируемыми БИС К1801ВП1-037.

Сигнал с магнитофона при чтении поступает на вход двухтранзисторного усилителя-ограничителя, а затем на вход А1 микросхемы К531АП2 (шинный формирователь), которая передает его на системную шину при чтении процессором адреса 177716. Вход А0 (вывод 4) этой микросхемы предназначен для ввода информации с телеграфной линии. В БК-0010 первых выпусков он был выведен на контакт В2 разъема внешнего порта, с 1986 г. это соединение отсутствует, но его легко восстановить самостоятельно. Сигнал на магнитофон при записи выдается с микросхемы К155ИР1 в двоичном коде в виде последовательности «длинных» и «коротких» импульсов, подобно точкам и тире азбуки Морзе. При этом комбинация «тире-точка» соответствует единице, а «три точки» — нулю. Одновременно с выводом на магнитофон происходит вывод на внутреннюю «пищалку» ЗП-1. Вывод 13 К155ИР1 предназначен для вывода на телеграфную линию. Его желательно соединить с контактом В3 разъема внешнего порта, как это было сделано в БК-0010 первых выпусков.

А. СОКОЛОВ

Конвертор Бейсик-ассемблер для БК

Создатели ПЗУ с интерпретатором Бейсика для БК-0010.01, тщательно засекретив «начинку» своего изделия, вынудили пользователей дублировать зашитые в Бейсик-ПЗУ функции в мизерном адресном пространстве доступной им памяти. Между тем пользователи, программирующие в кодах или

на ассемблере, уже давно ищут ключ к этим пэззушкам, съедающим почти половину всей памяти: ведь если научиться использовать хранящуюся в них информацию в качестве подпрограмм, то можно существенно сократить объем своей программы, что для БК немаловажно.

Постепенно общими усилиями Бейсик-ПЗУ становится все более доступным — свидетельством тому является статья Авсеевых «Особенности транслятора с языка Бейсик для БК-0010.01», напечатанная в «ИНФО» № 2 за 1990 г. Авторы приводят очень интересную информацию о методике поиска адресов подпрограмм Бейсик-ПЗУ. Подобная методика была разработана и мной; разными были только цели поиска. Если у авторов статьи конечной целью поиска адресов является «ручное» составление шитых кодов обращения к Бейсик-подпрограммам, то моей целью было по возможности «автоматизировать» этот трудоемкий процесс. Ведь в конечном счете пользователю вовсе не нужно знать адреса подпрограмм, нужно лишь получить цепочку шитых кодов, исполнение которых даст тот же результат, что и написанный на Бейсике фрагмент.

С этой целью была написана программа COD.ASC, позволяющая получить на внешнем носителе шитый код введенной Бейсик-программы в формате MICRO10 (MICRO11). В исходной программе разрешается использовать любые переменные, однако на идентификаторы целочисленных переменных накладывается ограничение: они могут быть только однобуквенными. Это вызвано тем, что в диалоге между кодовой программой и подпрограммой на Бейсике для передачи параметров удобно использовать именно такой тип переменных. Программа распознает обращения к ним и заменяет адреса переменных в рабочей памяти Бейсика на идентификаторы. При трансляции программы произойдет увязка ссылок на эти переменные с одноименными полями основной ассемблерной программы. Все двух- и более символьные целочисленные идентификаторы будут усечены до одного символа. Переменные других типов могут быть любыми, допустимыми Бейсиком.

Программа COD.ASC приведена в традиционной Бейсик-кодовой форме и рассчитана на работу в составе КУВТ-86. Если предполагается использование магнитофона в качестве внешнего носителя, то строку 1280 следует заменить

```
1280 DATA &0104036,&0240,&0207,
      &0117224,&0106221,&0214
```

Порядок работы с программой приводится ниже.

После ввода директив

```
NEW
CLEAR,&037000
```

введите с клавиатуры текст программы, тщательно проверьте его и запишите на внешний

носитель командой SAVE «имя». Теперь начиная со строки 30 введите текст Бейсик-программы, код которой хотите получить. Программа не должна содержать операторов CLS. Если нужно очистить экран, используйте команду PRINT CHR\$(12). Запустите программу командой RUN. Экран очистится, и в левом верхнем углу появится угольник — приглашение к вводу шифра фрагмента. В качестве этого шифра может выступать любой алфавитно-цифровой символ. Он будет последним символом имени полученного модуля и внутренних меток. Подготовьте устройство вывода. Нажмите клавишу, соответствующую выбранному шифру фрагмента. Экран опять очистится, и в верхней его части появится полоса — это в экранной области памяти создается исходный модуль ассемблера, содержащий шитый код программы. Он выводится на внешний носитель с именем IJQCD*.BIN (* — шифр фрагмента), и работа COD.ASC на этом заканчивается.

Полученный таким образом исходный модуль можно оттранслировать и использовать объектный модуль при создании программ. Можно в режиме LF подключить его к основной программе. В обоих случаях возможно редактирование полученного текста.

Пример: составить программу на ассемблере, выводящую на экран в восьмеричном виде содержимое области памяти с адреса 0 до адреса &O400. Для печати восьмеричного представления чисел использовать фрагмент Бейсик-ПЗУ.

Пример приводится для варианта с КУВТ-86.

Здесь с начала строки идут приказы оператора, а 16-й позиции ответы БК, с 20-й — комментарии автора.

```
NEW                               Ok
CLEAR,&037000                       Ok
LOA"TT:COD                          COD .ASC 000
30 PRINT OCT$(NX);TAB(8X);OCT$(PEEK(NX))
RUN                                  очистка экрана
>
A                                    на МД записан файл
                                      IJQCD.A.BIN
BL"TT:MICR11",R
Micro 11 >
SC                                   вошли в текстовый
                                      редактор
BEGIN: CLR N
10:   JSR PC,FRINOC
      ADD #2,N
      CMP N,&400
      BLE 10
      HALT
```



```

N1      .E
CY+>
LF
IJCDA
SC      length=000305

```

```

к тексту добавилось:
MOV #JWA,R4
JMP @(R4)+
JZA:  RTS FC
JWA:  .#156232
      .#N
      .#160556
      .#161624
      .#156460
      .#156700
      .#JZA

```

```

меняем первую строку
добавленного
фрагмента:

```

```

PRINOC: MOV #JWA,R4
CY+>

```

```

Micro 11 >

```

```

CN
LS14000
STOP

```

```

вышли в монитор

```

```

EВВОД

```

К сожалению, не всегда удается использовать полученные коды без редактирования, как в нашем примере. Программа не производит пересчет внутренних адресов шитого кода (например, в командах GOTO, IF, GOSUB и т. д.). Кроме того, не всегда происходит распознавание целочисленных переменных. Поэтому лучше считать эту программу не частью самостоятельной методики, а инструментом для облегчения ручного создания программ в кодах Бейсика.

В заключение — несколько наблюдений и рекомендаций для тех, кто пользуется кодами Бейсика.

1. Бейсик при своей работе широко использует стек. Особенно «агрессивна» в этом отношении команда PAINT. Старайтесь по возможности выделять под стек больше памяти.

2. Для нормальной работы Бейсика не забудьте в начале программы занести в ячейку 34 значение 146404. Этот адрес программы обработки прерываний позволит запустить вашу программу не только из Бейсика функцией `USR`, но и из монитора.

3. Постарайтесь расположить свою программу «пониже» в памяти. Младшие адреса (до 3500) могут использоваться Бейси-ком.

4. Интересные результаты можно получить, если предусмотреть в цепочке кодов Бейсика обращение к своим подпрограммам на

ассемблере. Для этого в желаемом месте цепочки укажите адрес своей подпрограммы. Выход из нее осуществляйте по команде `JMP@ (R4) +`.

Тем, кого эта публикация побудила к творчеству, автор желает больших успехов. А людей, небезразличных к программированию на ассемблере, и, в частности, с использованием Бейсик-ПЗУ, приглашаю к переписке. Мой адрес: 244004, Сумы, ул. Горького, 36, кв. 97. *Ингорь А. И.*

```

10 GOTO 1000
20 GOX=0X
1000 CLS
1010 NX=&037000
1020 READ SX
1030 POKE NX,SX
1040 NX=NX+2
1050 IF NX<>&037672 THEN 1020
1060 DEF USR=&037000
1070 A=USR(A)
1080 END
1090 DATA &012700,&076,&0104016,
&0104006,&0110067,&0547
1100 DATA &0110067,&0576,&0110067,
&0616,&0110067,&0632
1110 DATA &0110067,&0277,&012701,
&037352,&05002,&0104020
1120 DATA &05067,&0464,&012767,
&040000,&0462,&012701
1130 DATA &037546,&04767,&0430,
&012701,&037572,&04767
1140 DATA &0420,&012701,&037614,
&04767,&0410,&012701
1150 DATA &037640,&04767,&0400,
&012703,&037232,&012367
1160 DATA &0414,&011367,&0412,
&012713,&02240,&012743
1170 DATA &0240,&013705,&02002,
&032705.1,&01401
1180 DATA &05205,&062705,&014,
&010504,&014467,&0346
1190 DATA &022715,&0156776,&01432,
&026727,&0332,&0156232
1200 DATA &01465,&026727,&0322,
&0156350,&01461,&012567
1210 DATA &0310,&016767,&0304,&0304,
&010546,&04767
1220 DATA &0162,&04767,&0176,
&012703,&037232,&016723
1230 DATA &0270,&016713,&0266,
&012605,&0743,&012701
1240 DATA &037654,&04767,&0224,
&012702,&0320,&010201
1250 DATA &012722,2,&012722,
&040000,&016712,&0224
1260 DATA &0162722,&040000,&012722,
&045111,&012722,&041521
1270 DATA &012722,&025104,&012722,
&041056,&012722,&047111
1280 DATA &04737,&0124650,&0207,
&0117224,&0106221,&0214
1290 DATA &04767,&034,&011501,
&0114100,&04767,&0110
1300 DATA &012700,&012,&04767,
&0100,&012567,&0122
1310 DATA &016767,&0116,&0116,
&0664,&012701,&037432
1320 DATA &04767,&066,&0207,
&020007,&021456,&0

```

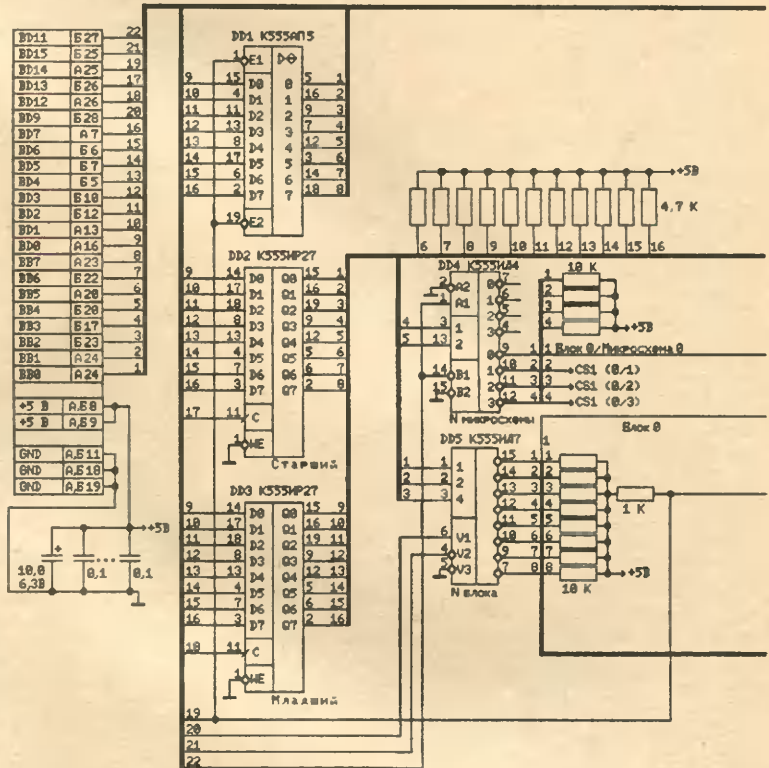

1330 DATA &012704, &037474, &0134,
 &012602, &012601, &0112100
 1340 DATA &04767, &024, &077204,
 &012700, &012, &04767
 1350 DATA &012, &0207, &0156232,
 &037532, &0161624, &037446
 1360 DATA &0110077, &030, &05267,
 &024, &0207, &0112100
 1370 DATA &01403, &04767, &0177756,
 &0773, &0207, 0
 1380 DATA 0, 0, 0, 0, 0, &020007
 1390 DATA &047515, &020126, &020040,
 &020040, &045043, &025127

1400 DATA &051054, &05064, 0, &020007,
 &046512, &020120
 1410 DATA &020040, &020040, &024100,
 &032122, &025451, &012
 1420 DATA &055112, &035052, &020040,
 &020040, &052122, &020123
 1430 DATA &020040, &020040, &041520,
 &012, &053512, &035052
 1440 DATA &020040, &020040, &021456,
 0, &020007, &021456
 1450 DATA &055112, &05052, 0, &025132, &012

А. ИНГОР

«Электронный диск»
 для БК-0010

72



Есть несколько способов увеличения оперативной памяти БК: можно разместить дополнительное ОЗУ в области адресов, занимаемых встроенным ПЗУ с интерпретаторами Бейсика или Фокала (адреса с 120 000 до 177 600), или организовать дополнительную память в виде внешнего накопителя — электронного квазидиска. Авторами разработаны такой квазидиск и его программное обеспечение.

Электронный квазидиск подключается к порту ввода-вывода БК и состоит из 32 микросхем K537PY8 и схемы обрамления на пяти микросхемах средней степени интеграции. Его емкость — 64К байта, она

может быть увеличена до 128К байт простым увеличением количества микросхем ОЗУ. При разработке электрической схемы квазидиска ставилась задача максимально упростить аппаратную часть.

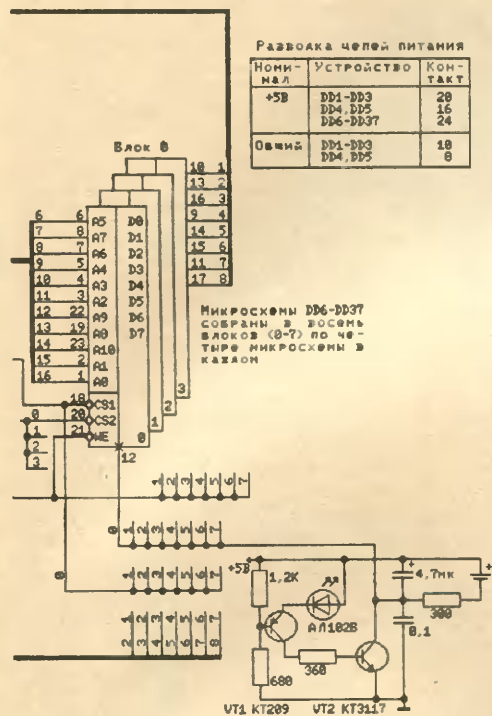
Микросхемы DD2 и DD3 образуют регистр адреса выбираемой ячейки памяти. Микросхема DD1 — буфер данных при записи информации. Микросхемы DD4, DD5 образуют дешифратор адреса для выбора нужной микросхемы ОЗУ. Устройство, собранное на транзисторах VT1 и VT2, служит для перехода на резервное питание от аккумулятора (3 В) при отключении компьютера.

Конструктивно квазидиск выполнен на

двухсторонней печатной плате размером 95×178 мм. Ток, потребляемый от автономного питания в режиме хранения информации, не превышает 600 мкА.

Для поддержки работы с квазидиском была разработана специальная операционная система КДОС, позволяющая любым программам работать с квазидиском, как с обычным магнитофоном, и выполняющая следующие функции:

переключение драйвера магнитофон/квазидиск;



загрузка с квазидиска и запуск системных программ;

загрузка с квазидиска и запуск программ пользователя;

стирание файлов на квазидиске;

выдача списка файлов на квазидиске;

инициализация квазидиска.

Объем, занимаемый операционной системой, не превышает 2К байт. В системное обеспечение входят также текстовый редактор, микроассемблер, дизассемблер, микроотладчик, графический редактор.

Полугодовой опыт работы с квазидиском показал, что хотя по объему хранимой информации квазидиск и уступает магнитофону, но по удобству работы и оперативности значительно превосходит его.

С. КУМАНДИН, А. СОКОЛОВ

Для владельцев БК, впервые выписавших наш журнал в этом году, сообщаем содержание «Клуба БК» за прошлый год.

В. Авсеев, А. Авсеев. Особенности транслятора с языка Бейсик для БК-001.01

использование зашитых в ПЗУ подпрограмм, реализующих операторы Бейсика

Д. Антонов. Увеличение тактовой частоты простейшая переделка, вдвое увеличивающая быстродействие БК

А. Барсуков. Убыстрение поиска файла

А. Бочаров. Хитрости Бейсика БК

В. Булитко. Вмешательство в работу программы

о методах введения данных в программу на Фокале «параллельно» ее работе

А. Бунцельман. Программа «Музыкальная заставка»

В. Василенко. Спасение программ при зависании

описание способа восстановления текста Фокал-программы после зависания

С. Гвоздев, Г. Эрнстсон. Еще раз о плавающей арифметике Фокала

Ю. Зальцман. Архитектура и ассемблер БК

С. Ивашинников. Многоголосие на БК-0010 музыкальная программа в кодах

Ингорь. Использование квазиподпрограмм оригинальный подход к оформлению повторяющихся частей программы, экономия ОЗУ

Калейдоскоп

объявления; повышение устойчивости синхронизации телевизора «Электроника Ц401М»; «маленькие хитрости»

И. Канивец. Знаете ли вы, что... о системных ячейках БК

С. Комаров. Строки вместо матриц об использовании псевдоцветов

А. Кузнецов. Еще раз о восьмицветном БК использование псевдоцветов путем простейшей переделки БК

Кумандин. Новый подход к построению баз данных

М. Ларкин. Экономия памяти БК-0010.01

Ю. Луцкевич. Подключение БК к телевизорам 4УСЦТ

И. Панченков. Приручение БК советы для начинающих (но не только!) пользователей БК, схема БК

С. Смирнов. Восьмицветная приставка реализация реального многоцветия без изменений в схеме БК

Советы и наблюдения

А. Тереховский. Универсальный каталогизатор

Е. Умников. Спрайты для БК

П. Чирков. Подключение БК к телевизорам ЗУСЦТ

В. Яковлев. Отсчет времени на БК-0010.01 (поправка в № 2 на с. 53)

использование встроенного таймера БК; методика оценки эффективности программ по критериям времени исполнения и занимаемой памяти

Однако у такого способа есть важное ограничение: ни в одном из байтов подпрограммы не должно находиться число &O12, интерпретируемое компилятором как код окончания строки: все находящееся после этого байта будет проигнорировано.

Зависимость положения адреса кодовой программы, записанной в ремарку, от процедуры редактирования любой предшествующей ей строки может быть использована для защиты программ от несанкционированных исправлений или изменений.

Защищать программу имеет смысл только после ее окончательной отладки, и первым шагом к этому является ввод в начало программы обращения к подпрограмме в кодах, например:

```
1 DEF USR=9999
2 X=USR(X)
```

Затем уже описанным способом резервируется в памяти место для подпрограммы, а в конец программы добавляются три строки для определения зарезервированной об-

ласти. После определения этой области туда записывается код &O207 (т. е. команда RTS PC), удаляются три последние строки программы, введенные для определения зарезервированной области памяти, и программа пересылается на ДВК с расширением .COD, где с помощью программы DESS девятки в строке 1 исправляются на действительный адрес подпрограммы. Теперь любая попытка редактирования текста программы неизбежно приведет к изменению начального адреса подпрограммы и, следовательно, к отказу в ее работе.

Нужно заметить, что такая защита становится еще более надежной, если кодовая программа необходима для работы основной программы, а не содержится в себе просто код возврата, как в приведенном примере.

Описанная здесь методика защиты Бейсик-программ носит общий характер и может быть (с точностью до незначущих технических деталей) распространена на компьютеры любых других типов.

О. ПАРСАДАНОВ, А. ЯНУШКО

75

Калейдоскоп

Хотел бы переписываться и обмениваться программами с владельцами БК. Имею 250 различных программ.

232007, Вильнюс, ул. Маргитес, 29, кв. 13.
П. И. Дейчко.

Сенсация!

По мнению А. Куликова (г. Курган), полной очистки памяти БК не происходит ни в каких случаях, кроме выключения питания! Ни оператор NEW, ни перезапуск процессора, ни переходы в монитор и обратно — ничто не способно «обеспамятить» БК.

Но что же делать, если БК завис и был перезапущен процессор? Не теряться! Если программа была в машинных кодах и располагалась в диапазоне адресов 4000—40000, смело запускайте ее. Если же программа была на Бейсике, то некоторые повреждения она получила, но основную часть спасти можно. Для этого сперва нужно восстановить значение ячейки 2002, где хранился адрес последнего байта текста программы. Затем нужно восстановить ячейку 2024, являющуюся счетчиком строк: для этого в нее нужно поместить число $36756 - 6 \times N$, где N — число строк, а 36756 — содержимое ячейки 2024 при верхней границе памяти 40000. При этой процедуре возможна утра-

та части текста программы, вызванная тем, что при восстановлении ячейки оператором РОКЕ транслятор создает соответствующий ему объектный код, попадающий на текст программы. Если возникают проблемы с восстановлением ячейки 2002, то восстановите ячейку 2024, наберите

```
DEF USR=&O140716
```

и

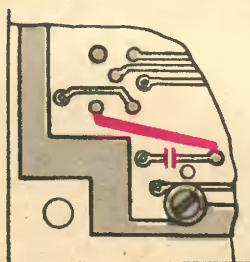
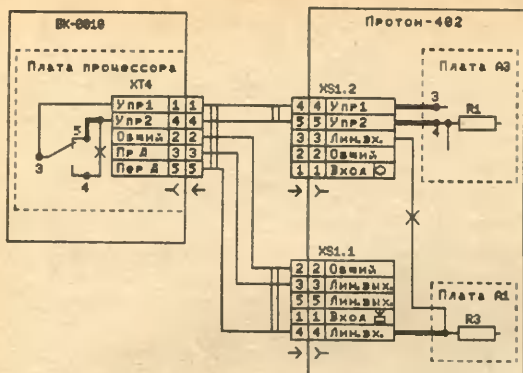
```
?USR(0)
```

— восстановление произойдет автоматически.

Доработаем наши магнитофоны (обзор предложений читателей)

Преимущества работы с магнитофонами, имеющими дистанционное управление, уже обсуждались на страницах журнала. Сегодня от их констатации мы переходим к публикации конкретных рекомендаций.

С. К. Максимов (г. Североморск) предлагает произвести в магнитофоне «Протон 402» и компьютере БК-0010(-01) достаточно простые и ясные из приводимых рисунков изменения (на схеме новые соединения показаны жирными линиями, ненужные



старые зачеркнуты). Они обеспечат малый ток управления (3 мА), щадящий контакты реле, и возможность работы с магнитофоном в автономном режиме без каких-либо дополнительных заглушек или переключателей, восстанавливающих штатный режим работы.

Р. Аскеров (г. Брянск) предлагает в магнитофоне «Электроника 302-2» разорвать цепь питания двигателя магнитофона, перепаяв провод, идущий к контакту 2 микропереключателя SA1 (от контакта переключателя SA2) на контакт 4 стандартной 5-штырьковой розетки, установленной вместо гнезда подключения внешних громкоговорителей, а контакт 5 этой розетки соединить с контактом 2 SA1. Теперь цепь питания двигателя магнитофона замкнется только при срабатывании пускового реле БК. Для работы магнитофона в автономном режиме параллельно контактам 4 и 5 розетки нужно подключить какой-либо выключатель. Для этого удобно использовать микропереключатель типа МТ-1, расположив его за декоративной решеткой встроенного микрофона.

Рекомендации по оформлению программного обеспечения ПЭВМ БК-0010/0010.01

Продолжая тему «приручения БК», предлагаем вниманию читателей рекомендации по стандартизации оформления программного обеспечения и (в следующем номере) интер-

фейса БК. Рекомендации составлены на основе опыта многих пользователей; хотелось бы узнать мнение наших читателей.

И. ПАНЧЕНКОВ

1. При написании программ

1. Программы, написанные на любом языке, должны автоматически устанавливать необходимый режим работы: 32/64, РП, КУРСОР и т. п. Для написания и модификации программ на Фокале используйте программу начальной установки STARTF.

2. Программы в машинных кодах должны иметь автозапуск (исключение допускается для инструментальных программ). При этом должен осуществляться контроль правильности загрузки.

3. Для программ в машинных кодах обязательно должен быть возможен запуск из пускового монитора.

4. Обращение к мониторам разрешается только через запросы ЕМТ (для обеспечения переносимости на другие машины семейства). Программы должны одинаково работать как при наличии МСД-монитора, так и без него.

5. Программы в машинных кодах должны обрабатывать прерывание по клавише СТОП (вектор 4).

6. При использовании блокировки клавиши СТОП должен быть предусмотрен выход в систему по команде СУ/С (код 3).

7. По окончании (прерыванию) программы должен осуществляться возврат в исходную систему, из которой она была запущена.

8. Номер версии или дата ее создания должны выводиться на экран при запуске программы. Желательно наличие в программе ФИО автора (авторов) или названия авторского коллектива, телефона, адреса или других координат для связи.

9. Программы должны иметь инструкцию по пользованию (документацию) либо диалог, позволяющий работать с программой без документации.

10. Документация на программу должна содержаться:

в самой программе (для программ, рассчитанных на работу с цветным изображением, она должна быть написана в режиме 32 символа в строке);

или в отдельном файле на том же языке, что и документируемая программа;

или в непосредственно исполняемом файле в машинных кодах (пользуйтесь такими инструментальными программами, как FORMAT, DOCUM, MONSTR).

11. При задании клавиш управления необходимо учитывать удобство их использования

в машинах как с пленочной, так и с «высокой» клавиатурой.

12. Подключение внешних устройств к порту параллельного ввода-вывода производится в соответствии с прилагаемым стандартом интерфейса.

13. При написании игровых программ рекомендуется использование унифицированного блока установки управления STAND или аналогично ему (например, как в LODE RUNNER).

II. При присвоении имен

1. Имена файлов пишутся заглавными латинскими буквами.

2. Длина присваиваемого имени не должна превышать шести знаков, после которого через точку могут следовать три символа расширения. Имя в возможно большей степени должно отражать содержание программы.

Не следует пренебрегать названиями программ на английском языке — они зачастую бывают информативнее русских. Используйте предлагаемый набор стандартных расширений.

.APL прикладная программа
.ASC стандартный Бейсик (Вильнюс 24.07.86): блоки ASCII кодов
.B27 ленточная версия Бейсик 27
.B85 ленточная версия Бейсик 85
.B87 ленточная версия Бейсик 87
.BIN стандартный Бейсик (Вильнюс 24.07.86): двоичный формат
.BSE файл в формате программы BASE (ASP согр.)
.C файл в формате C
.COD стандартный Бейсик (Вильнюс 24.07.86): внутренний формат
.DAT файл данных
.DCM файл с системой просмотра DOCUM (ASP согр.)
.DEM демонстрационная программа
DOC документация
.EDP текст в формате редактора EDASP (ASP согр.)
.FCD файл с системой FOCOD
.FND файл с системой FOND
.FOC файл в формате FOCAL
.FRM файл с системой просмотра FORMAT/FORMAT2

.FRT файл в формате FOPTH
.G.. игровая программа (общий признак семейства)
.GD. игровая программа динамическая
.GG. игровая программа графическая
.GL. игровая программа логическая
.GM. игровая программа музыкальная
...K игровая программа с управлением от клавиатуры
...J игровая программа с управлением от джойстика
...U игровая программа с унифицированным управлением
.INF информационная программа
.IWF файл с системой «Интерфейс Встроенной Функции»
.MAN управляющая программа
.MBS файл в формате программы MICRO-BASE (ASP согр.)
.MEL файл в системе MELOMAN/MELOMAN2
.MIC исходный текст программы в формате MICRO# ...S
.MIK исходный текст программы в формате MICRO# ...K
.MST файл с системой просмотра MONSTR (В. Прохоров)
.MUS музыкальная программа
.OBM объективный модуль MICRO
.RED файл с неизвестной системой просмотра
.RID файл с системой просмотра READER
.RIG файл с системой RIGA
.SAN загрузочный модуль MICRO
.SCH учебная программа
.SPH речевая программа
.SYS системная программа
.TCH обучающая программа
.TST тестирующая программа
.TXT текстовая программа
.USR пользовательская программа в кодах
.WOR рабочая версия для всех типов файлов
.XFC файл с системой XFOCAL (OSOO10F)

77

Расширения различных версий Фокала пишутся без разделительной точки сразу после имени файла. Расширений для стандартных версий Фокала БК-0010 не предусматривается.

Пользователи учебной вычислительной техники на базе ПЭВМ «Агат», БК-0010, УКНЦ («Электроника» МС 0202, МС 0511):

Чебоксарское научно-исследовательское предприятие «Логось», занятое развитием новых информационных технологий обучения, извещает о новых информационных услугах на 1991 г.

Разовое обслуживание

Мы предлагаем широкий выбор педагогических программных средств, включающих программную поддержку школьных курсов информатики, математики, физики, химии, истории, биологии и других предметов; математическое моделирование естественных и гуманитарных явлений; новые прикладные, системные и инструментальные средства, графические игры.

Цены весьма умеренны: от 5 до 300 руб. Мы с удовольствием вышлем подробный каталог программ бесплатно.

Информационное обслуживание

Мы открываем на 1991 г. подписку на «Логось-информатор». В шести выпусках подписчики получают на шести дискетах:

учебные материалы по операционным системам, языкам программирования (включая «хитрое» программирование), известным прикладным и системным средствам; математические модели естественных и гуманитарных явлений; решения задач по информатике различной степени трудности; подробно откомментированные и методически значимые учебные листинги; описание архитектуры вашей ПЭВМ; оригинальные программы; интересные материалы по информатике; рекламные объявления, каталоги программных и аппаратных средств, справочные материалы, новости из мира компьютерного бизнеса; индивидуальные консультации специалистов.

Полная подписка может быть оформлена в течение всего 1991 г.
Стоимость подписки — 300 руб.

Программное приложение к подписке на информационное обслуживание

Подписчиков ждут оригинальные коммерческие продукты предприятия «Логось», педагогические программные средства по различным предметам, новые системные и прикладные программы, рекламные продукты, новые версии уже известных на рынке программ.

Шесть выпусков в год на шести дискетах. Стоимость подписки на 30 % ниже, чем суммарная стоимость входящих в нее программных продуктов.

Полная подписка может быть оформлена в течение всего 1991 г.
Стоимость подписки — 500 руб.

Бесплатное информационное обслуживание

Для частных лиц и организаций мы предлагаем также подписку на бесплатное информационное обслуживание из фонда материалов, переданных нам для некоммерческого использования.

Чтобы стать нашим «бесплатным» клиентом, достаточно сообщить нам о своем желании получать такую услугу и указать точный почтовый адрес.

Вы будете получать материалы (шесть дискет за год), оплатив только стоимость дискет и накладные расходы.

Полная подписка может быть оформлена в течение всего 1991 г.

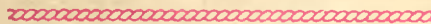
Оформление заказа

Для оформления подписки на любой вид обслуживания следует направить в наш адрес письмо-заказ, указав в нем вид выбранного обслуживания, название вашей ПЭВМ, тип используемого на ней дисковод, точный почтовый адрес, а также ФИО лица, ответственного за подписку.

Работа с авторами

Предприятие заключает договоры с авторами — разработчиками программных продуктов для указанных ПЭВМ на тиражирование, размещает в выпусках оригинальные статьи, методические материалы, объявления рекламного характера, некоммерческие материалы, оставляя за собой право отбора материалов, помещаемых в выпуск. Гарантируется возврат присланных магнитных дискет и соблюдение авторских прав.

Адрес для заказов и справок:
428034, Чебоксары, а/я 99, предприятие «Логосъ».



Всем пользователям терминальных классов «Корвет»

Научно-техническое предприятие «Альтернатива» объявляет о создании пакета учебных программ для классов «Корвет», предназначенного для обеспечения преподавания курса «Основы информатики и вычислительной техники» в средних учебных заведениях.

В составе пакета:

пересылаемая по сети операционная система CP/M;
языки программирования Паскаль, Си, Форт, Бейсик, Ассемблер;
экранные редакторы текстов;
графические редакторы;
электронные таблицы;
система управления базами данных типа dBASE II;
клавиатурные тренажеры;
пакет игровых программ.

Все программы написаны в кодах ЭВМ и укомплектованы подробной технической документацией.

Гарантируются высокое качество программ, их пересылаемость по сети и работоспособность на рабочем месте ученика.

По составу, назначению и условиям поставки данный пакет аналогичен пакету «Альтернатива» для КУВТ-86 (см.: Информатика и образование. 1990. № 3. С. 55).

Заказы высылать по адресу:
656099, Барнаул, пр. Социалистический, д. 60, каб. 15.

ЧТО? МОЖЕТ ЭВМ

Стоит бросить взгляд...

...и компьютер, угадывая ваше желание, перейдет к новому режиму работы или нарисует замысловатую линию. Ненавязчиво следить за движением ваших глаз ему позволит видеотрэккер — устройство массой около 140 г, которое фиксируется на шлеме или голове с помощью ленты. Видеотрэккер позволяет определять не только направление взгляда, но и скорость его перемещения, и степень внимания к объекту.

Помимо компьютера, трэккер можно объединить с фотоаппаратом или видеокамерой для фиксации объектов пристального наблюдения, а точности прибора достаточно, чтобы обеспечить работу системы прицеливания.

Маленький, да удаленный

Именно так можно охарактеризовать новый разъем японской фирмы Mitsumi Electric. Судите сами: при размерах всего 20×18×10 мм он имеет пропускную способность 6 миллионов бит в секунду. Хотя главное, конечно, не размеры.

Предлагаемый фирмой разъем предназначен для коммутации светового луча, передаваемого по световолоконному кабелю. А это одно из самых уязвимых и сложных мест оптических линий связи. Вот теперь можно по праву восхититься габаритами разъема и относительно невысокой ценой 214 долларов за комплект, в который входят гнездо и штеккер с гибким пятиметровым стекловолоконным кабелем.

Преобразование оптического сигнала в стандартный электри-

ческий с уровнем TTL-микросхем производится серийной интерфэйсной платой.

Остается добавить, что разъем разработан специально для использования в автомобильных цифровых аудиосистемах.

Видеотрэккер пока еще достаточно дорог и используется в основном в военной авиации, но со временем можно надеяться на появление простых и изящных трэккеров и на головах программистов.

Так почему же?

Цены определяются рынком: чем он обширнее, чем больше на нем продавцов и покупателей, тем объективнее цены. Поэтому даже за барьером неконвертируемости информация с бостонской компьютерной биржи представляет интерес.

Итак: IBM PC без винчестера — от 250 до 660 долларов; IBM PC XT, винчестер 20М байт — от 475 до 825 долларов; IBM PC AT с винчестером 20М байт — от 700 до 1375 долларов; IBM PS/2 модель 30 с винчестером 20М байт (такие ПЭВМ, только без винчестера, используются в качестве РМУ в недавно закупленных у фирмы IBM КУВТак) — от 1025 до 1385 долларов; IBM PS/2 модель 50 с винчестером 30М байт (РМП в тех же КУВТак) — от 1550 до 2000 долларов. Лазерный принтер фирмы «Хьюлетт-Паккард» Laser-Jet II стоил от 850 до 1050 долларов.

Что реально означают эти цифры? Это поможет понять цитата из западного журнала: «Многие пользователи ПК мучительно размышляют, на чем остановить свой выбор при покупке нового компьютера, — то ли на дешевой машине с микропроцессором 80286, то ли на более мощном, но дорогом компьютере с микропроцессором 80386». Для ясности: ПЭВМ с микропроцессором 80286 — это машина типа IBM PC AT; ее рыночная стоимость в СССР — около 50 тыс. руб.

Впрочем, обладатели валюты могут недорого купить компьюте-

ры не только в Бостоне, но и в Москве. Например, тайваньская компания Well Join Industry Co., Ltd. предлагает XT-совместимые ПЭВМ за 500—600 долларов, AT-совместимые — за 1100 долларов.

И компьютер, и телефон

Новая интегрированная система разработана японской фирмой Hitachi. Все устройства, входящие в ее состав, объединены с помощью шарниров в один изящный и очень компактный конструктивный узел.

Основой системы является персональный компьютер PROSET, имеющий цветной монитор, жесткий диск объемом 40М байт и два дисководы для 3,5-дюймовых 1,2М-байтных гибких дисков. Кроме того, в общий блок встроены принтер, телефон и система сетевой поддержки, позволяющая обмениваться информацией с внешним миром по каналу со стандартным протоколом со скоростью до 10 миллионов бит в секунду.

Система выпускается в двух вариантах стоимостью 5.159 и 4.469 долларов.

Цитата

«Пороку декларируется, что Паскаль был разработан как язык для обучения. Хотя это и правильно, его использование в обучении не было единственной целью. В действительности я не верю в использование в обучении таких средств и формализмов, которые не годятся для какой-нибудь практической задачи. По сегодняшним стандартам Паскаль имеет очевидные недостатки для программирования больших систем, но 15 лет назад он представлял собой разумный компромисс между тем, что было желанно, и тем, что было эффективно».

Никлаус Вирт.

Из выступления при получении премии Тьюринга.

И. ЗАБАРА, С. РАКОВ

Харьковский педагогический институт

Тренажер «DIANA»

Одно из наиболее распространенных направлений использования компьютера в обучении — работа с педагогическими программными средствами [1]. В статье описывается интеллектуальный тренажер «DIANA» [2], позволяющий отработать технику дифференцирования у учащихся. Эта программа используется при изучении темы «Производная» в Харьковском педагогическом институте и на факультативных занятиях со школьниками.

Для автоматизации этого навыка необходимо, чтобы каждый ученик самостоятельно решил большое количество задач, в которых были бы задействованы все типы элементарных функций. Такая цель может быть достигнута, если у учителя будет возможность работать с каждым учеником индивидуально, подбирая ему соответствующие задания. Наблюдая за ходом рассуждений ученика, учитель указывает ошибки, объясняет, как следовало бы решать тот или иной пример, определяет сложность и тип следующей задачи. Таким образом, вскрывая пробелы в знаниях и подробно разъясняя применение каждого правила на практике, учитель формирует устойчивый навык дифференцирования.

Если организовать проведение части занятий при изучении темы «Производная» в дисплейном классе, то функции преподавателя, индивидуально работающего с учеником, может взять на себя компьютер. В нашем институте разработана программа-тренажер «DIANA» [1] для организации практических занятий по теме «Производная». Она работает в следующих режимах: экспертном, контролирующем, генерирующем. Программа «DIANA» имеет систему меню, позволяющую пользователю выбрать необходимый режим работы.

На программу возлагаются функции подбора заданий, оценивания правильности ответов, консультирования и демонстрации правильного решения.

Режим работы программы выбирается учителем в зависимости от типа урока.

Экспертный режим

Если цель учителя — разбор конкретных примеров к изложенному на предыдущем занятии теоретическому материалу, то используется экспертный режим работы: ученик вводит в компьютер функцию из набора, составленного учителем, и просматривает процесс дифференцирования, выполняемый компьютером. Задания можно расположить в порядке возрастания сложности. Количество решенных учеником примеров в течение одного урока и уровень их сложности зависят от усвоения. Просмотр проводится в удобном для ученика темпе, предусмотрена возможность многократного повторения просмотров.

Но такой режим усвоения знаний является пассивным, и на следующем этапе учитель может видоизменить задание: ученику предлагается самостоятельно продифференцировать функцию, сверить результат с компьютерным и только после этого полностью просмотреть процесс решения, выполняемый компьютером. При такой организации в работу вносится элемент состязательности, повышается интерес ученика как к работе с компьютером, так и к изучаемому предмету [2, 3].

Контролирующий режим

Учитель может провести контрольную работу по теме «Производная» (или по каким-

либо ее разделам) в дисплейном классе. Для этого в тренажере предусмотрен контролирующий режим. Задания, которые будут предъявляться ученикам, заранее помещаются в память компьютера (имеется возможность подбора индивидуального варианта для каждого ученика и случайной генерации заданий).

Во время контрольной работы программа дает ученику для выполнения каждого задания три попытки. Ответ вводится в произвольной форме и проверяется программой: сравниваются значения в трех случайно выбранных точках. Если все три попытки ученика были безуспешны, компьютер демонстрирует процесс дифференцирования заданной функции (с возможностью многократных повторений). Если ученик ответил правильно, программа, сообщив об этом, предлагает посмотреть решение, выполняемое компьютером. Ученик может воспользоваться возможностью просмотра, а может (так как на контрольной работе он ограничен во времени) позднее посмотреть компьютерное решение в экспертном режиме. Результаты работы (все попытки) заносятся в память и могут быть просмотрены преподавателем на экране или выведены на принтер.

Для облегчения проверки результатов компьютера, записывая ответ ученика, указывает, является ли данный результат верным. Окончательно оценивает контрольную работу учитель.

Режим автоматической генерации заданий

В состав тренажера входит программа, которая по указанным параметрам сложности и трудоемкости генерирует функции определенного класса.

Структура тренажера «DIANA»

Программа-тренажер «DIANA» содержит блок символьного дифференцирования, использующий базу знаний, состоящую из

правил дифференцирования и таблицы производных;

блок синтаксического контроля, основу которого составляют правила синтаксиса языка Паскаль, дополненные операцией возведения в степень и группой элементарных функций — \lg , \arccos , $\sqrt{\quad}$, sh и др.);

блок проверки правильности ответов (сравнение ответа ученика с ответом, полученным компьютером);

блок генерации заданий по заданным параметрам (указывается, какие элементарные функции должны входить в учебный пример, какова степень вложенности и т. д.).

Программа «DIANA» разработана для «Ямахи» (в отличие от известной системы АСОД [4], не имеющей реализации на персональном компьютере). В настоящее время ведутся работы по созданию более совершенной версии, включающей блок тождественных преобразований, для IBM PC (отсутствующий в аналогичной программе DIFF, поставляемой в составе демонстрационного пакета к языку Пролог версии 2.0).

Литература

1. *Забара И., Раков С.* Методические рекомендации по использованию комплекса программ «DIANA» при изучении курса «Алгебра и начала анализа» для преподавателей и студентов педагогических институтов, учителей математики, старшекласников. Харьков.: ХГПИ, 1990 (на укр. яз.).
2. *Машибуц Е.* Психологические основы управления учебной деятельностью. Киев.: Вища школа, 1987.
3. *Нокс Д.* Что могут дать компьютеры педагогике: взгляд из американской школы // Информатика и образование. 1990. № 1.
4. *Раатс Ю., Толмачева А.* Адаптивная система обучения дифференцированию // Автоматика и вычислительная техника. 1980. № 3.

В. УШАКОВА г. Горно-Алтайск, ГАГПИ
Ю. ВОЛОДИН, Ю. МАРДАШЕВ
Москва, МПГУ им. В. И. Ленина

Программа «Допуск»

Важный аспект использования компьютера в учебном процессе — организация эффективной деятельности преподавателя при подготовке и проведении занятий. Правильно составленная учебная контролирующая программа позволяет каждому студенту рабо-

тать в темпе, обусловленном типом его темперамента и уровнем подготовки, освобождает дополнительное время для занятий с хорошо успевающими студентами.

В настоящее время в Горно-Алтайском педагогическом институте и в МПГУ

им. В. И. Ленина начались разработка и формирование программно-методического обеспечения для занятий по физической химии.

При существующей системе обучения преподаватель тратит значительную часть времени на опрос студентов, для того чтобы определить их готовность к предстоящей практической или лабораторной работе.

Разработанная нами программа «Допуск»* позволяет за 5 мин опросить до 10 студентов и интенсифицировать проведение практических работ, сместив акцент на качество эксперимента, обработку его результатов, их анализ.

Программа реализована на языке Бейсик (операционная система РАФОС) для ДВК-2М и ДВК-3М.

Программа организована так, что после выполнения одной лабораторной работы (например, «Измерение ЭДС гальванического элемента») блок данных, объединенных оператором DATA, может быть заменен на аналогичный блок по другой теме («Потенциометрические измерения» или «Изучение кинетики химических реакций» и т. д.). Причем замена одного блока данных на другой не трудоемка. Достаточно набрать новые данные на клавиатуре, сохранив номер строки и оператор DATA.

Программа «Допуск» состоит из двух частей. Первая — проверка теоретического материала. Она содержит блок вопросов, которые задаются студенту с помощью генератора случайных чисел, т. е. порядок вопросов при повторном запуске программы (при контроле каждого следующего студента) меняется. Студенту задается один вопрос и сразу даются пять ответов, из которых он должен выбрать верный. Существенно, что номера верных ответов также рандомизированы, что практически исключает возможность передачи информации между студентами.

Каждый верный ответ фиксируется счетчиками. После этого опроса студент получает оценку с комментарием и разрешение перейти ко второму заданию — второй части программы «Допуск», которая включает справочник необходимых физико-химических данных и задачник. Студенту предлагается решить три задачи. По желанию преподаватель может уменьшить или увеличить количество задач или заменить их другими. Если студенту требуется та или иная спра-

вочная величина, он вызывает на дисплей справочник.

Студент должен решить задачу и ввести в машину номер правильного ответа. Счетчик верных ответов зафиксирует, верно или неверно ответил студент, после чего ему будет предложена вторая задача, а затем третья. Если студент справился с решением всех или двух задач, то при условии положительной оценки за первую часть работы он будет допущен к выполнению практической работы по теме. В противном случае машина после соответствующего комментария отправляет его к списку литературы, используя которую студент может здесь же, во время занятия, подготовиться для повторного контроля-допуска.

Для удобства текст задач может быть дополнительно напечатан на карточках и выдан студенту одновременно с выводом его на дисплей. Это удобно тем, что при вызове справочника текст задачи «уходит» с экрана и остаются только тексты ответов. В таком случае студент может обращаться к тексту задачи на карточке.

Решение задачи (верное или неверное) должно быть представлено преподавателю для анализа ошибок. Это исключает возможность случайного выбора верного номера ответа. После заключения машины о допуске и просмотра преподавателем решения задач в тетради студент выполняет практическую работу.

Программа легко может быть адаптирована для других типов ЭВМ.

Программа «Допуск» разработана как методическое пособие не только для физической химии. Ею может пользоваться преподаватель любой химической дисциплины, физики, биологии и других негуманитарных предметов. Для этого достаточно изменить данные в программе, приготовить карточки с текстом задач или упражнений. Данные обновляются после вывода программы на дисплей прямо с клавиатуры ЭВМ, при этом программе присваивается новое имя. Предыдущая программа сохраняется записанной на дискету и может быть повторно использована. В соответствии с текстом новых данных обновляется и справочник. Может быть расширен и конкретизирован список литературы.

Желающие ознакомиться с программой могут обратиться по адресу: 119021, Москва, Несвижский пер., 3, химический факультет МПГУ им. В. И. Ленина, кафедра физической и аналитической химии.

* Программа является частью программно-методического обеспечения занятий по физической химии, разрабатываемого Горно-Алтайским педагогическим институтом и МПГУ им. В. И. Ленина.

Программно-аппаратный комплекс сопряжения УКНЦ с нестандартным периферийным оборудованием ПАКС —

мощный активизатор учебного процесса.

Если ваши ученики плохо осваивают алгоритмизацию; несмотря на все усилия, относятся к ПЭВМ как к игрушке; если вы сами хотите глубже изучить особенности использования ЭВМ для управления процессами —

используйте ПАКС!

Он позволяет наглядно, в диалоговом режиме, с использованием меню создавать алгоритмы управления устройствами, причем с использованием обратной связи, а главное, «оживить» создаваемые программы. Для этого к УКНЦ подключаются до пяти устройств с питанием от сети 220 В (лампа, магнитофон, электродвигатели и т. п.) и абонентский громкоговоритель для вывода компьютерной речи.

Базовый вариант ПАКС содержит:

- блок сопряжения пяти внешних устройств с УКНЦ (расширяется до 15 каналов);
- один канал дискретного вывода (уровень ТТЛ);
- один канал дискретного ввода (уровень ТТЛ);
- устройство вывода компьютерной речи;
- интерактивную программу взаимодействия с пользователем.

84

Программное обеспечение позволяет задавать конфигурацию подключаемого оборудования; с помощью меню выбирать блоки алгоритма и задавать выполняемое действие или условие (алгоритм представлен в виде блок-схемы); контролировать шаг алгоритма, выполняющийся в данный момент, приостанавливать и возобновлять исполнение алгоритма.

Предусмотрены блоки расширения (их номенклатура постоянно растет) для сопряжения с телефоном, диапроектором и т. д.

Обмен файлами между ЭВМ линии DEC — *это гибкость вашего класса*

Программно-аппаратный комплекс позволяет обмениваться файлами между ПЭВМ ДВК-2М, ДВК-3, «Электроника МС 0511», «Электроника-85». Со стороны ПЭВМ типа ДВК используется устройство последовательного обмена (УПО), выполненное на БИС КР580ВВ51; «Электроника МС 0511» соединяется с УПО двумя линиями «витая пара» через оптронную гальваническую развязку.

Скорость обмена — 19200 бод, длина линии связи — до 500 м.

CasGen —

это генерация Бейсик-системы на УКНЦ с магнитофона. CasGen страхует вас на случай неисправности дискового.

Для IBM-совместимых машин — программы «Школьное расписание» и «Классный журнал». При любом числе классов в школе — максимальная простота составления расписания и контроля успеваемости.

Если вам нужны ППС, инструментальные и системные программные средства для

КУВТ-86

«Корвета»

УКНЦ

КУВТ «Русич» —

обращайтесь к нам.

Школьный кооператив «Интерфейс».

309250, Белгородская обл., г. Шебекино, а/я 36.

Тел.: (8-072-48) 4-17-36.

«Уральские» задачи

Как же сделать проявитель?

В одном журнале Петя Косичкин нашел заметку, где подробно описывались характеристики проявителя Р-6726 с замечательными свойствами — утверждалось, что он повышает светочувствительность фотопленки в 200 раз! Подробно описывался и способ его применения. К сожалению, способ приготовления Р-6726 приводился в следующем номере журнала, который Пете найти не удалось. В данной же заметке лишь приводился перечень реактивов (с весами) и говорилось, что проявитель — их раствор в 1 л воды.

Поскольку в таком проявителе у Пети по некоторым обстоятельствам была большая необходимость, он решил поэкспериментировать. Попытка растворить компоненты Р-6726 в воде в произвольной последовательности показала, что все не так просто и безопасно: в жидкости образовались бурые хлопья, и с бульканьем стал выделяться какой-то душающий газ. Чтобы не взорваться и не отравиться (да и реактивы зря не переводить), Петя решил обратиться к опыту предков — порыться в книгах. И поиски привели к некоторому успеху.

В одном из справочников в разделе «Водные растворы» для некоторых из компонентов Р-6726 приводился список веществ, которые «непреренно должны быть в растворе при растворении данного реактива» (все эти вещества входили в состав Р-6726). Кроме того, для некоторых из компонентов проявителя давался перечень веществ с указанием: «при растворении данного реактива не допускается присутствие в растворе ни одного из перечисленных веществ». Видимо, нарушение какого-то (или каких-то) из этих условий, отсутствующее при «фирменном» способе, и привело к неудаче эксперимента.

В условии использована математическая задача И. Ф. Сивергиной об упорядочении последовательности.

Итак, требуется следующее.

1. Построить разумную, на ваш взгляд, математическую модель процесса определения такого способа приготовления проявителя, который удовлетворял бы приведенным в справочнике условиям и ограничениям.

2. Разработать алгоритм определения допустимой последовательности растворения реактивов.

3. Если алгоритм не может найти решения, разобраться, что это значит. Ведь авторы же как-то готовят Р-6726!

Математическая модель

Поскольку имеющаяся в нашем распоряжении информация, как это обычно и бывает, не отличается большой точностью и строгостью, уточним для себя некоторые формулировки.

Пусть $P = \{X_n, n = 1, \dots, N\}$ — множество веществ, составляющих проявитель. Приведенные в справочнике условия обозначим как

$$R_i: X_i \\ R_i = \{X_{p(i,1)}, X_{p(i,2)}, \dots\} \quad i = 1, \dots, I, \quad (1)$$

где «A: B» означает, что наличие в растворе всех веществ из множества A делает возможным растворение вещества B, и

$$X_j/S_j \\ S_j = \{X_{q(j,1)}, X_{q(j,2)}, \dots\} \quad j = 1, \dots, J, \quad (2)$$

где «A/B» означает, что наличие в растворе любого вещества из множества B исключает растворение A.

Будем полагать, что все вещества проявителя простые в смысле данной задачи и имеет место ситуация независимости: процесс растворения любого из веществ не зависит от присутствия в растворе реактивов, не указанных ни в одном из двух списков для него.

Если речь идет о последовательности растворения, то каждому реактиву соответствует некоторый номер в этой последовательности. Если веществу X_i соответствует номер l_i в некоторой допустимой последовательности, то условия (1) перейдут, естест-

венно, в условия

$$l_{p(i,1)} < l_i, l_{p(i,2)} < l_i, \dots, i=1, \dots, I \quad (3)$$

и

$$l_j < l_{q(j,1)}, l_j < l_{q(j,2)}, \dots, j=1, \dots, J, \quad (4)$$

где $<$ — обычное числовое неравенство.

Таким образом, задача свелась к следующей математической. Имеется некоторое множество P неизвестных чисел a_1, a_2, \dots, a_N . Имеется набор неравенств вида

$$a_{nk} < a_{nk}, k=1, \dots, K. \quad (5)$$

Требуется определить какой-нибудь гипотетический порядок элементов множества P , т. е. цепочку неравенств

$$a_{m(1)} < a_{m(2)} < \dots < a_{m(N)},$$

для которого были бы выполнены все условия (5). Разумеется, решение может оказаться не единственным либо отсутствовать. **Алгоритм решения**

Для решения предлагается алгоритм со следующей идеей. Пусть множество U первоначально совпадает с исходным множеством P . Пусть последовательность V первоначально имеет нулевую длину (пустая). Пусть множество N первоначально является набором неравенств (5).

Определим какой-нибудь элемент a из U , не присутствующий в правой части ни одного из неравенств N . Запишем a в конец последовательности V , исключим a из U и исключим из N все те неравенства, где a стоит в левой части. И будем так делать до тех пор, пока либо множество U не станет пусто (тогда последовательность V будет одной из искомым), либо процедура остановится из-за того, что в текущем еще непустом множестве U не окажется ни одного элемента, которого нет в правой части неравенств из множества N (тогда, утверждается, задача не имеет решения).

Доказательство правильности решения

Разумеется, как и при решении любой проблемы, требуется обоснование гипотезы о том, что такой алгоритм дает решение задачи. Прежде всего нетрудно убедиться, что для элементов текущей последовательности V не нарушается ни одно из неравенств (5), т. е. нет ни одного неравенства, где правый элемент оказался бы записан в V ранее, чем левый: ведь в V на каждом шаге добавляются элементы, не присутствующие в правой части ни одного из неравенств, у которых в левой части — элемент, уже принадлежащий V . Таким образом, если в конце работы алгоритма множество V будет содержать все N элементов исходного множества P , то V будет решением задачи (по крайней мере, одним из решений).

Предположим теперь, что на каком-то шаге алгоритм остановился «нехорошо», т. е. в еще непустом текущем множестве U не оказалось ни одного элемента, которого не было бы в правой части ни одного из неравенств из множества N . Это, разумеется, может означать одно из двух: либо задача действительно не имеет решения, либо у нас плохой алгоритм. Покажем, что причиной останова может быть лишь первое.

По построению на каждом шаге как в правых, так и в левых частях неравенств из N имеются лишь элементы из U . В то же время на шаге останова в U нет ни одного элемента, которого не было бы в правых частях N . Таким образом, при останове мы имеем в N систему неравенств, где в левых частях имеются элементы, образующие некоторое подмножество элементов, расположенных в правых частях. Такая система неравенств несовместна (противоречива). Действительно, если предположить иное, что существуют числа, удовлетворяющие всем этим неравенствам, то среди этих чисел найдется, по крайней мере, одно наименьшее. Но оно непременно будет присутствовать в правой части одного из неравенств, что противоречит выбору его как наименьшего (число в левой части этого неравенства будет еще меньше!). Противоречие доказывает, что сложившаяся на момент останова система неравенств N противоречива. Следовательно, поскольку текущая система неравенств N является некоторой подсистемой исходной, исходная система неравенств несовместна.

Таким образом, мы доказали, что останов нашего алгоритма до опустошения множества U произойдет тогда и только тогда, когда исходная система неравенств (5) несовместна, т. е. когда задача принципиально неразрешима. Следовательно, предлагаемый алгоритм решает поставленную задачу.

Осмысление алгоритма

Заметим, что правильность предложенного алгоритма неочевидна: в описании алгоритма слова «какой-нибудь элемент из U , не присутствующий в правой части ни одного из неравенств N » могут вызывать сомнения. Неужели при неоднозначности, при наличии альтернатив все равно, которую выбрать? Ведь в таких ситуациях существования альтернативности обычно при «плохом» завершении алгоритма приходится использовать перебор с возвратом, т. е. проверять все сочетания разрешенных альтернатив. Но здесь, как видим, перебор с возвратом делу не поможет: если алгоритм останавливается при каком-то выборе альтернатив, то он остановится и при любом другом (удивлены?). В то же время, если бы нас интересовала задача с каким-либо критерием оптимизации

ности (например, когда требуется найти подмножество Q исходного множества P, элементы которого удовлетворяли бы соответствующей подсистеме (5), и оно притом содержало бы наибольшее количество элементов по сравнению с другими такими подмножествами), то без перебора с возвратом мы бы уже, возможно, не обошлись.

Наш алгоритм позволяет, заметим, дать и большую информацию, чем требуется в постановке. Представьте, что мы будем на каждом шаге добавлять в V не один «какой-нибудь элемент из U», а сразу множество всех элементов из U, отсутствующих в правых частях неравенств текущего множества N! Тогда результатом работы модифицированного алгоритма будет последовательность «кучек» элементов P, где порядок кучек фиксируется, а порядок следования элементов в пределах одной кучки может быть любым по желанию.

Если вспомнить об исходной фотографической задаче, то этот последний модифицированный алгоритм не только может давать некоторую дополнительную свободу выбора, но и позволяет дать рекомендации по возможности растворения компонентов проявителя не по одному, а группами, что позволит как сократить время растворения, так и заранее фасовать компоненты не в индивидуальных пакетиках, а по нескольким, смешанных в одном пакете. Представьте, окажется, что состоящий из пары десятков веществ проявитель можно фасовать всего, скажем, в пару пакетиков, растворяя при приготовлении сначала первый, затем второй!

Кстати, а можно ли получить все допустимые последовательности элементов, манипулируя всевозможными последовательностями внутри кучек? И еще. Получим ли мы наименьшее возможное количество кучек? Формальная запись (спецификация) алгоритма (она же — программа)

Дадим формальную запись описанного выше (модифицированного) алгоритма на языке SETL (в нотации, в основном соответствующей версии для транслятора Д. Я. Левина, широко используемого в работах по искусственному интеллекту). Мы фактически приведем даже не только формальную спецификацию нашего алгоритма, а и программу для ЭВМ, позволяющую при наличии компьютера найти решение задачи «автоматически». Дело в том, что язык SETL не только является удобным алгоритмическим языком, но и реализован как язык программирования на ряде ЭВМ (БЭСМ-6, «Эльбрус», ЕС-10XX, IBM PC). Заметим к слову, что SETL был базой для разработки известного школьникам языка Рапира (который тоже реализован как язык программирования для

многих ЭВМ — «Агат», СМ-4, «Ямаха», УНКЦ, «Корвет»).

В приводимой записи алгоритма (программы) некоторые из использовавшихся выше обозначений заменены (в традициях программирования) на более читаемые многобуквенные имена (например, N на Ogr). Фигурные скобки в SETL означают множество, угловые — упорядоченную последовательность (кортеж); | означает при формировании множества «такое, что», : — примерно то же самое, но в других контекстах. Остальное, надеемся, будет и так понятно.

```

\X-----\X
\X Ввод исходной информации (ограничений) - множества из элементов-кортежей из элементов;
\X - множество номеров или названий тех веществ, которые необходимы для растворения Xi;
\X - номер или название Xi;
\X - множество номеров или названий тех веществ, которые допустимы при растворении Xi.
\X-----\X

```

```

_Нач
  Вывод ('Введите множество веществ');
  Ввод (Исх)
_Кон
_Цикл_До Множество (Исх) _И Исх+{ };

```

```

_Нач
  Вывод ('Введите множество ограничений - ',
        'см. описание алгоритма');
  Ввод (Инф)
_Кон
_Цикл_До Множество (Инф) _И Инф+{ };

```

```

\X\Введенная информация должна быть\X
\X по крайней мере множеством,\X
\X при том непустым \X

```

```

\X-----\X
\X---Сведение исходных ограничений---\X
\X-----к системе неравенств-----\X
\X-----\X

```

```

Ogr := ( ); \X Ogr - множество \X
\X\ограничений,\X
\X\пока пустое \X

```

```

_Для_всех A _Из Инф_Цикл
_Нач
  _Для_всех B _Из A[1]_Цикл
    <B, A[2]> _Добавить_к Ogr;
  _Для_всех B _Из A[3]_Цикл
    <A[2], B> _Добавить_к Ogr;
_Кон;

```

```

\X-----\X
\X-----Основной алгоритм-----\X
\X-----\X

```

```

Рез := < >; \X Рез - последовательность \X
\X множеств номеров \X
\X уже растворенных \X
\X веществ, пока пустая \X

```

```

Ост := Исх;   Ж\Ост - текущее   \Ж
              Ж\множество     \Ж
              Ж\еще не раство- \Ж
              Ж\ренных веществ \Ж
_Нач
А:= (Ж _Из Ост : _Не Существует
      у _Из Огр : у[2]=х);
_Для_всех у _Из Огр : у[1] _Из А
      _Цикл у _Удалить_из Огр;
А _Добавить_к Рез
_Кон
_Цикл_До Ост=( ) _Или А=( );

Ж\-----\Ж
Ж\-----Вывод результата-----\Ж
Ж\-----\Ж

Вывод( _Если #Ост=0 Ж\Упорядочили все\Ж
      _То Рез
      _Иначе 'Система Ваших
      уравнений несовместна" )
Ж\-----\Ж

```

Примеры

Пусть на приглашение программы «Введите множество веществ» мы ввели

```
{ 'акрол', 'окситен', 'пелан', 'фетин',
  'цират', 'NaCN' }
```

а на приглашение «Введите множество ограничений» —

```
{ << 'фетин', 'окситен', 'пелан' >>,
  << 'акрол', 'фетин', 'пелан', 'NaCN' >>,
  << ' ', 'цират', 'NaCN' >>,
  << 'окситен', 'NaCN', { } >> }
```

Тогда программа выдаст:

```
< { 'акрол', 'фетин', 'цират',
  'окситен',
  'пелан', 'NaCN' } >
```

А теперь, допустим, мы ввели

```
{ 'фигол', 'витал' }
```

и

```
{ << { }, 'фигол', 'витал' >>,
  << { }, 'витал', 'фигол' >> }
```

Программа на сей раз выдаст

Система Ваших условий несовместна

Обсуждаем дальше

Что может означать ответ нашего алгоритма в последнем примере?

Тому может быть масса причин. В журнале могла быть опечатка, авторы могли просто пошутить, и такого проявителя вовсе не существует. Могла быть ошибка и в справочнике. Могла быть ошибка и в нашем алгоритме, которую мы не заметили из-за ошибки в доказательстве... Но может ли оказаться в этом случае все же так, что ошибок ни в журнале, ни в справочнике, ни в алгоритме нет, а проявитель есть, хотя мы и получили такой ответ?

Если такое допустить, то причина здесь может быть, вероятно, только одна: програм-

ма-то наша безошибочна, да, видимо, для не совсем той задачи. Ведь мы, разрабатывая алгоритм, исходили из некоторой содержательной, а потом и математической модели ситуации. Стало быть, наша модель не соответствует действительному способу изготовления проявителя!

Так что настало время вспомнить о гипотезах, которые мы принимали на веру, создавая математическую модель задачи. А их было немало (причем некоторые были приняты сознательно, а некоторые — как «очевидные»). Например, мы фактически предположили, что процессы растворения участвующих веществ независимы. Быть может, это не так и некоторые из веществ реагируют друг с другом? Так что засыпали А, засыпали В, а в результате в растворе нет ни А, ни В, а появился результат их реакции С... Ведь то, что пишут авторы справочника, в принципе не противоречит такому обороту дел... Или, быть может, в ходе процесса в раствор временно вводят (и потом удаляют) какое-то вспомогательное вещество (что-то вроде катализатора)?

Чтобы уточнить математическую модель с учетом таких возможностей, есть одно средство — еще глубже ознакомиться с литературой по данным вопросам. Ну и попробовать, что выйдет, если исходить из других моделей. Действительно, несколько наивно было полагать, что на базе почерпнутых успехов знаний из первого попавшегося справочника можно создать качественную математическую модель для нетривиальной задачи! Может статься, что для разработки модели, предполагающей весьма непростую ситуацию, не хватит и всей информации, имеющейся на сегодняшний день у специалистов в данной области. Тогда потребуется проведение специальных, быть может, длительных научных исследований для разгадки технологии изготовления. Ведь хорошую математическую модель могут не знать и сами авторы проявителя: вполне возможно, что свое зелье они открыли интуитивно, экспериментально или вообще случайно.

В том, что мы не учли в нашей математической модели такие возможности, мы особо не виноваты: ведь у нас попросту не было достаточной информации... Но в одной из концепций построенной модели отсутствие информации в справочнике ни при чем, хотя ее ошибочность могла тоже стать причиной невозможности изготовления проявителя в рамках нашей модели. Если еще раз посмотреть на все наши предположения, оценить, все ли они вызваны какими-то внешними для нас требованиями и не могут быть заменены другими, то мы обнаружим, что не все. Ведь в нашей модели (а далее и в алго-

ритме) существенно используется предположение, что приготовление проявителя производится путем последовательного растворения в сосуде с водой всех его компонентов. Так что наша задача и свелась к определению неизвестной последовательности растворения.

Но откуда мы взяли, что проявитель требуется делать именно так? Да ниоткуда! Просто нам это почему-то показалось очевидным. Но ведь вполне возможно делать проявитель, растворяя вещества в нескольких сосудах, сливая их на каких-то этапах вместе! Это ничему не противоречит. Хотя математически это будет уже другая, более сложная задача. И вполне возможно, что «не разрешимая» старым способом задача окажется имеющей решение. По крайней мере, в принципе. Кстати, в этом легко убедиться: в приведенном выше «неразрешимом» примере имеющимся ограничениям не будет противоречить такой способ: в одной банке растворить фигол, в другой — витал, а потом слить их вместе. Ведь «не любящие» друг друга вещества «не любят» друг друга в формулировке справочника лишь в процессе растворения, а когда они уже растворены, они могут сосуществовать вполне мирно (по крайней мере, информации о том, что это не так, у нас нет).

Конечно, для реализации способа «в нескольких сосудах» нам нужна дополнительная информация; ведь не ясно, скажем, в каких пропорциях делить общий объем воды на несколько банок. Но все равно, по крайней мере, ответ о принципиальной невозможности выполнения всех условий в данной модели уже будет говорить о необходимости поиска еще более сложных способов. Да и положительный ответ сделает дальнейшие поиски более целенаправленными, позволит аргументировать более интенсивную проработку возможности такого решения.

Какой же может быть математическая модель для «многобаночной» содержательной модели?

Новая математическая модель

Удобно использовать математическую модель, базирующуюся на языке теории графов. Итак, будем полагать, что процесс приготовления проявителя описывается древовидным

графом, ориентированные дуги которого соответствуют эволюции каждой из частей жидкости. Вершинам графа, вероятно, целесообразно сопоставить процессы, при которых изменяется состояние раствора, стало быть процессы растворения веществ либо слияния нескольких растворов. Таким образом, вершины могут быть двух групп: вершины N типов группы A , каждая из которых имеет ровно одну выходящую дугу и не более одной входящей (это вершины, соответствующие растворению одного из N реактивов), и вершины группы B , в которые входят несколько дуг, — без ограничения общности можно считать, две (вершины, соответствующие слиянию нескольких растворов вместе).

Таким образом, требуется найти древовидный ориентированный граф, включающий вершины всех N типов группы A (быть может, даже не по одному разу) и, быть может, какое-то количество вершин группы B . Граф должен быть связным, его корневая вершина, в которую сходятся все пути, соответствует получению желанного продукта. Для требуемого графа должны выполняться «условия допустимости»:

а) необходимо, чтобы всякой вершине типа x_i предшествовали в этом ориентированном графе вершины всех типов из множества R_i (см. выше (1));

б) необходимо, чтобы никакой вершине типа x_i не предшествовала ни одна вершина из множества S_i (см. выше (2)).

Попробуйте теперь разработать алгоритм, определяющий требуемый граф (по той же информации, что и алгоритм, приведенный нами для более простой модели). Не забудьте о необходимости доказательства его правильности: только тогда работу можно будет считать выполненной!

И совсем уж в заключение заметим, что для поставленной нами только что задачи составлять алгоритм может не потребоваться, если в вашем распоряжении есть транслятор с языка логического программирования (например, «Микропролог» ИПС АН СССР для «Ямахи» или «Пролог-Д» из Йошкар-Олы для «Ямахи» и БК-0010). На Прологе данная задача записывается весьма коротко и естественно. Попробуете?

В. ПРОХОРОВ

Международный компьютерный лагерь «Байтик»

В г. Троицке Московской области уже пять лет существует Центр информатики и профориентации школьников «Байтик» при филиале Института атомной энергии им. И. В. Курчатова (ФИАЭ). В Центре проходят обучение все желающие в возрасте от 6 до 17 лет. Дети выбирают занятие по интересам — это может быть программирование, издательский кружок, школа юных менеджеров, кружок технического рисования, занятия в физических лабораториях, экологическая секция и множество других кружков, в которых непременно присутствуют компьютеры. «Байтик» сегодня — это классы с самым современным оборудованием — персональными компьютерами типа IBM PC, Duet-16, «Ямаха», видеотелефонами, лазерами, новейшими физическими приборами. «Байтик» сегодня — это более 1500 детей и 40 преподавателей.



Свыше 1000 учеников занимаются здесь после школы в различных кружках и около 500 приходят на обычные школьные занятия. Функцию преподавателей взяли на себя сотрудники ФИАЭ — ученые, программисты, конструкторы, электронщики. Главное для них — идеи, поиск новых, прогрессивных форм обучения. Ежегодно в «Байтике» проводятся олимпиады и конкурсы по различным дисциплинам с использованием компьютеров.

«Байтик» и увлечение детей компьютерами стали базой для контактов с американскими школьниками. Инициаторами этих связей с американской стороны были преподаватели — члены организации CUE (Computer Using Educators), которые привлекли к участию в программе обмена детей из школ района залива Сан-Франциско. CUE — это

общественная организация, которая объединяет учителей, использующих компьютеры в образовании. С одной стороны, тяга к взаимному познанию культуры, с другой — общий интерес к компьютерам сделали эти контакты достаточно прочными. Обмен происходит в рамках программы «Дети — творцы XXI века».

В 1990 г. лагерь проводился в третий раз. По традиции программа включала три основных направления. Во-первых, это культурные мероприятия — поездки в Ленинград, Суздаль, Загорск, знакомство с достопримечательностями Москвы, поход на байдарках, множество праздников и концертов. Кроме того, наш лагерь отличается тем, что американские дети проживают в семьях своих сверстников. С этим связана вторая часть программы — «бытовая». Дети познают жизнь рядовой советской семьи, вместе готовят обеды, ходят в магазин, ездят на городском транспорте. Для этого в программе предусмотрены специальные семейные дни. Но главное в работе — образовательная программа, организованная силами сотрудников ФИАЭ, которые одновременно являются преподавателями в Центре информатики «Байтик».

Каждый год образовательная программа совершенствуется, но в ней остаются базовые направления, которые всегда пользуются успехом и у советских, и у американских школьников. Это группы экологии, русского языка, пресс-центр, физические лаборатории, группы компьютерной графики, прогнозирования.

1. **Русский язык.** Советские школьники приступили к этим занятиям заранее. Во время летней практики они составили на компьютере англо-русские мини-разговорники, которыми можно пользоваться в повседневной жизни. Были подготовлены программы-тесты, позволяющие ребятам проверить свои знания русского языка. Кроме того, на занятиях использовались программы обучения русскому алфавиту, написанные для детского сада. Начиная с изучения букв. Оказалось, что детсадовская программа вполне подходит для обучения американских школьников! А роль советских ребят заключалась в обучении правильному произношению букв. Следующий этап назывался «Давайте поговорим!» Ребята произносили те

слова и фразы, которые уже знали по-русски. Дети рассказывались в кружок — американец рядом с русским. У каждого в руках наш разговорник, произносили фразу по-русски, фразу по-английски. Эффект был замечательный. На последующих занятиях проверялись знания детей. Когда становилось понятно, что все фразы уже прочно осели в памяти, ученики и преподаватели обсуждали вопросы, связанные с обучением русскому языку.

2. Пресс-группа. В этой секции ребята должны были выпускать газету, используя текстовые и графические редакторы. Интернациональная группа из десяти школьников готовила материал для восьмистраничного еженедельника. Создание газетной страницы включает следующие этапы: написание текстов с помощью текстовых редакторов, выполнение рисунков и чертежей с использованием графических пакетов, подготовка рисунков или фотографий с последующим их кодированием в файл соответствующего формата.

Всю подготовленную информацию с помощью специальной программы, предназначенной для редакционно-издательских работ, школьники располагали на листе бумаги. Необходимо сказать, что помимо знания технической стороны дела, эта работа требовала и художественного вкуса, литературных и изобразительных способностей, что для компьютерного класса, может быть, и не типично.

3. Физический эксперимент. Основное внимание уделялось использованию компьютеров в обучении физике и в научных исследованиях. Работы проходили в лабораториях вакуумной техники, высокотемпературной сверхпроводимости, лазерной техники.

Школьники использовали демонстрационные компьютерные программы, иллюстрирующие физические явления, обучающие программы для работы с измерительными приборами, а также программы для графической обработки экспериментальных резуль-

татов. Рассматривались вопросы сопряжения внешних устройств с ЭВМ и управления игрушкой-роботом.

В этой группе было много дискуссий по поводу социальных последствий внедрения компьютеров, было решено продолжить обсуждение этого вопроса посредством электронной почты.

4. Экологическая программа. Она включала семинар «Экологические аспекты энергетики будущего», занятия по экологии водной среды и качеству воды в реках Подмоскovie, проведение необходимого анализа в лабораториях и в полевых условиях, проведение работ по оценке качества воды на Клязьминском водохранилище с борта яхты. Необходимо отметить, что обработку полученных результатов ребята проводили в компьютерных классах «Байтика».

С помощью специалистов лаборатории была поставлена задача по созданию компьютерной карты Москвы-реки для выполнения учениками работы по дизайну зеленых массивов в ее окрестностях. На занятиях ученики познакомились с одним из графических пакетов, на базе которого были сделаны проекты зеленых посадок.

5. Группа технического рисования. Тематика работ группы ежегодно изменяется. В этом году целью занятий было создание компьютерных комиксов. Быстро освоив один из графических редакторов, ребята с живым интересом принялись за дело. Результатом работы стал сборник комиксов, который получили в подарок и американские, и советские дети.

6. «Прогнозирование». Целью занятий была попытка дать школьникам представление о методах научного прогнозирования, поиска закономерностей в природных явлениях и процессах, о применении компьютеров в статистической обработке данных. Программа курса включала знакомство с начальными понятиями теории вероятностей, математическими методами статистики и самостоятельную научную работу. При знакомстве с современными методами прогнозирования школьники экспериментально проверили закон больших чисел на примере игры в кости. Сначала кости бросали вручную, а когда устали, предложили это делать компьютеру и быстро получили достаточное количество экспериментальных данных. Затем рассматривались задачи о «блуждании пьяницы», прохождении нейтрона через стенку, диффузии газа и азартные игры без стратегии. Компьютер позволил провести вычислительный эксперимент по прохождению пучка нейтронов через слой вещества, найти коэффициенты пропускания и отражения в зависимости от параметров пучка и среды.



На примере исследований зависимости между солнечной активностью и частотой рождения талантливых людей школьникам был продемонстрирован метод поиска закономерностей. При этом ученики освоили работу с базами данных и программами построения графиков. Завершающим этапом была самостоятельная работа, во время которой исследовалась связь между солнечной активностью и землетрясениями, торнадо, ураганами, влияние солнечной активности на научный прогресс (число изобретений). Занятия вызвали огромный интерес у советских и американских детей, причем понравилась не только тема занятий, но и особенно возможность проявить себя в самостоятельных исследованиях.

В разные годы в смысловой части программы лагеря работали секции астрономии, Superquest, САПР, рукоделия, группы по изучению языков программирования, краеведения и др.

В рамках направления «Астрономия» дети смогли посетить планетарий, Институт космических исследований АН СССР, послушать лекции и, конечно, поработать на компьютере над созданием модели полета космических кораблей.

Superquest. Идея создания группы в рамках лагеря пришла из США, где существует конкурс для школьников под таким названием. Каждая школа, которая участвует в конкурсе, разрабатывает какую-нибудь научную задачу, например проблему таяния льда в Антарктиде или увеличения озоновой дыры. При этом в начале работы основное внимание уделяется пониманию школьниками сути проблемы. Ученики знакомятся со специальной научной литературой. Для постановки задач привлекаются видные ученые, которые являются научными руководителями у школьников. В конце года специальное жюри оценивает, насколько глубоко изучены проблемы различными школами, и называет победителя конкурса. Естественно, что в такой работе компьютер используется уже не как средство обучения, а как инструмент для работы. Все расчеты, модели, графики выполняются только с помощью вычислительных машин. Понятно, что участниками этой группы были дети, которые уже имели немалый опыт работы на компьютерах. Одной из самых интересных была задача по изучению проблемы глобального потепления климата на Земле.

Интересно проходили занятия и по обучению основам САПР. Советские и американские школьники совместно проектировали международный лагерь дружбы. Вместе с опытными архитекторами и конструкторами они выбирали площадку для лагеря, делали съемку местности и переносили данные на

компьютер. В лагере, который они спроектировали, есть и остров Фенимора Купера, и площадки для костров, и загадочный водный лабиринт и много другого, интересного, что так любят дети.



Все время, пока американская делегация гостила в г. Троицке, непрерывно работала электронная почта. Американские дети и преподаватели постоянно имели возможность отсылать письма домой и получать ответные послания от родителей и родственников. Кроме того, отправлялось много материалов для американской прессы, которые публиковались в местных и школьных газетах. Электронная почта пользовалась очень большой популярностью. Этот вид связи во многом помогал и нам. С его помощью решались оперативные вопросы, обсуждалась программа визита, в подготовительный период осуществлялся поиск наиболее оптимальных и интересных занятий как для советских, так и для американских детей. Несомненное достоинство электронной почты заключается в том, что после отъезда делегации связь между ребятами не прекращается. Они пишут друг другу письма, вместе участвуют в телекоммуникационных проектах, например, в работе таких телеконференций, как School Link, которая включала в себя обмен новостями, участие в работе по научной и социальной тематике, и World Class, которая была посвящена проблемам экологии.

Четыре недели гостили школьники из США в г. Троицке, и столько же времени советские ребята провели в Калифорнии. За это время они успели посетить ряд школ и посмотреть, как в них организованы компьютерные занятия, поиграть в компьютерные игры с американскими партнерами (эта форма работы будет продолжена через коммуникации). Наши ребята узнали много интересного о применении компьютеров в обучении американских сверстников. Занятия в образовательном Центре Ливерморской национальной лаборатории были посвящены проблеме

глобального потепления климата. Центр очень похож на «Байтик», в нем много компьютерных лабораторий, тематика которых во многом совпадает с тематикой кружков «Байтика». Посещение экологического лагеря дало детям представление о том, как относятся в США к проблемам охраны окружающей среды. Полезны были и контакты преподавателей, которым приобретенный опыт поможет найти новые подходы к организации обучения. Кроме этого, будет проведен

ряд работ совместно с американскими учителями.

Опыт проведения лагерей, подобных «Байтику», показывает, что школьное обучение можно продолжать и в летнее время, совмещая занятия с активным отдыхом. На таких занятиях дети не только закрепляют знания, полученные в школе, но и узнают много нового и интересного. Несомненную пользу от этого получают все: и школьники, и преподаватели.

III Всесоюзная олимпиада по информатике

Олимпиада проходила 18—25 апреля в г. Харькове с участием 18 команд: союзных республик (кроме Литовской ССР), Ленинграда, Харькова, Министерства путей сообщения. Возглавлял жюри олимпиады академик АН УССР В. Л. Рвачев, в состав вошли члены методической комиссии по организации Всесоюзных олимпиад по информатике при ГКНО СССР (председатель академик Н. Н. Красовский), представители команд, преподаватели и сотрудники ХГПИ им. Г. С. Сковороды и ХГУ им. А. М. Горького.

Олимпиада проводилась в два тура — теоретический (2 задачи) и практический (1 задача). Оба тура проводились с использованием ПЭВМ, причем в теоретическом оно было необязательным.

В распоряжение участников олимпиады были предоставлены в соответствии с их пожеланиями ПЭВМ «Ямаха» или IBM PC с программными средствами Бейсик, Турбо-Паскаль, Турбо-Си. Максимальная оценка в каждом туре — 100 баллов. Тесты и критерии оценки определялись жюри.

Задачи к олимпиаде готовила методическая комиссия в избыточном количестве (5 задач на теоретический тур, 3 задачи на практический). Окончательный отбор проводило жюри непосредственно перед туром.

В результате обсуждения были выбраны следующие задачи.

Теоретический тур

Задача 1 (автор П. С. Панков). Круг разрезан несамопересекающейся ломаной, координаты вершин которой заданы парами натуральных чисел $(X_1, Y_1), \dots, (X_k, Y_k)$. Первая и последняя вершины лежат на границе круга, а остальные — внутри него. Определить, можно ли разъединить две получившиеся части круга (выход из плоскости не допускается).

Задача 2 (автор Я. Н. Зайдельман).

Какое минимальное количество одинаковых (P, Q)-коней требуется для контроля:

- ограниченной шахматной доски размером $M \times N$ клеток;
- бесконечной полосы шириной M клеток;
- бесконечной плоскости.

Примечание: 1. (P, Q)-конь — фигура, которая за один ход перемещается на P клеток по горизонтали и Q клеток по вертикали или наоборот. Например 1,2-конь — это обычный шахматный.

2. Фигура или группа фигур контролирует поле, если каждая его клетка доступна за один или несколько ходов хотя бы одной фигуре. Например, для контроля стандартной доски размером 8×8 требуются два (1,1) коня.

Практический тур

Задача 1 (автор В. В. Прохоров).

Обозначим $S(n, a, b) = \sum_{k=1}^n \frac{k}{a^{\lfloor \log_b k \rfloor}}$ где, $\sum_{k=1}^n x_k = x_1 + x_2 + \dots + x_n$; $\lfloor \log_b k \rfloor$ означает целую часть $\log_b k$; через $\log_b k$ обозначено такое число Z , что $b^Z = k$. Требуется составить программу, которая выводит результат с возможно большим числом верных десятичных знаков:

- $S(100, 3, 2)$;
- $S(100, 3, 1.9)$;
- $S(100, 3.1, 1.9)$.

Следует отметить возросший по сравнению с прошлой олимпиадой уровень подготовки школьников. По итогам III Всесоюзной олимпиады дипломы I степени получили 6 школьников (они набрали 98—115 баллов), дипломы II степени — 11 школьников (60—85 баллов), дипломы III степени — 14 школьников (48—58 баллов).

Многие школьники были награждены призами общественных, научных организаций и спонсоров олимпиады — НТТМ «Поиск». В частности, за лучшее решение первой задачи I тура призы получили Ю. Зайцев (г. Киев, СШ № 57, XI класс), Т. Докшицер (Ленинград, СШ № 239, XI класс) и Л. Натрошвили (г. Тбилиси, СШ № 25, XI класс).

За лучшее решение второй задачи I тура приз получил Г. Еськов (Ленинград, СШ № 239, XI класс). За лучшее решение задачи II тура приз получил О. Таборовец (г. Пинск, СШ № 4, XI класс).

Жюри олимпиады приняло решение о дополнительном включении в состав кандидатов в сборную СССР для участия во II Международной олимпиаде по информатике А. Демакова (Кировская обл., Даровская СШ) и Г. Еськова (Ленинград, СШ № 239). В состав кандидатов для подготовки к участию в III Международной олимпиаде по информатике, которая состоится в 1991 г. в Греции, включены призеры олимпиады, не оканчивающие в этом году школу.

III Всесоюзная олимпиада по информатике подвела итоги олимпиадного движения по этому предмету в нашей стране в 1990 г. Говорить об установившихся формах проведения этих олимпиад еще рано — достаточно проблем в ее методическом, организационном и техническом обеспечении. Однако имеющаяся тенденция вселяет определенную уверенность в ее будущем, и хочется надеяться, что эта олимпиада явилась важным шагом на пути формирования профессиональных навыков всех ее участников. Желаем им успехов в этом трудном деле.

Призеры III Всесоюзной олимпиады школьников по информатике

Дипломы I степени получили:

Козлов Д. (Ленинград, СШ № 566, XI класс),
Зайцев Ю. (г. Киев, СШ № 57, XI класс),
Таборовец О. (г. Пинск, СШ № 4, XI класс),
Варблане Р. (п. Нью ЭССР, Ньюская СШ, XII класс),
Демаков А. (Кировская обл., Даровская СШ, XII класс),

Еськов Г. (Ленинград, СШ № 239, XI класс),

Дипломы II степени получили:

Чельшев А. (г. Чирчик УзССР, СШ № 24, IX класс),
Москаленко А. (г. Омск, СШ № 20, XI класс),
Натрошвили Л. (г. Тбилиси, СШ № 25, XI класс),
Кузьменко Т. (г. Киев, СШ № 51, XI класс),
Датуашвили Г. (г. Тбилиси, СШ № 25, XI класс),
Яценко Р. (Москва, ФМШ при МГУ, XI класс),
Курбасов М. (г. Рязань, СШ № 2, XI класс),
Тарасов Ю. (г. Киров, СШ № 35, XI класс),
Докшицер Т. (Ленинград, СШ № 239, XI класс),
Свинарев А. (г. Харьков, СШ № 106, XI класс),
Капильш А. (Латвийская ССР, XI класс).

Дипломы III степени получили:

Гурдюмов А. (г. Благовещенск, СШ № 1, XI класс),
Лепехин В. (г. Волгоград, СШ № 132, X класс),
Ткачук К. (г. Ташкент, СШ № 243, X класс),
Труу А. (п. Нью ЭССР, Ньюская СШ, XII класс),
Зильберфайн В. (г. Новосибирск, СШ № 27, XI класс),
Дуда Е. (г. Харьков, СШ № 27, XI класс),
Зиновьев Н. (Москва, СШ № 843, XI класс),
Николаев М. (г. Брест, СШ № 3, XI класс),
Костюк А. (г. Воронеж, СШ № 1, XI класс),
Герштейн С. (г. Свердловск, интернат № 19, X класс),
Уваров Д. (г. Новокузнецк, СШ № 11, X класс),
Григарошвили С. (г. Тбилиси, ФМШ при ТГУ, XI класс),
Быструев Д. (Москва, СШ № 47, XI класс),
Хаменя В. (г. Гродно, СШ № 1, X класс).

Могут ли программы не содержать ошибок?

В шуточной статье, опубликованной в «ИНФО», излагалась «теория программирования», начинающаяся с «аксиомы»: «Всякая программа содержит ошибки». Исходя из этой теории программирование — дело безнадежное и лучше заняться чем-нибудь другим — кройкой, шитьем или выпиливанием лобзиком.

Но, может быть, под программирование можно подвести некоторые твердые основания и надеяться, что хоть иногда программы хотя бы у некоторых людей могут получаться не содержащими ошибок? Для доказательства этого приведем пример очень простой программы на языке Бейсик, сопроводив его описанием алгоритма.

10 *СУММА ЧИСЕЛ	алг "сумма чисел"
20 INPUT "A="; A	нач
30 INPUT "B="; B	запрос "a=", a
40 S=A+B	запрос "b=", b
50 PRINT "S="; S	c:=a+b
60 END	вывод "c=", c
	кон

Результатом работы данной программы, как легко видеть из ее текста (и описания реализуемого ею алгоритма), является вывод значения числа s , равного сумме a и b . Какие бы два числа a и b мы ни вводили, в ответ машина будет давать значение $s=a+b$. Таким образом, эта программа всегда будет давать правильные результаты, а следовательно, никаких ошибок в ней нет; зато есть надежды на возможность составления алгоритмов и программ, которые не содержат ошибок.

Но что тогда делать с приведенной выше «аксиомой»? По-видимому, она не совсем верна — есть программы, которые содержат ошибки, но существуют и программы, которые ошибок не содержат! Как отличать их друг от друга — вот в чем вопрос.

Сначала разберемся: а что такое ошибки в программах? Традиционно на этот вопрос отвечают, что они подразделяются на синтаксические и алгоритмические. Синтаксические — ошибки в записи программы, а именно нарушения правил используемого языка программирования. Алгоритмические — ошибки в алгоритме, реализованном в данной программе. Но тогда возникает вопрос: где искать алгоритмические ошибки — в алгоритме или в программе?!

Совершенно нетривиальным будет такой ответ: ошибки в программах нужно искать на ЭВМ! Но он вызывает еще два вопроса:

как искать ошибки на ЭВМ и как распознать, есть ошибки в программе или нет? Для ответа на второй вопрос нужно выяснить, каковы проявления алгоритмических ошибок. Имеющие опыт отладки программ на ЭВМ сразу же скажут: отказы (зацикливание программ, их аварийное завершение из-за деления на 0, вычисления корней из отрицательных чисел и т. п.); сбои (потеря части результатов, получение непонятных сообщений, отсутствие каких бы то ни было сообщений и т. д.); получение неверных результатов.

Отсюда понятно: чтобы доказать наличие ошибок в представленной программе, нужно придумать такие проверочные (тестовые) данные, что машина даст сбой, отказ или неверный результат. Но что такое — неверный результат? И что такое верный результат? Корнем этих понятий является слово «вера». Но вера в кого или во что — в ЭВМ, программиста или в некое высшее существо, которое скажет, какие результаты верные, а какие нет?

В отличие от программирования в математике результаты подразделяются не на верные или неверные, а на правильные или неправильные, в зависимости от требований, которые должны быть четко формулируемы. Если требования не сформулированы, то оценки правильности приходится принимать на веру от того лица, которое имеет право судить и давать оценки. Если требования сформулированы нечетко (расплывчато, двусмысленно), то опять же необходимы толкователи, которые должны высказать окончательные суждения о правильности полученных результатов.

Дело можно поставить на твердые основания, только если мы будем иметь четкие определения — что мы хотим иметь в тех или иных ситуациях. Руководствуясь этими определениями (спецификациями), можно будет однозначно судить о правильности (неправильности) полученных результатов (без ссылок на личности и высокие авторитеты).

Таким образом, для обоснованности суждений о наличии ошибок в алгоритмах и программах необходимо иметь спецификации, описывающие, что должно быть результатами их работы. В отсутствие спецификаций об алгоритмах и программах можно говорить все что угодно, в частности, что все они содержат ошибки! Если же такого

рода спецификации у нас будут, то можно рассчитывать на более или менее полную проверку правильности предоставляемых алгоритмов и программ в части соответствия этим спецификациям.

Подводя итоги, можно сформулировать некий вариант определения наличия ошибок в программах. Будем говорить, что программа содержит ошибки, если можно указать такие исходные данные, при которых произойдет сбой, отказ или будут получены неправильные результаты. Соответственно назовем программу не содержащей ошибок, если она дает правильные результаты при любых допустимых исходных данных.

Теперь понятна суть отладки программ на ЭВМ: это процесс поиска в них ошибок путем подбора контрпримеров (тестовых исходных данных, при которых программа даст сбой, отказ или неверные результаты). Однако на этом пути возникает проблема: как гарантировать полноту проверки, а именно, как удостовериться, что проверены все возможные ситуации и в ходе отладки выявлены все ошибки?

Действительно, если для утверждения о наличии ошибок в программе достаточно указать и продемонстрировать на ЭВМ хотя бы один контрпример, то для утверждения об отсутствии ошибок необходимо показать, что правильные результаты будут получаться при любых допустимых исходных данных. Если же вариантов допустимых исходных данных чрезмерно много (хуже того — бесконечное число), то показать отсутствие ошибок только путем отладки становится практически или даже в принципе невозможно!

Таким образом, выявить все ошибки или гарантировать их отсутствие в программах только путем проверки на ЭВМ в общем случае нельзя. Ясно, что полнота проверки может быть обеспечена только путем исчерпывающего анализа. И здесь важно знать, для какого класса алгоритмов и программ может быть проведен такой исчерпывающий анализ и как его провести.

Всякий имеющий опыт отладки программ, опыт анализа и выявления ошибок знает, что это проводится путем разбора различных случаев и ситуаций. Давайте рассмотрим простейший пример.

```

алг"   ???   "
нач
  v:=x*x;
  v:=v*x;
  y:=v*x;
кон

```

пусть:
 x=2
 v1=2*2=4
 v2=4*4=16
 y=16*2=32

Справа выписаны результаты выполнения алгоритма в ситуации, когда величина x равна двум. Первое присваивание $v:=x \cdot x$ дает пер-

вое значение для переменной v , равное $v1=2 \cdot 2=4$. Второе присваивание вычисляет значение $v \cdot v=4 \cdot 4$ и записывает число $4 \cdot 4=16$ опять в переменную v . Результатом этого присваивания будет новое, второе, значение переменной v , которое оказывается равным $v2=16$. Третье присваивание $y:=v \cdot x$ вычисляет значение $v2 \cdot x=16 \cdot 2=32$ и записывает его в переменную y , которая становится равной 32.

Ясно, что получена величина $y=32$ и она есть не что иное, как 2^5 . Аналогичный результат можно показать разбором результатов выполнения алгоритма при $x=3, 4, 5$ и т. д. Но утверждение о том, что конечный результат вычислений есть $y=x^5$, можно доказать и в общем случае, описав результаты выполнения каждой из операций присваивания в этом алгоритме.

```

алг"  y=x^5  "
нач
  v:=x*x;
  v:=v*v;
  y:=v*x;
кон

```

результаты:
 v1=x*x=x^2
 v2=v1*v1=v1^2=x^4
 y=v2*x=x^4*x=x^5

Действительно, результат выполнения первого присваивания $v:=x \cdot x$ в общем случае будет равен $v1=x \cdot x=x^2$, результат выполнения второго присваивания $v:=v \cdot v$ даст $v2=v1 \cdot v1=v1^2=x^4$, результат последнего присваивания $y:=v \cdot x$ окажется равным $y=v2 \cdot x=x^4 \cdot x=x^5$. Следовательно, данный алгоритм при произвольном исходном значении x в качестве конечного результата вычисляет величину $y=x^5$.

Таким образом, при любых допустимых исходных данных этот алгоритм будет давать правильные результаты; мы доказали, что данный алгоритм правильный и никаких ошибок в нем нет. Что и требовалось.

Итак, программирование не такое уж безнадежное дело. Но как велик класс алгоритмов и программ, для которых можно доказать их правильность, как строятся эти алгоритмы и как обосновывается их правильность?

Данной серьезной теме посвящено много книг. Из них стоит познакомиться прежде всего с работами профессора Э. Дийкстры [1, 2], в которых изложены основные идеи и разобрано большое число примеров алгоритмов и программ с доказательствами их правильности. Среди отечественной учебной литературы этой теме посвящены работы [3, 4, 5], в которых изложены математические основы, примеры и методика составления алгоритмов и программ с одновременным обоснованием их правильности.

В целом же техника анализа правильности алгоритмов строится на умении формулировать суждения о результатах вы-

полнения отдельных действий и операций, составляющих алгоритмы и программы, и делать выводы и умозаключения, исходя из этих фактов. Кроме того, необходимо умение проводить несложные, но математические по существу рассуждения и доказательства, четко формулировать суждения и утверждения, четко ставить задачи, а также подбирать и обосновывать методы их решения.

Знаний одного, двух или даже пяти языков программирования и опыта проведения отладок программ на ЭВМ для этого недостаточно. Программирование — большая и серьезная наука, опирающаяся на строгий математический фундамент. И каждому, желающему посвятить себя этому делу и добиться в нем успехов, стоит потратить время на освоение математического аппарата. Как и всякие фундаментальные знания, изучение математики окупается сторицей.

Более серьезным является утверждение о том, что освоение науки, искусства и ремесла программирования невозможно без изучения соответствующих математических знаний. А для реализации этого положения изучение информатики и программирования необходимо проводить в интеграции с изучением соответствующих разделов математики [6].

Литература

1. Дijkstra Э. Заметки о структурном программировании // Структурное программирование. М.: Мир, 1975.

2. Дijkstra Э. Дисциплина программирования. М.: Мир, 1978.

3. Каймин В. А. Методы разработки программ на языках высокого уровня. М.: МИЭМ, 1985.

4. Каймин В. А. Основы доказательного программирования. М.: Изд-во МИЭМ, 1987.

5. Каймин В. А., Щеголев А. Г. и др. Основы информатики и вычислительной техники: Пробный учебник для средних школ. М.: Просвещение, 1989.

6. Каймин В. А., Угринович Н. Д. и др. Методика преподавания информатики по машинному варианту. М.: Изд. МГИУУ, 1990.

От редакции. Итак, может ли программа не содержать ошибок? В. А. Каймин (в отличие от В. В. Тихонова) считает, что да. Редакции очень хотелось бы согласиться с ним, но... Достаточно в первой программе из статьи задать $a=0.4444444444444444$, $b=0.5555555555555554$, и результат, вопреки утверждению автора, будет неверным (можете проверить сами на «Ямахе», а заодно и сравнить с результатом при $b=0.5555555555555555$). Сомнительным будет и результат работы второй программы при, например, $x=$ «два»...

97

Как верно замечено в статье, эти ошибки коренятся не в программе, а в спецификациях. Только слова автора «какие бы два числа a и b мы ни вводили...» позволили использовать довольно причудливые значения исходных данных. Следовательно, вынесенный в заголовок вопрос о правильности программ как таковых не вполне корректен: ответ на него зависит от точности спецификаций.

«Правильная программа» — выражение традиционное, но неверное. Помните об этом!

Малая научно-техническая кооперативная фирма «ИНДЕКС» предлагает

классы обучающих компьютеров «Атари» в составе:



98

8—10—12 компьютеров «Atari 65XE» с цветными мониторами «Электроника»; локальная сеть «Индекс ЛС-1»; один общий дисковод, один кассетный накопитель; джойстик к каждому компьютеру; сетевая версия дисковой операционной системы;

система АТАРИ-УЧИТЕЛЬ — инструментальное средство создания учебных программ;

обучающие программы по языку Атари-Бейсик; обучающая программа по базам данных; обучающая ЛОГО-СИСТЕМА; языки программирования: Атари-Бейсик, ассемблер, Паскаль; текстовый редактор; музыкальный редактор; графический редактор; документация по программным и аппаратным средствам.

Возможна поставка только сетевого оборудования «Индекс ЛС-1» для пользователей компьютеров «Атари».

В расширенной конфигурации в состав класса включаются принтер и (или) персональные ЭВМ типа IBM PC XT/AT и «Atari ST»

Высокая надежность и живучесть класса обеспечиваются благодаря однородности структуры класса и отсутствию головного компьютера. Отказ любого компьютера приводит не к отказу всего класса, а лишь к уменьшению числа рабочих мест. Фирма «Индекс» обеспечивает гарантийное и послегарантийное обслуживание классов.

Заявки направлять по адресу: 103045, Москва, а/я 41, «Индекс».
Тел. 386-13-09. Факс 123-63-55.

А. МАТЮШКИН-ГЕРКЕ

Школьная информатика: уроки, проблемы и перспективы

99

Использование в любой сфере человеческой деятельности тех или иных принципиально новых средств, резко усиливающих физические и/или интеллектуальные возможности исполнителей, оказывается действительно эффективным лишь при том обязательном условии, что сама эта деятельность претерпевает кардинальнейшие изменения. Как состав и содержание задач, так и методы, применяемые для их решения, должны коренным образом обновиться. Разумеется, далеко не всякие, пусть даже весьма радикальные, перемены могут обеспечить успех. Однако уж заведомо обречены на провал попытки оставить «всё, как есть», использовать эти мощные средства для решения старых задач.

Мы с чувством интеллектуального превосходства рассматриваем карикатуру, где антропоморфные работы, подобно репинским бурлакам, тянут баржу. Мы смеемся над подслеповатой крыловской мартышкой, так и не сумевшей воспользоваться очками. Но ведь мы сами сплошь и рядом уподобляемся ей! И никак не можем (или не хотим?!) это понять! Хотя сам по себе этот фундаментальный методологический принцип — привлечение в любую область деятельности качественно новых средств должно сопровождаться соответствующими изменениями ее содержания и методов — одинаково справедлив и в простых, и в сложных ситуациях, осознается он далеко не всегда.

Пожалуй, особенно не повезло в этом отношении электронно-вычислительной технике. Именно поэтому она сплошь и рядом не только не дает ожидаемого эффекта, но и приводит к значительным материальным потерям, к бессмысленной растрае интеллектуальных ресурсов. Для под-

тверждения этого тезиса достаточно напомнить хотя бы про эпопею с созданием АСУ, обошедшую стране в сотни миллиардов рублей и миллионы впустую растроченных человеко-лет труда высококвалифицированных специалистов.

Ничуть не лучше положение дел в сфере образования. Разве что потери здесь выражаются меньшими числами. Но это только потому, что гораздо меньшие средства сюда и вкладывались. Радужные прогнозы прошлых лет даже в малейшей степени не оправдались, и теперь самые ретивые сторонники компьютеризации учебного процесса вынуждены признавать, что в лучшем случае (!) так называемые обучающие программы работают на уровне среднего педагога. На самом деле и такое является большой редкостью. И не только в нашей стране!

Почему же не дают отдачи многочисленные научные исследования по этой проблеме, разнообразные «методики написания обучающих программ», создание «комплексных бригад», объединяющих в своем составе программистов и учителей-предметников, разработка всевозможных «инструментариев», облегчающих создание педагогических программно-информационных средств?

В качестве главных причин обычно называют слабую программистскую подготовку учителей, трудности достижения взаимопонимания между ними и специалистами по вычислительной технике. На самом деле это не так! Великолепно владеющий и своим предметом, и компьютером педагог с группой отличных помощников-программистов заведомо обречен на неуспех, коль скоро он будет в новых условиях строить обучение

в рамках тех же программ, по которым велось преподавание в безмашинном варианте. *Для школьных курсов в их ныне существующем виде, будь то математика или язык, география или физика, вычислительная техника практически не нужна!* Эти курсы сложились в докомпьютерные времена, и их становление происходило с ориентацией на гораздо более примитивные средства обучения, с обстоятельной притиркой к этим средствам и содержания предмета, и методики обучения. Они хорошо идут всякой вычислительной да и иной сложной техники, потому что именно в расчете на это «без» и создавались, и именно в этих условиях просуществовали много лет. Попытки искусственного «вживления» компьютеров в традиционную схему этих курсов аналогичны, по образному выражению академика В. М. Глушкова, установке ракетного двигателя на телегу.

Ведь нелепо, согласитесь, имея под рукой компьютер, ограничиваться в курсе математики решением лишь линейных и квадратных уравнений, в курсе физики — рассмотрением только равномерного и равноускоренного движения, в химии — полностью игнорировать кинетику реакций... Да и в преподавании предметов гуманитарного цикла неправомерно ограничиваться примитивными обучающими и контролирующими программами. Другое дело, что при определении нового содержания учебных курсов во главе угла должны стоять грамотно поставленные цели и задачи обучения. И далеко не все то, что оказывается возможным (и порой заманчивым) сделать с помощью компьютера, будет действительно полезным и целесообразным. Не исключено даже, что в каких-то случаях компьютер станет попросту не нужен! Примеров внешне эффектных «учебных» программ, которые на самом деле способны лишь дискредитировать компьютеризацию, сегодняшняя практика дает, к сожалению, очень много.

Представляется, что для многих разработчиков компьютерных методик обучения именно само использование ЭВМ в учебном процессе является главной целью. Это не трудно заметить хотя бы по характеру подавляющего большинства докладов и публикаций на эту тему. Ведь среди них практически нет таких, где во главу угла ставились бы новые педагогические цели и задачи именно в конкретной предметной области, а компьютер появлялся бы уже потом, в качестве одного из средств достижения этих целей. В наших (да и не только в наших) научно-методических журналах нет ни одной статьи, где вопрос ставился бы примерно так: «Традиционный школьный курс физики

не дает возможности изучать неравноускоренное движение, большинство явлений гидро- и аэродинамики, переходные процессы в электрических цепях и т. д., что недопустимо обедняет этот учебный предмет. Какими путями можно было бы это положение поправить? Не окажется ли для этого полезным компьютер?..»

Не приходится сталкиваться с подобной постановкой вопроса и на многочисленных сейчас конференциях, семинарах, совещаниях. Зато уж автоматизированных решателей типовых задач, программированных опросников, «компьютерных мультиков», привязанных к традиционным схемам учебных курсов, более чем достаточно. Так сказать, «возьмем ЭВМ и сделаем с ее помощью то, что раньше делали без нее».

И при этом подобная «компьютеризация» еще и всячески восхваляется. Без компьютера, дескать, было плохо, а с ним стало хорошо! Ну а если к шикарному автомобилю приделать оглобли, запрячь туда лошадь и использовать его в качестве телеги?! Тоже ведь будет «лучше»: и дождь не мочит, и радиоприемник под рукой... Только вот то ли это улучшение, ради которого стоит автомобиль приобретать?

К счастью (!), пока для математики, физики, литературы компьютеров у нас еще крайне мало, поэтому в ближайшее время особо крупного вреда от их безграмотного применения в обучении традиционным школьным предметам не будет. Гораздо более остро стоит проблема преподавания информатики. Курс этот относительно новый. В учебные планы массовой школы он вошел с 1985/86 учебного года в порядке реализации мартовского (1985) постановления ЦК КПСС и Совета Министров СССР.

Не столько даже само решение о столь поспешном введении этого курса, сколько официальную трактовку его содержания бывшим Минпросом СССР и выбранные им способы и средства реализации этого решения следует оценить весьма критически.

Действительно, почти полное отсутствие подготовленных учительских кадров, катастрофическая нехватка вычислительной техники, необеспеченность курса не только приемлемым учебником, но даже и сколь-нибудь удовлетворительной программой — все это требовало существенно большей постепенности в сроках введения курса и в то же время гораздо более четкой и квалифицированной организации подготовительной работы. К сожалению, в этой подготовительной работе и Минпрос СССР и институты АПН СССР показали свою полную несостоятельность. Во многом это было обус-

ловлено тем, что решение всех вопросов, связанных с введением курса основ информатики, оказалось монополизированным узкой группой лиц, не обладавших к тому же должным уровнем компетентности.

Первоначальная подготовка учителей (обычно в объеме 72 ч, очень редко — 144 ч), как правило, проводилась силами различных вузов. Уже в это время обозначились весьма различные точки зрения на содержание курса. Одни трактовали его сущность как «обучение алгоритмизации» (?) и заставляли учителей (а те потом школьников) разрабатывать алгоритмы сбора моркови, перехода улицы, посещения столовой и т. п. Другие рассматривали его как начальный курс программирования на том или ином языке. Третьи, четвертые и пятые пытались определить его содержание, исходя из того или иного истолкования не устоявшегося еще термина «информатика». К сожалению, наиболее важные и нужные для формирования научного и технического мышления учащихся аспекты чаще всего попросту игнорировались или преподносились в чрезмерно упрощенном виде, чисто декларативно.

Вряд ли в тех условиях можно было бы ожидать иных результатов. Ведь если бы в похожем положении оказалась, например, математика, то в одних случаях ее школьный курс ограничивался бы арифметикой и задачами «про стаю гусей», а в других — включал бы топологию, абстрактную алгебру, функциональный анализ и тому подобные разделы!

Появившиеся на свет с недопустимым опозданием минпросовская программа школьного курса (конец июня 1985 г.) и пробное учебное пособие под ред. А. П. Ершова и В. М. Монахова (август — сентябрь 1985 г.) не поправили положения. Скорее, они даже ухудшили его. Ведь для многих неопытных учителей и методистов эти программа и пособие стали, хотя бы на первое время, своего рода ориентиром, на роль которого на самом деле они никак не годились из-за содержащихся там грубых ошибок и просчетов. А всякая их публичная критика (равно как и альтернативные предложения по построению курса) в то время жестко блокировалась под тем предлогом, что она-де будет деморализовывать учителей.

За прошедшие пять лет существенно менялся корпус учителей информатики. Многие учителя, прежде всего математики и физики, взявшиеся поначалу (в добровольном или добровольно-принудительном порядке) за ведение курса информатики в дополнение к своему основному предмету, столкнувшись с тяжелыми условиями и малой от-

дачей, постарались вернуться целиком к преподаванию привычных для них дисциплин. Преобладавшие в так называемой второй волне инженеры и программисты принесли с собой в школу свой профессиональный опыт и знания, к сожалению не всегда сочетавшиеся с пониманием задач, стоящих перед этим общеобразовательным курсом. Далеко не всем и из этой «второй волны» удалось адаптироваться и закрепиться на новом для них поприще, тем более что большие дополнительные трудности были связаны и с отсутствием надлежащего педагогического опыта. Наверное, только сейчас можно говорить о начавшейся стабилизации учительских кадров в этой области.

Почти полное отсутствие в школах каких бы то ни было средств вычислительной техники, особенно в 1985—1986 гг., потребовало поиска внешних возможностей для организации практических занятий учащихся по этому курсу. С 1986 г. было положено начало созданию межшкольных центров, на которых проводились выездные практические занятия учащихся закрепленных за этими центрами школ. Одновременно с этим проводились и проводятся в настоящее время занятия на базе некоторых вузов и НИИ.

Организуются такие занятия по-разному. В одних случаях со школьниками приезжает их учитель, который полностью проводит практическую часть курса, в других — преподавание информатики во всех школах «куста» осуществляется небольшим коллективом учителей, входящих в штат межшкольного центра. В последнее время именно такой подход начинает преобладать.

К сожалению, имеются (хотя и в небольшом числе) примеры и иного рода. Например, бывает так, что «теоретические» занятия в классе проводит школьный учитель, а практические занятия на ЭВМ (на которые этот учитель порой даже не допускается) ведутся инженерами вычислительного центра вне всякой связи с вышеупомянутой «теорией». Хорошо еще, если эти учителя и инженеры воздерживаются от сообщений ученикам нелестных отзывов друг о друге!

Весьма различны содержание и методика занятий. К сожалению, в отсутствие хоть сколько-нибудь приемлемого учебника срывается своего рода импринтинг. Учителя, начавшие в свое время преподавание по пробному учебному пособию А. Ершова и В. Монахова, продолжают преподавание в том же ключе. Перешедшие же на работу в школу специалисты по вычислительной технике основной упор делают на освоении учащимися программирования на том или ином алгоритмическом языке или же (в зависимости от собственного опыта) на технике работы с так

называемыми программными средствами высокого уровня: текстовыми и графическими редакторами, электронными таблицами, системами управления базами данных и т. п.

Модным в последнее время становится использование различного рода обучающих и игровых программ, хотя во многих случаях эта мода явно не согласуется с педагогической целесообразностью. Необходимо отметить, что рынок этих программно-информационных средств в настоящее время практически неуправляем, а качество большинства фигурирующих там продуктов весьма низкое. Все больше и больше на нем проявляют себя коррупция и мошенничество.

Очень тяжелым остается положение с вычислительной техникой. Ее, разумеется, стало гораздо больше, чем в год введения в учебные планы курса информатики. Но «гораздо больше» ноля — это не всегда «много». И сейчас далеко еще не каждая школа располагает своим компьютерным классом. А ведь один такой класс на школу в 1000—1500 учащихся — это всего 3—5 мин работы за дисплеем в день на одного ученика. Этого едва-едва хватит на небольшой курс информатики и, быть может, на несколько «показательных уроков» по другим дисциплинам.

Количество различных типов вычислительной техники, используемой в учебном процессе в средней школе, составляет несколько десятков. А поскольку особенности технического обеспечения тоже существенно влияют на характер преподавания, то число различных вариантов постановки курса информатики оказывается попросту необозримым. Обмен опытом для владельцев разных типов учебных классов оказывается возможным разве лишь на концептуальном уровне, да и то далеко не во всем. Крайне осложняются централизованная разработка и тиражирование педагогических программно-информационных средств (ППИС), с громадным трудом решаются вопросы технического обслуживания компьютерных классов.

Техническая политика как бывших, так и нынешних руководителей народного образования была и остается неудовлетворительной. Даже тогда, когда главной и чуть ли не единственной задачей курса основ информатики официально провозглашалось обучение алгоритмизации, что само по себе никоим образом не согласуется с теми задачами, которые должен решать курс, тот же Минпрос СССР представил в качестве основной модели школьного компьютера ПЭВМ «Агат». Учить просто алгоритмизации с успехом можно было на гораздо более простой и дешевой технике. Затем появились «гибридные» классы КУВТ-86 на

базе ДВК и БК-0010, их сменили «Корветы» и УКНЦ. Теперь же под предлогом необходимости «использования новых информационных технологий фирмы IBM» курс берется на профессиональные ПЭВМ семейства PS/2 (?!).

Не говоря уже о недопустимости буквально ежегодной «смены ориентиров» (а переход с одного типа ЭВМ на другой — достаточно трудное дело), такой подход порочен и по ряду иных причин.

Прежде всего, большинство школ оказываются просто-напросто не в состоянии обеспечить себя такой вычислительной техникой. Производится ее слишком мало, а стоимость очень высока. Десятки тысяч рублей стоят самые дешевые классы, а одна (!) ПЭВМ семейства PS/2 — до 100 тыс. и более. С другой стороны, при нормальном, а не надуманном содержании занятий большинство возможностей этой дорогостоящей техники оказывается попросту не востребованым и коэффициент ее полезного действия будет не выше одного-двух процентов. Даже сегодня с трудом (и не всегда успешно) решаются вопросы технического обслуживания школьных компьютерных классов. Что же будет тогда, когда их число возрастет хотя бы в несколько раз? Как горько пошутил один наш коллега, если у Гособразования СССР и есть здесь какая-то стратегия, то она очень искусно им скрывается. А ведь кроме ЭВМ, распределяемых союзным руководством народного образования, в школу попадает различными путями самая разнообразная техника вплоть буквально до самоделок. Эту-то технику кто будет поддерживать в рабочем состоянии? Кто, наконец, хотя бы объяснит неопытному учителю, что мало приобрести компьютерный класс, надо еще суметь обеспечить его нормальное чисто техническое функционирование, надо заполучить к нему программно-информационные средства?

Плачевное состояние, в котором находится сейчас наша школьная информатика, — закономерный результат волюнтаристских методов руководства. С самого начала научно-педагогическая общественность оказалась не только фактически отстраненной от решения основных проблем компьютеризации среднего образования, но даже лишенной необходимой информации как о существующем положении дел, так и о текущих и перспективных планах и мероприятиях по осуществлению этих планов.

Начинавшийся весьма многообещающе всесоюзный семинар АН СССР и АПН СССР «Компьютер и образование» постепенно превратился в заурядное мероприятие. Вряд ли его судьба могла быть иной: ведь все

прозвучавшие там идеи, конкретные рекомендации и предложения обернулись в итоге лишь сотрясением воздуха. Они оказались ненужными для «власть предержащих».

В школьной секции Всесоюзного научно-методического совета по информатизации образования немало активных, болеющих за дело, глубоко понимающих проблемы компьютеризации средней школы ученых и педагогов. Но ведь ни бывший союзный Минпрос, ни теперешний Госкомитет СССР по народному образованию не проявляли и не проявляют ни малейшего желания считаться с этим коллективом лучших, казалось бы, в стране специалистов.

Действительно, в обход этой секции было принято решение о выпуске в качестве школьного учебника по информатике сначала книги А. П. Ершова, А. Г. Кушниренко и др. (1988), а затем А. Г. Кушниренко и др. (1990). И это при том, что уже были определены победители завершившегося в апреле 1987 г. конкурса на лучший учебник, а рукописи вышеупомянутых книг в этом состязании не участвовали. Благополучно «замороженным» оказалось принятое еще в 1987 г. принципиально важное решение о двухэтапном проведении нового конкурса: сначала на программу курса и лишь затем уже на соответствующий учебник. Секция признала необходимым при переходе на новые учебные планы сохранить информатику как самостоятельный предмет, но Гособразование СССР фактически проигнорировало это решение. Не принималась на секции ни одна из учебных программ общего курса. Без участия секции решались вопросы выбора моделей школьных компьютеров. Келейным порядком принималось соглашение о совместном проекте Гособразования СССР и фирмы IBM «Пилотные школы».

Многочисленные критические высказывания и конструктивные предложения, опубликованные в журнале «Информатика и образование», в сборниках трудов различных семинаров и конференций, имели, образно говоря, «эффект холостого выстрела». Они были полностью проигнорированы той узкой группой лиц, которая до сих пор монопольно решает все узловые вопросы школьной компьютеризации. Естественно, что в сложившихся условиях научно-педагогическая общественность вынуждена искать альтернативные пути для нормализации положения дел. Не случайно практически одновременно в различных регионах стали возникать ассоциации учителей информатики, начинающие брать на себя те функции, с которыми так и не справились ни органы управления народным образованием, ни АПН СССР. Вряд ли можно считать правильным выпол-

нение этими общественными образованиями всей той работы, которую положено делать официальным учреждениям и организациям. Далеко не все необходимое можно сделать на общественных началах. Но как мы видим, и одними только административно-командными методами наладить дело тоже нельзя. Здесь необходимо деловое и, главное, равноправное сотрудничество общественности и официальных структур. Именно в расчете на то, что такое сотрудничество все-таки удастся наладить, и пишется эта статья.

В настоящее время существуют различные, порой полярные друг другу точки зрения как на глобальные цели компьютеризации средней школы, так и на возникающие в этой деятельности конкретные проблемы. Нужен ли вообще нашей общеобразовательной школе самостоятельный курс информатики? Если да, то каким он должен быть сейчас, в каких направлениях трансформироваться? Как должен он соотноситься с компьютеризацией учебного процесса в целом? Да и нужна ли она? И в каком виде? Что, наконец, кроме общих идей и добрых пожеланий требуется для того, чтобы и этот курс, и эта компьютеризация все-таки состоялись, коль скоро мы придем к выводам об их необходимости?

Начнем с констатации того бесспорного положения, что на современном этапе научно-технического и социального прогресса компьютеризация проникает практически во все основные сферы человеческой деятельности. Естественно заключить, что школа должна готовить своих питомцев к жизни в этом компьютеризованном мире. Но вот на вопрос о том, в чем же должна состоять такая подготовка и как ее следовало бы организовать, столь очевидного ответа дать уже нельзя. Спектр различных взглядов на этот счет достаточно широк.

Точка зрения идеологов первой официальной программы курса «Основы информатики и вычислительной техники» в ее наиболее концентрированном виде выражена в [1—3]. По существу, она сводится к «овладению основными умениями алгоритмизации» и «формированию представлений об автоматизации различных видов деятельности человека на основе алгоритмов». Несостоятельность такого подхода была показана рядом авторов неоднократно, в том числе на страницах «ИНФО» [4].

В последнее время модно стало декларировать, что школьный курс информатики тем или иным образом должен быть тесно связан с НИТ: либо — базироваться на НИТ, либо обеспечивать овладение школьниками этими самыми НИТ. Главное, чтобы была связь. С такими утверждениями нельзя со-

гласиться, но и возразить на них тоже нельзя! Ведь аббревиатура НИТ (новые информационные технологии) представляет совершенно еще не устоявшееся понятие (см., напр. [5—8]). Нередко этот термин пытаются интерпретировать, вводя в прямое противоречие даже со смыслом понятия «технология».

Бытует мнение и о ненужности отдельного курса информатики. Злые языки, правда, утверждают, что здесь мы имеем дело просто-напросто с попыткой уйти от ответственности за грубые ошибки прошлых лет.

Учитывая исключительно высокие темпы развития как самой вычислительной техники, так и средств ее программного обеспечения, следует признать, что даже выпускники вузов, не говоря уже о школьниках, придя на производство, столкнутся с принципиально новой для них техникой и совершенно незнакомыми для них пользовательскими интерфейсами. Иными словами, конкретных знаний и навыков, привязанных к определенным типам ЭВМ и программным средствам, впрок дать учащимся средней школы невозможно. Следовательно, подготовка школьников должна быть построена так, чтобы максимально облегчить им восприятие тех (неизвестных сейчас) новых архитектурных и программных концепций, с которыми в будущем им придется столкнуться.

Хорошее владение клавиатурой, готовность к восприятию новых панельных и сенсорно-моторных средств, к овладению техникой работы с программно-информационными системами высокого уровня — это далеко еще не все, что должен бы получить сегодняшний школьник. Более того, чрезмерный акцент в обучении на технику и технологию работы с современными программными средствами попросту опасен, ибо может привить ученикам вредную иллюзию о якобы «всемогуществе ЭВМ». А такие иллюзии в конечном счете могут обернуться катастрофой страшнее чернойбыльской. В жизни сегодняшним школьникам придется решать реальные, а не учебные задачи. Задачи, для которых нет не только готовых рецептов, но зачастую даже и сколько-нибудь подходящих аналогов. Задачи, которые, как правило, только частично будут поддаваться формализации. И если в результате своего обучения ученики (а для того и до того — их учителя) так и не поднимутся на новую ступень научно-технического мышления, то никакие их частные успехи в алгоритмизации и программировании, в овладении приемами работы с электронными таблицами, базами данных и базами знаний, никакие самые совершенные ЭВМ впрок ни им

самим, ни нашему обществу не пойдут!

Образно говоря, ребятам, сидящим сегодня за партой, надо привить прочный иммунитет против бездумного использования вычислительной техники. Именно этот иммунитет должен впоследствии предохранить их от слепого следования рекомендациям «электронного диагноста» (ну как он посоветует в качестве средства от облысения ампутацию головы!), от опаснейших попыток полностью автоматизировать управление в таких системах, поведение которых в значительной степени является неформализуемым, и т. д.

Понимая принципиальные возможности современных компьютерных систем и получив определенные представления о ближайших перспективах, учащиеся должны в то же время отдавать себе отчет в том, что, во-первых, реальные возможности (в отличие от «теоретических») самым существенным образом зависят от имеющихся ресурсов и наличествующих ограничений и, во-вторых, далеко не все из реально возможного является еще и целесообразным.

Попросту говоря, мы должны научить школьников хорошо думать, когда речь пойдет о практических применениях вычислительной техники. А выражаясь по-ученому — дать им необходимую методологическую подготовку. Каким же образом этого можно было бы достичь?

Нельзя согласиться с теми, кто рекомендует как можно больше учебного времени уделять иллюстрациям техники работы с текстовыми и графическими редакторами, языками процедур, электронными таблицами и иными программными средствами высокого (и «сверхвысокого») уровня, переходя к их использованию чуть ли не с самого начала курса. Такое построение занятий будет аналогично демонстрации дикарям граммофона и огнестрельного оружия. Впечатляет, иной раз позволяет получить кое-какие примитивные навыки и ничего полезного не дает уму. Не имея возможности хотя бы в общих чертах понять, «как это устроено», ученики теряют интерес к предмету, более того, начинают чувствовать себя в каком-то смысле неполноценными. Естественно, что их мыслительная активность при этом будет резко падать.

А ведь на самом-то деле кроме понимания принципиальных особенностей информационно-алгоритмической структуры этих средств ученик должен хотя бы на качественном уровне представлять себе характер возможных ограничений при их практическом применении. Например, неудачный выбор расчетных формул может настолько исказить результаты, что использование электронных

таблиц будет приводить к грубым ошибкам (хорошо еще, если эти ошибки будут вовремя замечены). В других случаях с расчетами может быть все в порядке, но идеально подготовленные ЭВМ документы оказывается невозможно использовать как, не имеющие юридической силы. Объем и структура базы данных могут оказаться плохо совместимыми с выбранной СУБД. Может статься и так, что вполне, казалось бы, подходящая программная система окажется неработоспособной из-за невозможности своевременного обеспечения ее достоверной исходной информацией или же из-за сложностей с «постмашинной» обработкой выходных данных. Серьезнейшие проблемы связаны с использованием так называемых систем искусственного интеллекта.

Умалчивать об этой стороне дела (что сейчас, к сожалению, практикуется почти повсеместно) недопустимо. Такое умолчание неизбежно повлечет за собой формирование у учащихся целого спектра ошибочных представлений. А негативные последствия этих ошибочных представлений могут оказаться очень серьезными. Но ведь и чисто декларативный подход вряд ли способен дать эффект, сколько-нибудь отличный от нулевого! Едва ли не единственным выходом здесь является использование разработанных специально для учебного процесса предельно упрощенных аналогов основных типов программно-информационных систем высокого уровня. Три основные педагогические задачи должны решаться с помощью этих «примитивных аналогов»:

разъяснение учащимся принципов построения и технического функционирования системы;

демонстрация процедур инициирования и эксплуатации системы и совместное с учащимися оценивание этих процедур применительно к особенностям конкретных прикладных областей;

демонстрация учащимся примеров, иллюстрирующих ситуации выхода системы за пределы области своего нормального функционирования, с последующим обсуждением этих ситуаций.

Для того чтобы этот материал был доступен восприятию школьников, необходима, разумеется, их определенная предварительная подготовка. Как минимум она должна включать в себя изучение основ (именно и только основ!) алгоритмизации и программирования и знакомство со структурой и функционированием средств вычислительной техники. Кроме того, в качестве своего рода «переходного мостика» от этих простейших разделов к изучению современных программно-информационных систем представ-

ляется весьма желательным цикл занятий, в ходе которых сравнительно простыми средствами решались бы задачи с конкретным физическим, техническим и иным прикладным содержанием. При этом очень важно, чтобы, с одной стороны, и постановки и способы решения этих задач были полностью доступны восприятию учащихся и, с другой стороны, сущность этих задач не позволяла бы их решать без обращения к средствам вычислительной техники.

Таким образом, общеобразовательная (!) подготовка школьников по информатике должна складываться из следующих разделов.

Раздел I. Основы алгоритмизации и программирования.

Раздел II. Структура и функционирование средств вычислительной техники.

Раздел III. Решение задач с прикладным содержанием (1-й уровень).

Раздел IV. Эволюция средств программно-обеспечения ЭВМ.

Раздел V. Вычислительная техника в народном хозяйстве. Компьютерная культура современного специалиста.

Рассмотрим теперь вопрос о том, какими же способами эту подготовку можно было бы обеспечить. Один только приведенный перечень разделов убедительно свидетельствует о том, что самостоятельный курс информатики безусловно необходим. В рамках одной или нескольких традиционных школьных дисциплин (например, математики или математики и физики и т. д.) реализовать требуемое обучение невозможно, даже (!) если бы мы располагали хорошо подготовленными для этого учительскими кадрами.

С другой стороны, изучение раздела III и в какой-то мере разделов IV и V могло бы получить сильное подкрепление от других учебных предметов, а в идеале — даже быть туда частично перенесено. Но для этого обязательно надо фундаментально пересмотреть содержание обучения, изменить его так, чтобы привлечение вычислительной техники в учебный процесс по этим дисциплинам было в полной мере оправдано сущностью решаемых там задач. В противном случае можно будет рассчитывать лишь на дискредитацию вычислительной техники в глазах учащихся.

В курс математики целесообразно было бы включить построение графиков и исследование функций достаточно широких классов, а не только того узкого перечня, как это имеет место сейчас. В определенной мере могут быть затронуты вопросы исследования одно- и двухпараметрических семейств кривых. Достаточно обстоятельно могли бы быть представлены в этом курсе вопросы прибли-

женного решения алгебраических уравнений и систем, приближенного вычисления площадей, объемов, длин дуг. Разумеется, численные методы здесь не должны противопоставляться аналитическим и вытеснять их за пределы курса, необходимо их грамотное сочетание и сопоставление.

Попытки реализации в школьном курсе математики классического подхода к построению основ теории вероятностей оказались, как известно, неудачными. Вычислительная техника дает великолепную возможность включить в этот курс даже более широкую тематику: основы математической статистики и теории вероятностей. Естественно, что начинаться изучение этого раздела будет уже именно с базовых понятий математической статистики, а вероятность затем будет вводиться как «теоретический аналог» понятия относительной частоты.

Широкое использование в курсе физики методов дискретного моделирования позволяет совершенно по-новому организовать изучение целого ряда ее разделов. Обработка результатов прогноза моделирующих программ может использоваться для иллюстрации качественной картины явления или процесса, для получения приближенных формул, описывающих те или иные зависимости. Наконец (что крайне важно с методологической точки зрения!), появляется хорошая возможность на вполне доступном для учащихся материале наглядно показать им, что результат решения той или иной физической задачи совсем не обязательно должен получаться расчетом по какой-то формуле, а может быть итогом выполнения некоторой машинной программы. Иными словами, непосредственно в своей учебной деятельности школьники будут находить подтверждение тому, что реальные зависимости между величинами могут описываться не только аналитически, таблично или графически, но также еще и алгоритмически.

Громадны потенциальные возможности ЭВМ и для кардинальной перестройки преподавания других дисциплин, составляющих учебные планы средней школы.

Три существеннейших обстоятельства препятствуют сейчас немедленной реализации обозначенного здесь подхода. Первое из них — это катастрофическая нехватка учебной вычислительной техники. Ведь действительная, а не показная компьютеризация того же курса математики или физики возможна только при том условии, что и ученики и учитель имеют постоянный, а не эпизодический доступ к ЭВМ.

Второе обстоятельство — почти полное отсутствие соответствующим образом подготовленных учительских кадров. Практикуе-

мое сейчас одно-двухнедельное «обучение предметников компьютеризации» нельзя расценивать иначе, как издевательство над здравым смыслом. В лучшем случае за такой мизерный срок можно дать учителю лишь отдельные начальные навыки работы с ЭВМ, но никак не подготовить его к сколько-нибудь эффективному использованию этой техники на уроках. Более того, как ни грустно это признавать, но у нас сейчас нет ни преподавателей, готовых такое обучение учителей произвести, ни даже сколько-нибудь законченной идеологии этого обучения.

Наконец, третье обстоятельство заключается в том, что нет ни новых учебных программ для переводимых на «электронные рельсы» школьных предметов, ни тем более учебных и методических пособий, педагогических программно-информационных средств. Нет, видимо, даже и должного понимания необходимости их создания ни у руководства Гособразованию СССР, ни в АПН СССР.

Нет сомнения в том, что в кратчайшие сроки фронтально выправить положение дел невозможно. Реально сейчас можно говорить об осознании существующих здесь проблем, определении очередности их решения и о практических шагах, эти решения обеспечивающих.

В настоящее время самой болезненной из этого круга проблем является проблема подготовки учителей информатики. Действительно, учитель любого другого предмета сначала изучал его в школе, будучи сам учеником, затем на протяжении 4—5 лет — в институте. И наконец, приступая к преподаванию, мог пользоваться советами и опытом своих старших коллег, следовать сложившимся традициям и т. д. Преподаватель информатики всего этого лишен. Наивно думать, что все вопросы его подготовки могут быть решены за те 72 ч, которые были для этих целей выделены. Как показывает и отечественный, и зарубежный опыт, даже 300—500 ч оказывается явно недостаточно. Учителя информатики должны иметь возможности для своего образования не меньше, чем это сейчас определено для врачей, инженеров и других специалистов народного хозяйства [9]. По крайней мере — годовичное, а лучше — двухгодичное обучение с полным отрывом от работы. И сверх этого — периодическая краткосрочная переподготовка, позволяющая быть в курсе новостей как в области самой вычислительной техники, ее программного обеспечения, так и в сфере их применения в учебном процессе.

Эта проблема первоочередная, ибо если компьютеризация даже таких учебных предметов, как физика и математика, — в основ-

ном дело будущего, то курс информатики — это реалии сегодняшнего дня. Он уже преподается практически повсеместно, и, как мы видели, чаще всего преподается плохо. Пока мы не будем иметь достаточно хорошо подготовленных учителей информатики, никакими иными мерами: великолепной техникой, самыми совершенными учебниками и комплексами педагогических программно-информационных средств изменить положение не удастся.

Напротив, наличие корпуса получивших полноценную профессиональную подготовку педагогов будет оказывать существенное положительное влияние и на решение вопросов технического, методического, программного и иного обеспечения учебного процесса. А самое главное — их ученики будут заканчивать среднюю школу, обладая необходимым для ее выпускника уровнем компьютерной культуры.

Ориентируясь в первое время на одногодичный срок обучения, можно считать реальным завершение этой начальной подготовки основной массы учителей информатики того или иного региона в течение трех-четырех лет. Примерная стоимость такого обучения (в ценах середины 1990 г.) составит, видимо, величину порядка 5000 руб. на одного учителя. Это, разумеется, складывается в небольшую сумму, однако все-таки значительно меньшую того, что до сих пор растрачивалось фактически впустую. В дальнейшем с учетом ожидаемого пополнения учительских рядов выпускниками педагогических институтов, обучавшихся там как раз преподаванию информатики, окажется возможным переход на режим переподготовки и повышения квалификации.

Содержание предполагаемого одно-двухгодичного цикла обучения прежде всего должно обеспечивать высокий уровень общенаучной и методологической (не путать с методической) подготовки педагога. Только на этой основе и может учитель прийти к правильному представлению о месте и роли информатики в различных направлениях человеческой деятельности, а отсюда уже и к подлинному пониманию целей и задач преподавания соответствующей учебной дисциплины. Учителю нужны достаточно детализированные знания не только и не столько (!) о существующих социальных, научных и инженерных достижениях, об используемых на практике организационных и технических решениях, сколько об их генезисе, о тех качествах, которые оказываются необходимыми специалистам-исследователям, разработчикам конструкций и технологий, эксплуатационникам и пользователям тех или иных сервисных систем, руководителям самых разных уровней и рангов.

Отсутствие таких знаний лишает педагога возможности (сколь бы высока ни была его узкопрофессиональная подготовка) самостоятельно ориентироваться в вопросах определения содержания и характера действительно необходимого его ученикам обучения. Он оказывается вынужденным слепо следовать кем-то другим подготовленным планам занятий и методическим разработкам, при этом интерпретируя их далеко не всегда хоть в сколько-нибудь приемлемом соответствии с авторским замыслом. Такой учитель не в состоянии правильно адаптировать свой курс ни к реальным условиям преподавания, ни к действительным потребностям обучаемых.

Обеспечение должной общенаучной и методологической подготовки педагогов представляет собой весьма сложную задачу. Ведь даже в тех учебных заведениях, где ведутся переподготовка и повышение квалификации руководящих и инженерно-технических кадров различных отраслей народного хозяйства, далеко не всегда ее удается достичь. К сожалению, среди профессорско-преподавательского состава вузов и ИПК широко распространены искаженные представления как о характере и сущности связанной с использованием ЭВМ производственной деятельности многих категорий работников, так и о необходимых им для этого знаниях, умениях и навыках. Соответствующее преподавание и в инженерно-технических, и особенно в педагогических учебных заведениях носит, как правило, чисто декларативный характер. Изложение учебного материала в этой части обычно оказывается весьма поверхностным и содержит немало фактических ошибок и неточностей. Иными словами, учителям здесь очень не хватает как собственных достаточно квалифицированных учителей, так и достаточно действенных средств и методов обучения.

Бурная рекламно-коммерческая деятельность фирм-производителей вычислительной техники и программного обеспечения даже в наиболее развитых странах заметно мешает правильной оценке и оптимальному выбору пользователями действительно нужных им изделий и систем, нередко толкает их на неоправданно частую замену используемых технических и программных средств. В нашей же стране, где средний уровень информационной культуры существенно ниже, куда необходимые сведения из-за рубежа поступают фрагментарно да к тому же весьма своеобразно фильтруются и интерпретируются представителями организаций, имеющих к этим сведениям преимущественный доступ, рядовому пользователю

приходится еще хуже.

В условиях недостаточной компетентности основной массы пользователей компьютерных систем процветают всевозможные спекуляции, сплошь и рядом происходит продажа негодной продукции, проталкиваются дорогостоящие, но бесполезные (а порой и вредные) проекты. Всем этим негативным тенденциям учитель должен уметь противостоять сам и готовить к такому противостоянию своих учеников.

Нереально да и не нужно, чтобы учитель информатики знал все. (Здесь мы не говорим о тех, кто занят обучением профессионалов. Эта тема — предмет отдельного обсуждения. Информатика в школе — это курс общеобразовательный!) Учитель должен быть знаком с вычислительной техникой и средствами ее обеспечения настолько, чтобы, во-первых, грамотно использовать их в учебном процессе и, во-вторых, видеть направленность их развития, ощущать грань между возможным и невозможным в автоматизации решения основных типов задач, возникающих в реальных условиях. Образно выражаясь, учителю в первую очередь нужны не детали, а принципы, не конкретные способы и приемы, а общие методы, не виртуозная техника общения с ЭВМ, а понимание тенденций информатизации общества в целом и применительно к образованию в частности.

Естественно, остается необходимой для учителя и методическая подготовка. Однако характер этой подготовки по сравнению с тем, что от нее требовалось в прежние времена, коренным образом меняется. Дело в том, что в условиях «нарастающей компьютеризации» учебного процесса приходит конец относительной стабильности как содержания учебных предметов, так и методики обучения. В первую очередь это относится к курсу информатики. Учитель должен самостоятельно отбирать и структурировать учебный материал в зависимости от конкретных условий проведения занятий. (А эти условия будут постоянно изменяться.) Он должен быть в состоянии применительно к этим условиям создавать свои и адаптировать существующие методики обучения. Эти умения необходимы учителю даже в тех случаях, когда им будут использоваться лишь стандартные учебные программы, учебники и учебные пособия, уже существующие педагогические программно-информационные средства.

Особо здесь следует выделить ту компоненту в подготовке учителя, которая должна дать ему возможность организовать эффективное управление учебным процессом. Используемые сейчас модели такого управления, несмотря на все их видимые и скрытые

различия, как правило, не обеспечивают ничего иного, кроме подтягивания учащегося (да и то не всегда) до так называемого репродуктивного уровня усвоения учебного материала. На развитие творческих способностей, приобретение новых навыков и умений мышления эти модели и основывающиеся на них методы управления никакого (во всяком случае, положительного) влияния не оказывают. Один из возможных подходов к построению действенной системы управления обучением описан в [10]. Разумеется, он не является единственно возможным, и учитель должен быть обстоятельно ознакомлен с различными способами организации управления учебным процессом на каждом из уровней этого управления.

Следующей по своей актуальности является проблема обеспечения школ учебной вычислительной техникой. Сейчас этой техники крайне мало, и то, что руководители школ, латая дыры, вынуждены зачастую приобретать любые, самые неподходящие ее образцы, — это вынужденный шаг. Ведь альтернативой здесь является так называемое безмашинное изучение основ информатики и вычислительной техники — прямой аналог обучения плаванию без воды. Но разумеется, дальше с такой «самодеятельностью» мириться нельзя. Можно и нужно выработать и проводить в жизнь хорошо продуманную, реалистичную техническую политику.

Именно от того, какой будет эта политика, во многом зависит, выйдет ли школьная компьютеризация за границы скромного курса информатики, станут ли наши школьники по-новому, с широким использованием вычислительной техники изучать физику, математику, языки и другие предметы или же дело ограничится шумовыми эффектами вокруг нескольких локальных экспериментов. Попросту говоря, не может быть компьютеризации там, где нет для этого компьютеров. Поэтому абсолютно неприемлемым следует признать рекомендации опекаемых Гособразованием СССР специалистов о принятии в качестве основных (!) моделей школьных компьютеров ПЭВМ типа IBM PC (или PS/2!). Эта техника очень дорогая, поступает (и, видимо, будет поступать) она к нам в небольших количествах. Ее не хватит даже на то, чтобы обеспечить потребности курса информатики. Странно, что столь очевидный факт руководство Гособразования СССР не замечает!

В ближайшие годы наиболее реалистичной представляется следующая схема обеспечения учебных занятий средствами вычислительной техники:

1. Программируемые микрокалькуляторы (типа МК-52, МК-85 и т. д.) — для систе-

математического использования на уроках математики и физики, эпизодического использования в других дисциплинах. Таких кабинетов в каждой (не малокомплектной) школе должно быть несколько.

2. Простейшие ПЭВМ (типа ПМ1-ПМ2 по ГОСТу 27201—87 или даже несколько выходящие за нижнюю границу этого ГОСТа — как БК-0010) — для систематического использования на уроках информатики и эпизодического — на уроках других дисциплин и во внеклассной работе. Желательно в каждой школе иметь один такой кабинет.

3. ПЭВМ типа ПМ3 (по ГОСТу 27201—87) — для изучения отдельных тем курса информатики и внеклассной работы. Кабинеты, оснащенные такой техникой, создавать в каждой школе нецелесообразно. Вполне достаточно иметь один кабинет на несколько школ.

На каждой из перечисленных групп вычислительной техники будут решаться свои классы учебных задач. Поэтому никаких особых требований типа «совместимости снизу вверх» тут не возникает. Совершенно иначе дело обстоит внутри этих групп. Не только внутри района или области, но и в пределах республики и страны в целом должны происходить интенсивный обмен ППИС, тиражирование и сопровождение вновь создаваемых, поэтому используемые ЭВМ должны быть в максимально возможной степени совместимы друг с другом. На уровне второй группы этого достичь будет труднее всего, ибо многие школы уже оснащены такой техникой и не то, что не хотят, а просто не могут себе позволить с ней расстаться, ибо не смогут в силу ряда причин оперативно ее заменить. Поэтому положение о внутригрупповой совместимости используемых ПЭВМ, о соблюдении принципа преемственности при переходе к новым модификациям и моделям относится прежде всего к той технике, которая еще только будет планироваться и создаваться для школы.

Здесь необходимо особо остановиться на программируемых микрокалькуляторах. К сожалению, ни школам, ни микрокалькуляторам не повезло. Министерство просвещения СССР фактически «опустило шлагбаум» перед этой техникой, объявив ее неперспективной. Не было сделано заказов на ее массовое производство. Новые модели микрокалькуляторов выпускаются мизерными партиями. Такое положение, на мой взгляд, следовало бы как можно скорее кардинально изменить. Иначе компьютеризация в обучении физике, математике, другим дисциплинам будет задержана на несколько лет.

Обеспечение школы вычислительной тех-

никой обязательно должно сопровождаться созданием и развитием системы ее технического обслуживания. При этом желательно, чтобы вышедший из строя компьютер или аппаратура передачи данных при невозможности немедленного устранения неполадок заменялись на время своего ремонта такими же машинами и аппаратурой из фондов обслуживающих организаций.

Более подробно точка зрения автора на проблему школьного компьютера изложена в [11].

Получение большей самостоятельности республиканскими и местными органами народного образования, школами и другими учебными заведениями не означает, естественно, их полного перехода на «натуральное хозяйство». По-прежнему будут нужны и централизованным образом издаваемые учебники и учебные пособия, и организации, ведущие распространение и сопровождение ППИС. Одной отдельно взятой школе или району «на самообслуживании» не прожить. Другое дело, что теперь требуется несколько различных учебников, различные комплексы ППИС, рассчитанные на разные условия преподавания.

Вряд ли можно считать правильным, что учебник у нас сначала создается, а уже после этого по нему начинают вести экспериментальное преподавание. Первичным все-таки должен быть эксперимент. Кроме того, одним только учебником проблемы обеспечения преподавания не решаются. Это относится прежде всего к преподаванию информатики, но скоро в таком же положении будут и другие дисциплины. Требуемый учебно-методический комплекс кроме учебника или учебного пособия должен включать разнообразные ППИС, инструктивные и методические материалы для учителя, различного рода демонстрационные материалы и т. д.

Разработка и уж во всяком случае апробация такого комплекса должны проходить в условиях реального преподавания. Оценку такого рода разработок следует производить на основе детального ознакомления не только с разработанными материалами, но и с тем, как они фактически используются в учебном процессе, в какой мере успешно усваивается обучаемыми учебный материал, как в ходе обучения развивается их мышление. В противном случае объективная оценка своевременно получена быть не может (что хорошо уже подтверждено опытом).

Рассчитывать на то, что вся эта громадная работа будет выполняться, как говорят, на голом энтузиазме, нельзя. Такая деятельность должна получать всестороннюю, в том числе и материальную, поддержку в регионе

и со стороны центра.

Конкурсный отбор из числа претендующих на проведение эксперимента целесообразнее всего проводить в крупных регионах. Для такого отбора, для оценки хода и результатов эксперимента и для выработки соответствующих рекомендаций в регионе требуется создать комиссию из высококвалифицированных специалистов. Такая комиссия должна получать как организационную, так и финансовую поддержку, иметь возможность оплачивать работу привлекаемых экспертов. Естественно, что далеко не в каждом регионе целесообразно вести такие эксперименты. С другой стороны, полученные в каком-либо регионе удачные результаты могут быть использованы и в других.

Учитывая вышеизложенное, проблему, требующую первоочередного решения, можно сформулировать так: создание механизма, обеспечивающего разработку и совершенствование учебно-методических комплексов, их квалифицированную и всестороннюю экспертизу, отбор и тиражирование лучших решений. Еще раз подчеркнем, что этот механизм должен быть создан как на региональных, так и на республиканских и на союзном уровнях.

В этой статье не случайно содержится много резкой критики в адрес как старого, так и нового руководства народного образования. Дело в том, что сегодня невозможно добиться успеха, действуя командно-административными методами, слегка лишь закамouflированными марионеточными общественными структурами. Необходим совершенно новый стиль работы. Безусловно, необходимо равноправное сотрудничество и взаимодействие научно-педагогической общественности и существующих административных структур.

Уже после того, как эта статья была подготовлена к печати, автору довелось познакомиться со своеобразным очерком истории компьютеризации обучения за рубежом [12]. Хотелось бы обратить внимание

читателей на этот материал, во многом перекликающийся с настоящей статьей.

Литература

1. *Монахов В. М.* О содержании курса «Основы информатики и вычислительной техники» // Математика в школе. 1985. № 3. С. 7—12.
2. *Ершов А. П., Монахов В. М.* и др. Изучение основ информатики и вычислительной техники. Ч. I. М.: Просвещение, 1985.
3. *Ершов А. П., Монахов В. М.* и др. Изучение основ информатики и вычислительной техники. Ч. II. М.: Просвещение, 1985.
4. *Матюшкин-Герке А. А.* Каким быть школьному курсу информатики // Информатика и образование. 1987. № 6. С. 105—111.
5. *Велихов Е. П.* Новая информационная технология в школе // Информатика и образование. 1986. № 1. С. 18—22.
6. Концепция информатизации образования (материалы для обсуждения) // Информатика и образование. 1988. № 6. С. 3—31.
7. *Шатров А., Цевенков Ю.* Проблемы информатизации образования // Информатика и образование. 1989. № 5. С. 3—9.
8. *Барсуков В. С., Тарасов О. В.* Новая информационная технология: искусственный интеллект, концепция банка знаний, экспертные системы // Вычислительная техника и ее применение. 1989. № 2. С. 39—45.
9. О перестройке системы повышения квалификации и переподготовки руководящих работников и специалистов народного хозяйства. Постановление ЦК КПСС и СМ СССР от 06.02.88 г. № 166.
10. *Матюшкин-Герке А. А.* Основные виды обеспечения АСУ процессом обучения // Автоматизированные обучающие системы. Казань, 1979.
11. *Матюшкин-Герке А. А.* Каким же быть школьному компьютеру? // Информатика и образование. 1988. № 5. С. 103—105.
12. *Борк А.* Компьютеры в обучении: чему учит история // Информатика и образование. 1990. № 5. С. 110—118.



Три учебника информатики

Скоро три года, как завершился конкурс школьных учебников по информатике. В нынешней ситуации, когда нет жестких предписаний из центра о единственно правильном преподавании курса, перед учителем стоит проблема выбора. Полагая, что идея курса лучше всего просматривается в учебнике, обсудим те из них, которые сегодня предлагаются вниманию учителя, обращая внимание в основном на научную состоятельность авторской позиции.

В. А. Каймин, А. Г. Щеголев, Е. А. Ерохина, Д. П. Федюшин. Пробное учебное пособие для X — XI классов средней школы. М.: Просвещение, 1989. 272 с. с ил. Текст разбит на 8 глав и 64 параграфа, в качестве приложений приводится справочник по Бейсику и терминологический словарь.

Данный учебник является доработанным вариантом представленного на вышеупомянутый конкурс учебника, занявшего второе место. Напомним, что в соответствии с формулировкой конкурсной комиссии рукопись была рекомендована не к использованию в преподавании информатики, а к дальнейшей значительной доработке. Ряд направлений такой доработки был приведен в статье «Конкурс на учебник информатики» (ИНФО, 1988, № 1). Можно констатировать, что доработка проводилась и пошла учебнику на пользу. Мы держим теперь в руках не ротапринтное, а хорошо иллюстрированное, прекрасно скомонованное издание. Можно сказать больше: перед нами образцовое полиграфическое издание. Однако доработка текста носит по преимуществу поверхностно-косметический характер. Глубинные содержательные претензии можно почти один к одному повторить и сегодня. Отметим лишь основные.

Содержание § 7 «Законы логики» ясно показывает, насколько бурное развитие логики XX в. не пошло на пользу авторам. Полностью игнорируя эту славную главу истории науки, авторы предлагают учащимся, на мой взгляд, наивно-аристотелевскую позицию. И дело не в том, что классической логике нельзя учить (можно), и не в том, что такое обучение не имеет дидактической ценности (имеет). Существует много внутренне непротиворечивых и по-своему интересных логик. Но как только мы лишаем ту или иную схему ее условно-модельного характера, объявляя самой реальностью, как тут же она

приходит с этой реальностью в противоречие. На с. 29 читаем в рамочке: «Истинно либо суждение, либо его отрицание». Простейший пример утверждения, которому нельзя приписать ни истинного, ни ложного значения, приводился мною в вышеупомянутой статье. Это широко известный парадокс лжеца. Приведем его еще раз. Цитирую по Э. Мендельсону («Введение в математическую логику». М.: Наука, 1971. С. 8). «Некто говорит: «Я лгу». Если он при этом лжет, то сказанное им есть ложь и, следовательно, он не лжет. Если же он при этом не лжет, то сказанное им есть истина и, следовательно, он лжет. В любом случае оказывается, что он лжет и не лжет одновременно.

Кроме этого примера там есть целый список, который можно с легкостью продолжить. При этом мы видим, что дело не в каких-то таинственных проблемах математики, связанных с бесконечными множествами, как продолжают нас уверять авторы (с. 214). Все гораздо хуже, т. е. лучше. К тому же всякая хозяйка знает, что если картофель «не плохой», это вовсе не означает, что он «хороший». Это о законе двойного отрицания (с. 29, нижняя рамка), точнее, о том, что модель-то эта (классическая логика) довольно грубая, поэтому нередко, в частности при построении экспертных систем, используются иные логики — полимодальные, вероятностные, нечеткие и пр. И наконец, в заключение § 7 нам преподносится вырванная из контекста и обведенная в голубую рамочку цитата из Лейбница: «Любое утверждение должно предполагать наличие аргументов и фактов, достаточных для его обоснования».

Контекст известен. Напомним читателю, что Лейбниц, а с ним и Декарт полагали, что всякое истинное утверждение выводимо путем формальных выкладок, т. е. чисто механически, и Лейбниц разрабатывал машину для такого рода вычислений. Эта линия, идущая от Аристотеля через Декарта и Лейбница и далее, вливается в мощный поток современной информатики, но не следует забывать, что на ее пути была теорема Геделя, которая показала, что Лейбниц в этом был совсем неправ (см. в связи с этим глубоко и живо написанную статью В. Н. Тростникова: Новый мир, 1989, № 12. с. 257—263). А если лишить высказывание в рамочке его

исторического контекста, то мы повергаемся в полное недоумение. По-видимому, многим из вас, уважаемые читатели, приходилось, и возможно не раз, утверждать, что ваша мать действительно является вашей матерью. Но авторский коллектив во главе с В. А. Кайминым требует аргументов и фактов, достаточных для его обоснования. А как быть, если их нет или они, строго говоря, недостаточные? Мне возражат: «Э, куда хватил! Одно дело жизнь, а другое — логика». Конечно, но в учебнике не сделано ни малейшей попытки отделить в сознании учащихся сферу модельных построений и нашу суровую реальность. Причем эта позиция проводится авторами последовательно и достигает своей кульминации в § 48 «Законы информатики» главы 6-й «Основная информатики», где учащимся для заучивания предлагаются шесть «законов информатики». Термин «закон» в зависимости от контекста может получать различное значение. Так закон в научном контексте — это одно, а в юридическом — несколько иное. Так вот, появление словосочетания «законы информатики» отсылает нас к первому, научному смыслу, но чтение формулировок и комментариев к ним убеждает нас в том, что мы ошиблись и перед нами нечто вроде войскового устава или призывов по революционному преобразованию действительности. Приведем формулировки.

Закон № 1. Предмет обсуждения должен быть строго определен и не должен меняться до конца обсуждения.

Закон № 2. Не может быть одновременно истинно суждение и его отрицание.

Закон № 3. Истинным может быть либо суждение, либо его отрицание.

Закон № 4. Всякое рассуждение необходимо доводить до определенного утверждения или его отрицания.

Закон № 5. Утверждения необходимо доказывать, исходя из суждений, истинность которых доказана.

Закон № 6. Всякое построение (предложение) должно иметь строгое обоснование, опирающееся на строго установленные факты и положения.

Похоже, что как только утверждение приобретает императивный характер, то сразу становится законом. Читатель без труда сможет сформулировать законы № 7, 8 и т. д.

Далее. Основной методической задачей курса провозглашается обучение технологии решения задач, основанной на технике построения алгоритмов с одновременным доказательством их правильности. Представьте себе, что вы учите ребенка ходить. Это непростая задача, это трудно. А теперь представьте, что вы учите его ходить с одновременным обучением контролю за правиль-

ностью походки. Задача становится невозможной трудной. Так и здесь. Обучение составлению алгоритмов — это серьезная работа, и если бы мы с ней справились, было бы очень хорошо. А работа по доказательству правильности алгоритмов — это другая, существенно более трудная деятельность, причем основанная на другой технике, требующая других, более сложных учебных навыков. Если с известной степенью условности оценить отношение трудовых затрат на доказательство правильности алгоритма к трудовым затратам на его составление и отладку, то это величина порядка 10 для обычных учебных алгоритмов, которая быстро растет с ростом его сложности. И не случайно, что примеры из учебника — комбинация двух-трех присваиваний. Те рекомендации, которые даны в учебнике в качестве методов доказательства, вызывают серьезные сомнения. Действительно, предложение доказывать правильность любых циклических алгоритмов по индукции, мягко говоря, опротивело, как и предложение делить сложные алгоритмы на простые (а если не делится?). Да и само стремление писать абсолютно безошибочные алгоритмы является технологическим просчетом. Как известно, можно достигать любого повышения надежности алгоритма не снижением процента ошибок, а чисто структурными решениями (надежные сооружения из ненадежных элементов). А потом ведь и в доказательствах бывают ошибки, и уж если на то пошло, то необходима технология доказательства правильности доказательств и т. д. Это стремление достигать абсолютной точности, формализованности звучит лейтмотивом во всей книге. Например, на с. 140 читаем:

«Точные, математические строгие постановки задач — ключ к успеху в любой работе — на производстве, в науке и технике, в учебе и в административной деятельности, а не только при решении задач на ЭВМ».

Ну ладно, в любой, так в любой. Или (с. 33):

«Мы будем говорить, что способ решения правильный, если он дает требуемые результаты при любых допустимых исходных данных. Соответственно мы говорим, что способ решения неправильный, если эти условия не выполнены».

Есть несколько хороших алгоритмов для решения уравнений математической физики. Физики десятилетиями используют их в том числе в тех ситуациях, когда математически задача является некорректной, и очень скептически относятся к замечаниям на этот счет. Заметим, что провести точное описание тех данных, на которых разностная схема корректна, — трудная задача, за которую и браться-то никто не будет. Да и что далеко

ходить! Как уже отмечалось, в учебнике предлагается доказывать правильность циклических алгоритмов по индукции. Теперь давайте рассуждать. Либо этот метод правильный, либо неправильный (см. закон № 3). Мы далеки от мысли, что авторы хотят научить школьников неправильному методу. Следовательно, он правильный и удовлетворяет определению правильного метода. Теперь на основании закона № 6 мы можем (а может быть, и должны) потребовать обоснования (с соблюдением законов № 4 и 5) этой правильности. То, что его нет в учебнике, не беда, авторы, видимо, им располагают, но не поместили из-за недостатка места и готовы представить его по первому требованию. В частности, они должны уметь отвечать на такой вопрос: «Правильность каких циклических алгоритмов можно доказывать по индукции (и как), а каких нельзя (и почему)?»

В заключение хочу признаться, что тогда, в упомянутой статье, я нарушил закон № 4 (см. выше). Спешу исправить положение и сделать свой вывод окончательно определенным: данное пробное учебное пособие не соответствует требованиям, предъявляемым к современным учебникам, совершенно не удовлетворяет требованиям научности. И то, что на его подготовку затрачено много труда и денег, не нахожу достаточным основанием для того, чтобы тратить их впредь.

А. Г. Кушниренко, Г. В. Лебедев, Р. А. Скворень. Учебник для средних учебных заведений. М.: Просвещение, 1990. 224 с.: с ил. Состоит из введения и трех глав, разбитых на 28 параграфов, которые в свою очередь делятся на подпункты. Распределение параграфов по главам таково: введение — 3 параграфа, 1-я глава «Алгоритмический язык» — 13, 2-я «Устройство ЭВМ» — 4, 3-я — «Применение ЭВМ» — 8. Имеется предметный указатель.

Занятия информатикой почти в любом ее жанре используют и опираются на некий фундаментальный умственный навык, который хотя и не принадлежит одной лишь информатике и является в известной мере общеучебным, но для информатики становится крайне существенным. Речь идет о навыке работы с формально-знаковыми конструкциями. Этот навык имеет два взаимосвязанных аспекта: статический — описание, представление, моделирование чего-либо формально-знаковыми средствами и динамический, т. е. формальные преобразования формально-знаковых конструкций. Данный курс по преимуществу направлен на формирование именно этого умственного навыка, на формирование именно этой интеллектуальной техники. При этом традиционно основной акцент поставлен на алгоритмическом, процедурном

аспекте этого навыка. В 13 параграфах 1-й главы идет работа по обучению составлению алгоритмов. Однако и это в основном относится к последнему изданию, наращиваются усилия, направленные и на первый, статический, объектно-модельный аспект: непосредственно в 3-й, а косвенно и во 2-й главе идет работа по отработке навыков описания информационных моделей исполнителей. И это происходящее на наших глазах перераспределение акцентов в учебном предмете естественно связать с теми глобальными процессами, которые формируют лицо информатики, и которые в приложении к программированию знаменуются заменой процедурной парадигмы на объектную, и в которых принимают активное участие авторы учебника (см.: А. Г. Кушниренко, Г. В. Лебедев, Программирование для математиков. М.: Наука, 1988).

Для достижения этой непростой цели авторы предлагают хорошо продуманную, проработанную, грамотно, ясно и доступно изложенную систему обучения. Тот, кто пройдет этот курс, прорешав предложенные задания, получит хорошую школу.

Во 2-й главе объяснение устройства ЭВМ ведется в круге тех же основных идей и понятий курса: ЭВМ как ансамбль исполнителей, работающий под управлением процессора.

Однако необходимо отметить, что на пути данного курса в школу есть немалые трудности. Стиль учебника, вполне адекватный излагаемому материалу, точный и лаконичный — стиль современной научной литературы — довольно сильно отличается от сложившегося стиля советских учебников. Далее, несмотря на всю доступность и посильность учебных заданий, учебник рассчитан на постоянное интеллектуальное усилие со стороны как ученика, так и учителя. И если работа учителя может быть сильно упрощена вспомогательными методическими разработками с решениями всех задач, то ученик, не давший внутреннего согласия на такую постоянную работу, не получит от курса почти ничего. Поэтому это учебник в первую очередь для тех, кто согласен работать. Но, смею заверить, что те, кто возьмет на себя этот труд, не пожалеют об этом.

И еще пара штрихов. Известным диссонансом ясному и лаконичному языку учебника звучит все-таки довольно туманное и романтическое введение (первые страницы), где упор сделан на апелляцию к чувствам. На с. 201 утверждается, что есть уравнения, которые «математика точно решать не умеет». Речь, по-видимому, идет о возможности представления решения в виде композиции элементарных функций, и следовало бы или

сказать это точнее, или промолчать.

К учебнику с самых разных позиций можно предъявить большие претензии за то, чего там нет. Но мы, конечно, помним, что курс написан под фиксированное количество учебных часов. И если у вас мало денег, а вы голодны — купите хлеба.

А. Г. Гейн, В. Г. Житомирский, Е. В. Линецкий, М. В. Сапир, В. Ф. Шолохович. Пробное учебное пособие (машинный вариант). Свердловск: Изд-во Уральского ун-та, 1989. 272 с.: с ил. Предисловие, 10 глав, 36 параграфов, 20 лабораторных работ, имеется предметный указатель.

Информатика, компьютеры, алгоритмы и пр. и пр. — для человека, не знакомого с этим или знакомого лишь по наслышке, возникает некая загадочная, возможно, пугающая, а главное — чуждая сфера. Ввести человека в этот мир, познакомить с ним, причем сделать такое знакомство по возможности легким и увлекательным — вот та благодарная задача, которую успешно решает этот учебник. Живым образным языком преподносится со вкусом отобранный материал (обо всем, и ничего лишнего), дающий вполне сбалансированное представление о новом предмете и виде человеческой деятельности. Причем практикум, пронизывающий курс, позволяет многое, так сказать, потрогать руками, проверить на деле. Конечно, при таком подходе неизбежна определенная поверхностность, отчасти компенсированная наличием более сложных и, по-видимому, необязательных заданий. Но как мы уже отмечали, число учебных часов фиксированно и чем-то жертвовать необходимо.

При этом можно констатировать, что та теоретическая концепция предмета, которая неявно просматривается в тексте, представляется научно состоятельной. Акценты, как правило, расставлены точно.

Позволим себе все же два существенных замечания.

Проследим логику введения понятий «алгоритм» и «исполнитель». Пара житейских примеров (с. 24), описательная формулировка понятия «алгоритм» (организованная последовательность действий, с. 25), серия из 12 задач, из которых лишь в четырех ситуация более-менее формализована. Далее вводится понятие «исполнитель» через список примеров: человек, робот, станок, живая клетка, животное в цирке (с. 28). Как написано на с. 26, «Понятие алгоритма в информатике является фундаментальным. Таким же, каким является понятие точки, прямой и плоскости в геометрии...». Да. Но если мы при введении понятия «точка» продемонстрируем кляксу и букашку, при пояснении понятия «твердость» предъявим лимон, при обучении

музыкальной грамоте так и не дадим чистый тон, то мы можем и не достигнуть успеха. И если в списке исполнителей формальные исполнители — это лишь примеры среди примеров, то давайте поставим мысленный эксперимент: уберем их из списка и будем строить предмет без них. Ничего не получится, нет такой науки. По отношению ко всем этим неформальным «алгоритмам» и неформальным «исполнителям» мы можем лишь строить аналогии, т. е. заниматься моделированием их формальными средствами. Итак, подчеркнем, что как понятия алгоритм и исполнитель — это формальный алгоритм и формальный исполнитель. Но эти общие понятия, универсалии с тем или иным успехом применимы для описания и изучения множества неформальных ситуаций. И шутивное сравнение движения Луны с выполнением циклического алгоритма нельзя признать удачным по многим причинам, в частности потому, что движение Луны является не периодическим, а лишь квазипериодическим, т. е. хотелось бы, чтобы в этой шутке было больше истины.

И еще. Каждый раз, говоря о моделировании, авторы говорят о математическом моделировании. Это терминологическая неточность, многократно закрепленная в официальных документах. Что такое модель? Это отражение объекта в той или иной среде теми или иными средствами. Гипсовая модель — изображение объекта в гипсе. Математическая модель — отражение объекта в гипсе. Математическая модель — отражение объекта математическими средствами (числа, функции и т. п.). И если бы те модели, которые реализуются на компьютере, были математическими, то сами математические объекты допускали бы прямую и адекватную компьютерную реализацию. Но это не так. И не только по отношению к таким объектам, как функции, но в той же мере и к числам. Чему равна информационная емкость числа π ? Бесконечности. Этот объект принципиально недоступен компьютеру. И когда мы говорим, что компьютер совершает операции над числами, то допускаем языковую вольность, из которой вырастает заблуждение. Компьютер работает с объектами совершенно иной природы — с текстами, построеными из букв конечного алфавита (неважно какого, они все эквивалентны), т. е. с информационными моделями объектов (чисел, заводов, функций, живых клеток, экосистем и пр.).

Но эти шероховатости вполне устранимы. Перед нами отличный ознакомительный курс, рассчитанный на удовлетворение нормально-детского любопытства. Причем есть прямой смысл использовать этот курс после небольшой адаптации не в старших, а в средних классах, где он будет полезен во многих

отношениях.

Итак, резюмируем наши позиции. Первый учебник (Каймин В. А. и др.) не пригоден для обучения информатике. Второй (Кушнirenко А. Г. и др.) — это учебник, который

годится для учащихся со сформировавшимся интересом к информатике, причем формы его использования могут быть весьма различными. Третий учебник (Гейн А. Г. и др.) — это хороший ознакомительный курс.

Ленинградский производственный кооператив «ОСТ» ПРЕДЛАГАЕТ

программное обеспечение для ПЭВМ БК-0010(-01), ДВК, УКНЦ, IBM PC, «Синклер Спектрум», комплектов КУВТ-86, КУВТ УКНЦ.

Мы располагаем обширным банком учебных, игровых, системных, прикладных программ.

Расценки значительно ниже государственных.

Каталоги программ (при указании типа ЭВМ) высылаются бесплатно.

115

Пользователям БК НОВИНКИ!

Программная разработка, позволяющая помещать на магнитофонной кассете в три раза больше программ по сравнению с традиционным способом записи и одновременно многократно увеличить надежность считывания; недорогой телефонный модем.

189510, Ленинград, а/я 649, «ОСТ».

Пользователям классов УКНЦ

Молодежный центр «Фактория» предлагает сетевое системное программное обеспечение

новая генерация ОС RT-11, корректно работающая с 8-битными кодами (символы квазиграфики, заливки фона, ESC — последовательности для управления атрибутами дисплея);

программы межмашинной связи для быстрой (10—15 сек) загрузки и обслуживания новой генерации ОС RT-11 на локальных ЭВМ;

новые варианты локального и дискового Бейсика: реализована возможность одновременного вывода символов и графики на дисплей, имеются программы межмашинной связи для поддержки всех файловых операций.

Программное обеспечение одобрено Государственным комитетом СССР по народному образованию и рекомендовано для внедрения.

Более подробную информацию можно получить по адресу:

103482, Москва, К-482, Зеленоград, корп. 06, МЦ «Фактория».

Тел. в Москве: 532-98-78, 534-65-06.

Д. ПЕРЦЕВ

заведующий отделом информатики ВМО «ГТГ»

Банк данных изобразительного искусства в Третьяковской галерее

Тенденции музейной компьютеризации

Большая часть крупных зарубежных музеев компьютеризирована, причем этот процесс продолжается уже около четверти века. Существуют сотни самых разнообразных музейных баз данных (БД), как документальных, так и фактографических*, относящихся к области изобразительного искусства. Десятки из них доступны любому пользователю в Европе, имеющему персональный компьютер и средства для его подключения к телефонной сети.

В развитии музейной компьютеризации за рубежом четко прослеживаются два пути.

Первый из них, типичный для США, Японии, Англии и других стран, характеризуется наличием множества центров (музеев), в каждом из которых развивается своя система. Так называемая музейная компьютерная сеть США (MCN, как утверждает ее создатель Дэвид Вэнс, есть не более чем реликтовое название, сохранившееся от нереализованного проекта сети терминалов с головной машиной IBM 360/20). Конечно, сегодня телекоммуникационные каналы связи используются, но примечательно, что нынешний глава информационного центра, финан-

сируемого из фонда Гетти, М. Эстер жалуется на их ненадежность. Иначе говоря, децентрализация — это не планируемый результат, а скорее следствие, причины которого — отчасти в отсутствии правительственного финансирования, поддерживающего какую-либо конкретную программу.

В странах, реализующих второй путь, — это прежде всего Франция, Канада, Дания, — правительства через министерства культуры финансируют программу музейной компьютеризации. Она характеризуется централизованными тематическими базами данных.

Примером такой БД является «Джоконда», которую составляют данные о французской живописи, скульптуре и графике из всех музеев страны. Поддерживается эта база мощной ЭВМ, связанной более чем с сотней терминалов в музеях и региональных центрах Франции. В течение последних пяти лет наряду с БД описаний создаются на интерактивных видеодисках или цифровых оптических дисках БД цветных изображений, сопряженных с БД описаний. В одной только Франции существует несколько десятков БД типа «Джоконда», как фактографических, так и документальных.

Безусловное преимущество второго пути — возможность посредством однократного обращения к системе получить данные из десятков музеев одновременно, что в условиях децентрализации затруднено, а по некоторым классам запросов просто невозможно. Конечно, это преимущество дается не даром, второй путь требует гораздо больших затрат и сопряжен с решением це-

* Не давая развернутого определения этим двум типам информационных систем, отметим, что отличительной чертой фактографических систем является необходимость предварительного определения круга объектов и их признаков еще до ввода информации в систему. Из этого определения вытекает, что основной задачей при разработке фактографических систем является проблема унификации структуры описания, которой в статье уделяется большое внимание.

лого ряда организационных, технологических и научных проблем. При реализации первого пути эти проблемы откладываются «на потом», когда встанет вопрос интеграции систем. И как это часто бывает, более дешевый путь может обернуться большими потерями впоследствии за счет проведения повторных работ.

Следует упомянуть о третьем компромиссном пути, по которому идут наши итальянские коллеги. Министерство культуры Италии выделяет большие средства на унификацию предмашиной обработки информации. В Риме создан Институт каталога, в котором в настоящее время находится свыше 2 млн. карточек. В них по единой схеме описаны памятники изобразительного и прикладного искусства, архитектуры, этнографии, археологии и т. д. Компьютеризация же осуществляется с использованием самой разнообразной техники и прикладного программного обеспечения. Несмотря на это, унифицированная структура описания сохраняет хорошие перспективы интеграции этих систем.

Работы в области компьютеризации отечественных музеев начались приблизительно четверть века назад, однако из фазы постановки задачи эти работы стали выходить только в последние годы. Началось финансирование крупнейших музеев, в ряде случаев, валютное, что позволило приступить к их укомплектованию современной техникой. Пока в этой области нет ни одного функционирующего банка данных (БНД), и речь может идти только о более или менее конкретных планах. Что касается баз данных, то в ряде музеев они существуют и регулярно пополняются.

Именно сейчас, когда есть возможность выбора, целесообразно оценить опыт тех, кто уже проделал часть предстоящего нам пути, чтобы не повторять ошибок или по крайней мере не совершать их по неведению. Как уже отмечалось, выбор того или иного пути музейной компьютеризации определяется не только желанием, но и возможностями технических и программных средств.

Технические средства и программное обеспечение АИС «ИЗО»

В 1986 г. был заключен контракт между Третьяковской галереей и итальянской фирмой «Оливетти». Фирма взяла на себя обязательства поставить оборудование и разработать прикладное программное обеспечение для централизованного банка данных (БНД) отечественного изобразительного искусства. Это оказалось сложным и трудоемким делом. Поэтому мы опишем не только поставленное и функционирующее оборудование,

но и то, которое войдет в комплекс технических средств при модернизации, проводимой фирмой в рамках контракта. Следовательно, Третьяковская галерея имеет неплохие перспективы для создания и ведения централизованного БНД по коллекциям отечественного искусства из различных музеев СССР (АИС «ИЗО»). Отличительной особенностью автоматизированной информационной системы «ИЗО» является видеоконтур, который позволит иметь БД цветных изображений.

БНД алфавитно-цифровой информации

АИС «ИЗО» функционирует в сетевом режиме, осуществляемом с использованием коаксиального кабеля (в Государственной Третьяковской галерее), позволяющего вызов как изображений, так и телефонных каналов для удаленных терминалов.

Головной машиной является мини-ЭВМ SP-644 (типа PDP-11/44, США):

- оперативная память 4М байта;
- 4 винчестера по 134М байта;
- 3 дисководов для съемных дисков по 256М байт;

- 2 магнитные ленты;
- дополнительное оборудование для локальной сети ETHERNET (контроллер DELUA, трансивер с кабелем);

- дополнительное оборудование для хранения файлов изображений (библиотека оптических дисков прямого доступа на 56 двухгигабайтных оптических дисках).

В качестве терминалов в сети используются персональные компьютеры М-290 «Оливетти» типа РС/АТ:

- процессор 80286, сопроцессор 80287, оперативная память 2М байта;



цветной дисплей с адаптером VGA;
флоппи-диск 3,5" — 1,44М байта;
винчестер — 100М байт;
стриммер — 40М байт;

дополнительное оборудование для визуализации: графический адаптер TARGA-16, цветной монитор;

дополнительное оборудование для локальной сети ETHERNET (плата 3-COM, трансивер с кабелем).

Данное оборудование позволяет ввести в центральную БД около 600 тыс. описаний более чем по 130 реквизитам.**

Чтобы не перегружать головную ЭВМ, пользователь сможет иметь локальную БД на своей ПЭВМ, соответствующую профилю его работы (например, «Дореволюционная графика из фондов Государственной Третьяковской галереи»). Только в том случае, когда ему нужно выйти за пределы этой локальной БД, он обращается к центральной БД. Объем центральной БД составит свыше полумиллиона описаний — это вся живопись, скульптура и часть оригинальной графики отечественного искусства всех музеев нашей страны.

Что касается реквизитов описания, то центральная БД позволит находить любые данные, которые входят в самые подробные каталоги. Отметим также, что с такой степенью подробности описана лишь небольшая часть коллекций, находящихся в фондах музеев. Что касается научного описания, то к нему мы вернемся в конце статьи.

Поскольку авторов существенно меньше, чем произведений, то они выделены в отдельную БД. Структура описания включает 21 реквизит.

Официально система еще не принята в эксплуатацию, однако в процессе отладки созданы локальные базы, включающие несколько тысяч описаний.

БД изображений. Видеокадр АИС «ИЗО»

По ряду соображений как организационного, так и чисто технического характера планируется создание и ведение нескольких БД изображений, отличающихся техническими характеристиками и функциональным назначением.

Цифровая БД, предназначенная для визуализации в АИС «ИЗО»

Ввод изображений в студийных условиях проводится телевизионной камерой, вне студии — «Камкордером» с промежуточной записью изображения на видеомagnитофон

**Реквизиты АИС «ИЗО» распадаются на две основные группы — учетные (автор, название, размер, техника, материал и т. п.) и научные.



в стандарте S-VHS. Ввод осуществляется как с оригиналов, так и со вторичных источников. Графический адаптер TARGA-16 (512×480 пиксел, 32 тыс. цветов). Накопитель на 56 оптических дисках WORM по 1 или 2Г байт типа Juke-box. Время поиска в заполненной БД (свыше 200 тыс. изображений) — от 30 до 40 с, что затрудняет диалоговый режим.

Поэтому по мере накопления БД изображений планируется использование в диалоговом режиме аналогового контура с использованием видеопроигрывателя, формат Laser Disk, вмещающий 100 тыс. изображений. Время визуализации изображения, размещенного на одной стороне диска, — 1—2 с. Устройство надежно и дешево. Недостаток в том, что запись БД на диск может быть проведена только в заводских условиях, т. е. этот контур может использоваться только

для уже готовых БД изображений, включающих 80—100 тыс. записей.

Цифровая БД для использования в экспресс-полиграфии

Ввод осуществляется с издательского слайда посредством сканера слайдов («KODAK» EIKONIX, разрешение 2800 лин/дюйм, 4096 уровней серого). Используется графический адаптер ATVista (1024×1024 пиксел, 16 777 216 цветов). Визуализация на высоко разрешающем дисплее 1280×1024 пиксел. Твердая копия в формате A4 на цветном принтере струйного типа JUKI 8000 (240 лин/дюйм) либо в виде слайда с разрешением 2732 лин/дюйм на слайдпринтере.

Таким образом, видеоконтур АИС «ИЗО» будет служить не только целям визуализации, но вместе с БД описаний при использовании издательской системы Desktop Publisher позволит оперативно готовить оригинал-макеты каталогов выставок, которые в настоящее время почти никогда не выходят в срок.

Помимо локальных и центральной БД, посвященных истории отечественной живописи, скульптуры и графики, планируется разработка и ведение ряда специфических подсистем, некоторые из них уже функционируют в модельном варианте. Это подсистемы для отдела учета, отслеживающие движение экспонатов, выдачу и приемку на постоянное и временное хранение, автоматизирующие составление сопроводительной документации.

Подсистема «Реставратор» будет иметь документальный и фактографический контуры описания, а также видеоконтур. Последний позволит в графическом виде фиксировать заболевания (например, растрескивание красочного слоя) и их границы, что, с одной стороны, предельно упростит ведение описаний сохранности, т. е. документацию, сопровождающую выставки, с другой стороны, на новой основе позволит реставратору вести научную работу, устанавливая корреляции между типом заболевания, с одной стороны, и условиями хранения или авторской техникой, материалом и т. д., с другой.

Заслуживает внимания уже существующая в виде модели подсистема «Дизайнер». Она позволяет получить на мониторе стену любой конфигурации и цвета, привести в соответствующий ей масштаб живопись или графику, которую предполагается экспонировать на этой стене, передвигать ее, пока не будет найдена удовлетворяющая дизайнера экспозиция. Затем изображение стены с найденной экспозицией распечатывается на принтере с указанием реальных координат

левого нижнего угла картины. Распечатка может быть передана непосредственно повесочной бригаде. «Дизайнер» позволит избежать значительных при подготовке любой выставки затрат на эскизы и макеты.

Проблемы и перспективы

Из изложенного выше вытекает, что комплекс технических средств и программное обеспечение АИС «ИЗО» позволяют создать и вести централизованный банк данных отечественного изобразительного искусства, т. е. пойти по пути наших французских коллег, однако условия, перечисленные выше, необходимы, но недостаточны.

Прежде всего должны быть решены организационные проблемы. Это «Положение о банке данных», которое будет регулировать финансовые и правовые нормы взаимоотношения держателя БД и его пользователей, определять условия, на которых в БД должны передаваться локальные БД из других музеев. Должны быть решены вопросы, связанные с технологией и программными средствами, которые позволят осуществить слияние баз, созданных с использованием различных форматов. Последняя задача хотя и не представляет принципиальных трудностей, однако требует внимания. Достаточно сказать, что в настоящий момент в Русском музее созданы и ведутся три локальные БД, каждая из которых имеет свой формат.

Необходима правовая защита информации, вводимой в БД, поскольку некоторая ее часть вводится до публикации. По-видимому, наиболее разумным решением было бы приравнять ввод информации в БД к публикации, для чего необходимо принятие соответствующего закона. Первые шаги в этом направлении предприняты по линии Министерства культуры СССР. Эта проблема важна для разработчиков фактографических систем во всех областях знаний.

Выше мы рассматривали проблемы, связанные с вводом в БД учетных данных. Веками складывавшаяся здесь конвенциональная система описания является всеобщей, и если между музеями и существуют различия учетного описания, то чаще всего они выражаются в степени подробности отдельных элементов. Поэтому перспектива интеграции этих БД выглядит достаточно оптимистично.

Но если мы ограничимся вводом в БД только учетных характеристик, то предельно сузим круг пользователей и превратим информационные системы в электронные картотеки, в то время как, по нашему глубокому

убеждению, автоматизированные системы, включающие в себя научное описание, будут означать новый этап в искусствоведении.

Что же такое научное описание в искусствоведении?

С полной определенностью можно утверждать, что в традиционном искусствоведении научное описание существует, но при этом определяется в достаточно широком диапазоне — от попыток формального анализа, предпринимаемого отдельными авторами, до описания, иногда высокохудожественного, собственной эмоциональной реакции на описываемый объект. Совершенно очевидно, что в качестве научного описания для ввода в информационную систему такой подход не годится.

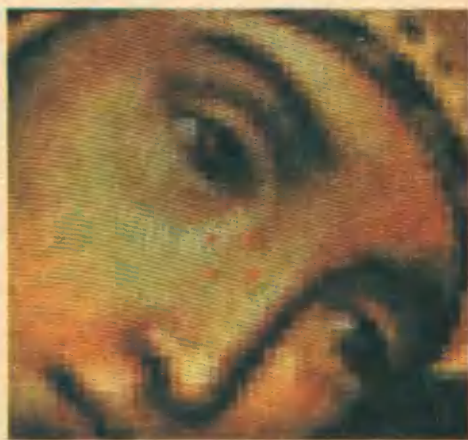
Поэтому, с одной стороны, ясно, что должна быть создана унифицированная система научного описания памятника искусства для ввода его в банк данных. (Конкретные попытки такого рода предпринимаются как у нас в стране, так и за рубежом, в частности АИС «Памятник».) С другой стороны, совершенно непонятно, как решать эту задачу. Ведь научные описания должны готовить специалисты-искусствоведы, следовательно, унифицированное описание должно носить конвенциональный характер, т. е. различные специалисты должны договориться между собой о его структуре и далее строго ее придерживаться.

Но сегодня не видно той платформы, которая послужила бы основанием для такой конвенции. Надежды на то, что информатика может послужить основой такой конвенции, сомнительны. При создании более или менее универсальных систем описания каждый автор или коллектив хочет, чтобы она стала общепринятой, для чего в виде «тройного коня» система снабжается прикладны-

ми программными средствами. Применение декретирования, предписывающего всем использовать какую-либо одну схему научного описания, на наш взгляд, не даст нужного эффекта, ибо будет саботироваться, а кроме того, такой подход, несомненно, обеднит бы искусствоведение — область, где сфера интуитивного очень велика. По крайней мере в археологии, где в течение четверти века предпринимались усилия создать единую систему кодификации, возникла ситуация прямо-таки феодальной раздробленности. Почти каждый автор предлагает свою классификацию исследуемого материала, притом действующую в узких пространственно-временных рамках.

Итак, подведем итоги. Для успешного функционирования информационных систем в художественных музеях необходимо унифицировать систему научного описания памятников. Эту унифицированную систему должны применять все искусствоведы, занимающиеся их описанием, иначе невозможно ввести в систему те миллионы памятников, которые хранятся в музеях. По мнению автора, сегодня нет платформы, которая позволила бы унифицировать описание на конвенциональной основе, а декретирование не дает ощутимых результатов. На первый взгляд кажется, что ситуация совершенно безвыходная, но это не совсем так.

Попробуем взглянуть на описанную проблему под несколько иным углом. В настоящее время в искусствоведческой науке практически нет формализованных методик, которые унифицировали бы процесс исследования; с другой стороны, имеется большое количество экспертов, накопивших громадный эмпирический опыт в области классификации исследуемых объектов. Эксперт, пользуясь интуитивным опытом, не выделяя в явном виде признаков, позволяющих



классифицировать объекты, прекрасно умеет, однако, распознавать предметы искусства определенного периода, региона, в ряде случаев может узнать конкретного мастера и определить период его творчества. Фактически любой искусствовед является экспертом в той или иной области. Не случайно в ряде европейских стран слово «эксперт» является синонимом слова «искусствовед» и, как кажется, употребляется чаще.

Следовательно, стремясь к унифицированной системе описания памятника искусства, необходимо в явном виде выделить и сформулировать характеристические черты, которые позволяют узнавать анонимный памятник, т. е. находить его место в пространственно-временной шкале (в истории искусства), иными словами, проводить атрибуцию.

Конечно, к проблеме атрибуции не сводятся все вопросы искусствознания. Например, определение эстетической ценности художественного памятника не входит в задачу атрибутирования, однако же очевидно, что эта задача приоритетна. Иными словами, сначала нужно формализовать узнавание объекта, а все остальные задачи формализации должны строиться на этом фундаменте.

В общем виде внеличностный критерий отбора признаков научного описания памятника можно сформулировать так: если выделенный признак описания позволяет определить место описываемого памятника в пространственно-временной шкале, т. е. провести его экспертизу или атрибуцию, то он полезен и его целесообразно ввести в систему научного описания. Отметим, что тот же самый принцип был использован Карлом Линнеем для систематики в естествознании.

Остается выбрать удобную методику, которая позволяла бы использовать эту громадную кладовую несистематизированных знаний, носителями которых являются эксперты-искусствоведы; в явном виде сформулировать признаки, которые позволяют им проводить атрибутирование, систематизировать и иерархизировать их.

Такая методика, по нашему глубокому

убеждению, может быть реализована на основе экспертных систем. В нашем случае — это экспертные системы автоматизированной атрибуции. Создаваемые в контакте с экспертом-искусствоведом, специалистом в той области, для которой строится система, они уже в процессе построения и особенно отладки будут играть роль эффективного фильтра для отбора атрибутивно-различительных признаков, образующих систему научного описания.

Изложенные выше соображения реализованы в экспертной системе автоматизированной атрибуции иконографии древнерусских сюжетных икон, которая создана в отделе информатики Государственной Третьяковской галереи. Планируются совместные работы по построению экспертных систем в археологии.

Перспективой развития такой экспертной системы является ее слияние с информационной системой и создание на этой основе базы знаний. Включение видеоконтур и интерактивная реализация на видеодиске позволят стать пользователями таких баз знаний как специалистам, так и людям, интересующимся вопросами искусства, имеющим в своем распоряжении персональный компьютер и видеоплеер.

Мы убеждены, что создание и распространение баз знаний такого рода будут иметь глубокие последствия как для развития соответствующих областей гуманитарных наук, так и для массовой культуры в целом.

Литература

1. V. Mirimanov, D. Pertsev. L'ordinateur — un fichier électronique ou une nouvelle étape dans l'étude de l'art? // AICARC. № 2. 1986; № 1. 1987.

2. Компьютеризированные банки данных музейной и археологической информации: (Материалы межведомственного рабочего совещания. Тбилиси, 1988 г.). / Перцев Д. Г. К вопросу об унификации научного описания памятников культуры.

ЧТО? МОЖЕТ ЭВМ

Чем хорош микро-процессор

Отрицательные эффекты (в частности, снижение надежности и управляемости), сопровождающие усложнение разнообразных систем, пропорциональны квадрату числа элементов системы. Почему «большие» ЭВМ (типа ЕС-1060) обслуживает бригада инженеров-электронщиков, поддерживающих их работоспособность, а персональный компьютер той же вычислительной мощности работает без особого ухода год-два-три, не ломаясь?

Сложная структура микросхемы требует высокой технологии изготовления; готовое же изделие работает как один элемент вычислительной системы. В результате надежность ПЭВМ, содержащей несколько десятков микросхем, примерно равна надежности одной платы «большой» ЭВМ, содержащей несколько десятков транзисторов. Именно поэтому за последние 13 лет крупные компьютеры сдали свои позиции почти целиком: если в 1977 г. в них были сосредоточены практически все вычислительные мощности, то сейчас — менее 1%.

Все ниже, и ниже, и ниже...

Приятно в наше стремительно дорожающее время узнать, что что-то дешевле. Тем более, когда это касается информации.

Да, цены на информацию продолжают снижаться. Точнее, не на информацию, а на ее хранение в компьютере. И теперь еще недавно постоянно требующий ответа вопрос «что выбросить?» возникает все реже. На магнитном носителе для хранения оставляется практически все. Ведь при покупке внешних запоминающих устройств употребление слова «килобайт» считается неприличным, «мегабайт» терпим с множителем не менее 100, но по-настоящему деловым человеком

вас будут считать только в случае, если вы собираетесь приобрести что-либо емкостью в несколько гигабайт.

Немалую роль в этой области сыграла американская корпорация Maxell, представившая недавно два новых картриджа для хранения больших объемов цифровых данных. Один из них рассчитан на использование магнитной ленты шириной 4 мм и позволяет хранить до 1,3 Гбайт, на второй, восьмимиллиметровый, можно записать 2,75 Гбайт.

Новая технология производства магнитной ленты, включающая проверку качества нанесения рабочего слоя по всей поверхности, гарантирует безошибочную запись, хранение и считывание данных любой длины.

В конструкции новых кассет предусмотрено даже то, что с увеличением объема хранимых данных возрастет и число карандашных записей на внешней стороне кассеты. Для этого увеличена площадь бумажной наклейки.

Большой популярностью пользуется 8 мм формат, а 4 мм картриджи используются в основном как промежуточные между уже существующими 320 мегабайтными накопителями на 1/4-дюймовой ленте и 8 мм картриджами емкостью 2,6 Гбайт. Хотя малые размеры, большая емкость и низкая цена могут вскоре сделать их самым популярным хранилищем драгоценной, но все более дешевой информации.

Своя память не тянет

Хорошо организованная и тренированная память — гордость культурного человека. Она раскрепощает творческие возможности, освобождает время и облегчает жизнь.

Видимо, помня об этом, конструкторы японской фирмы Canon решили оборудовать персональный факс Canofax 70D встроенной цифровой памятью емкостью 1934 байта, в которой может храниться телефонный справочник, график работы и даже небольшой музыкальный фрагмент.

Факс может автоматически набирать телефонный номер, пока абонент не откликнется на вызов, а затем принять или передать текстовое или графическое послание форматом с машинописную страницу всего за 30 секунд. Несмотря на разнообразие выполняемых функций, факс сравнительно недорог — 968 долларов. Новый факс очень компактен и занимает на столе пространство не более $306 \times 250 \times 110$ мм. Но если он начнет мешать, его совсем нетрудно переставить, ведь вес 4,1 кг может поднять даже ребенок.

Не выходя из дома

Мало кто может позволить себе иметь собственный концертный зал или стадион, чтобы слушать громкую музыку в ее естественном концертном звучании. Но даже если у вас маленькая комнатка, втиснуть в нее целое футбольное поле позволит цифровой процессор звука EQS-1000 японской фирмы Fujitsu, который может модифицировать оцифрованный звуковой сигнал таким образом, что у слушателя создается ощущение большого открытого пространства.

Процессор может работать в нескольких режимах, устанавливая которые слушатель переносится из небольшого джаз-клуба в концертный зал, из кафедрального собора на стадион. А если вам надоело необъятные просторы, можно нажатием кнопки вернуться в родное жилище. Тем более, что процессор умеет это делать сам, реагируя на переход музыкальной программы в дикторский текст.

Мы за «Агат»

Выбор школьных компьютеров невелик: БК-0010, БК-0011, «Корвет», УКНЦ и «Агат». Первые четыре — это КУВТы, управляемые с головной учительской машины. «Агат» же — настоящая персональная ЭВМ: каждая машина имеет свой дисковод, что позволяет ученику самостоятельно, независимо ни от кого работать, надеясь только на себя, на свои знания и на своего помощника — ПЭВМ «Агат». А учителю, знающему способности своих учеников, их уровень подготовки, — давать персональные задания, обеспечивать индивидуальный подход к каждому ученику.

Известно, что прототип «Агата» — американская ЭВМ «Apple II». От нее пришли открытая архитектура и на редкость доброжелательный к пользователю интерфейс, который обеспечивает простоту общения и быстрое обучение работе с ПЭВМ.

Основой «Агата» являются распространенные в советской электронике недефицитные серии микросхем, что позволяет осуществлять быстрый ремонт в условиях эксплуатации.

Управление народного образования Карагандинской области в 1989 г. заключило прямой договор на поставку ПЭВМ «Агат» с Волжским заводом электронно-вычислительной техники. За полтора года было приобретено компьютеров на сумму свыше 9 млн. рублей — это 180 классов. Деньги МНО Казахской ССР отказалось выделить, но они нашлись у города, предприятий, заводов, колхозов и совхозов области. Хотя «Агаты» ругали на всех уровнях, мы сделали ставку на них — и не ошиблись. Результаты сказались незамедлительно. Через год наши дети заняли два первых и два

вторых места на республиканской олимпиаде по информатике и поехали на всесоюзную олимпиаду в Харьков.

Нам нравится внимательное отношение Волжского завода ЭВТ к пользователям, и это не только наше мнение. Завод имеет очень тесные контакты с ВЦ Томского политехнического института, Киевским областным ИУУ, Волгоградским ИУУ и др.; обеспечивает доставку компьютеров к потребителю, их бесплатную установку и бесплатный ремонт при вводе в эксплуатацию. Предоставляет программные средства, ЗИП для ремонта, дополнительные дискеты. «Агаты» Волжского завода укомплектованы мониторами МС 6105.01, использующимися в профессиональных ПЭВМ типа ЕС 1840 и имеющими наилучшие показатели по всем параметрам. По данным медицинских исследований, использовать их в текстовом и динамическом режимах можно более длительное время, чем другие отечественные мониторы.

Сейчас Волжский завод ЭВТ совместно с Карагандинским облУНО, Карагандинским НПЦ «Ритм», Томским политехническим институтом создает пакеты учебных программ по всем предметам с X по XI класс, позволяющие информатизировать учебный процесс. Эти программы будут прилагаться к «Агатам» без увеличения их стоимости.

На базе Волжского завода ЭВТ планируется создание Всесоюзной ассоциации пользователей ПЭВМ «Агат». Ее задачи — разработка пакетов обучающих программ, создание учебника и методик для учителей и учеников, создание банка идей, программного обеспечения, обучение пользователей, распространение программного обеспечения

и методических разработок через филиалы ассоциации, изучение, обобщение и перевод на русский язык вновь появляющихся ПС, а также связь с Международным компьютерным клубом и с фирмой «Apple».

По нашему мнению, «Агат-9» незаменим при углубленном обучении информатике в УПК, в спецклассах. В Карагандинской области с этого года в качестве эксперимента в некоторых школах вводится курс информатики с V класса. И мы убедились, что для изучения информатики в младших

классах средней школы, детских садах оптимальный вариант — «Агат-7», для старших спецклассов — «Агат-9».

А главный недостаток «Агата» — высокая стоимость. Госкомцен СССР должен обратить очень серьезное внимание на данный вопрос и снизить их цену. Тогда эти ПЭВМ станут доступными каждому учебному заведению.

Л. КОЙСИНА, методист ИВТ
Карагандинского ОИУУ

Против помех на УКНЦ

В «ИНФО» № 5 за 1989 г. опубликовано письмо Г. Беркова «Печальное приобретение», в котором отмечены плохая работа локальной сети и другие недостатки КУВТ УКНЦ. Действительно, некоторые мониторы дают помеху. Проведенные нами осциллографические измерения показали, что импульсная помеха проникает даже в сеть питания 42 В и в конечном счете нарушает работу локальной сети.

Нам удалось устранить сбой очень простым способом. В старой аппаратуре связи мы нашли кольца из альсифера такого диаметра,

что в них проходит вилка сетевого шнура монитора, намотали этим шнуром на кольце 3—4 витка и снова включили вилку в розетку сети питания. Получился простейший сетевой фильтр. А чтобы уменьшить амплитуду помехи с антенного гнезда монитора, нужно контакт, к которому припаян экран коаксиального кабеля, соединить с корпусом монитора в непосредственной близости от гнезда.

А. КАРАБАЕВ

«Правец» на уроках математики

В этом году в кабинете математики (СШ № 10 г. Янгиабада) установили КУВТ «Правец 8С». Программ для работы на уроках математики не было ни одной. Пришлось начинать с нуля. Постепенно была создана и отработана на уроках система программ для занятий математикой в V классе практически по всем темам. Она дает неплохие результаты при обучении детей с замедленной реакцией, которые из фронтальной работы, как правило, выпадают. Среди составленных программ есть тестовые, определяющие уровень обученности класса в целом,

обучающие с предложением теории и полного решения задачи в случае неправильного ответа, тренажерные для отработки определенных навыков. Для индивидуализации обучения предусмотрены более сложные задачи, требующие комплекса знаний, и более простые, ориентированные на обязательные результаты обучения. Может быть, мои программы кому-либо будут нужны? А мне взамен кто-либо предложит аналогичную работу по другим классам? Буду очень рада.

Е. Н. СИДОРОВА.
702520, Ташкентская обл., Янгиабад, квартал 14, д. 18, кв. 38.

Хочу реализовать свои идеи по работе со школьниками разного возраста. Ищу работу заведующего компьютерным классом в школе, клубе или другом учреждении любого города. Необходимо жилье. Обращаться по адресу: 152140, г. Переславль-Залесский, ул. Свободы, д. 45, комната 18. Зайдельману Якову Наумовичу.

Ищем единомышленников

Программа моделирования кубика Рубика занимает почти все ППЗУ МК-52. Она состоит из четырех частей, каждая из которых заканчивается адресом следующей части в ППЗУ. Максимальная продолжительность работы одной части — 4 мин. Исходные данные заносим в регистры 1—9 из ППЗУ. Программа легко управляема: используются только клавиша С/П, две клавиши обращения к ППЗУ. Поворот горизонтальной и профильной граней задается тремя цифрами, например 123, 963 и т. п. Ряд цифр на клавиатуре соответствует грани. Порядок следования цифр определяет направление поворота. Поворот фронтальной грани

задается одной из цифр: 1, 2, 3 — в зависимости от слоя. Если цифра с минусом, то поворот против часовой стрелки. Программа иллюстрирует работу кубика. Есть у нас и другие интересные программы.

Наша школа приобрела БК-0010-01, и мы с учащимися попытаемся увидеть на экране кубик и его преобразования. А пока осваиваем программы для БК из вашего журнала. Будем рады, если через журнал найдем единомышленников.

Адрес: 412300 Саратовская область, р/п Романовка, ул. Береговая, 14а, кв. 15, *Перову В. А.*

125



Миф-91

Состоит из обучающих, демонстрационных и контролирующих программ по курсу физики и компьютерного учебника математики для средних общеобразовательных и физико-математических школ.

Цена комплекта: в заказе до 10 комплектов — 1000 руб.;
от 10 до 100 — 750 руб.;
свыше 100 — 500 руб.

Игрушка

В состав пакета входят игры: «Conan», «Sammy», «Knight», «Lode Runner». Поставляется с комплектом джойстиков.

Цена: в комплекте с 12 джойстиками — 2500 руб.;
с 6 джойстиками — 1400 руб.;
с 1 джойстиком — 500 руб.

Квант

Профессиональный инструментальный графический редактор, предназначенный для интерактивной подготовки и коррекции графических изображений.

Цена комплекта: в заказе до 10 комплектов — 500 руб.;
от 10 до 100 — 400 руб.;
свыше 100 — 350 руб.

Школа-семинар в Ленинграде

126

В сентябре 1990 г. Ленинградской Ассоциацией преподавателей информатики* совместно с областным и городским институтами усовершенствования учителей и ЛГПИ им. А. И. Герцена была проведена школа-семинар «Концептуальные основы, содержание и методика преподавания информатики в средней школе» по проблемам преподавания информатики в средних учебных заведениях. В программе обсуждение концепции информатизации среднего образования в целом и курса информатики в частности; узловые вопросы содержания и структуры этого курса и методики его преподавания.

Цикл лекций, прочитанных А. А. Матюшкиным-Герке, позволил слушателям с новых, порой неожиданных точек зрения рассмотреть проблемы становления курса, внедрения компьютеров в учебный процесс в целом. На практических занятиях можно было познакомиться с рядом очень интересных педагогических находок и учебных программно-информационных средств. Из числа последних упомянем экранную модель ЭВМ, учебный кросс-транслятор, программы решения прикладных задач, автоматизированную систему контроля и самоконтроля.

Участники школы-семинара познакомились с работой своих коллег, постановкой учебного процесса в педагогическом институте и в институтах усовершенствования учителей.

Слушатели увезли многочисленные методические материалы, которыми их снабдили организаторы семинара.

Резолюция

1. Мероприятия подобного рода в настоящее время исключительно актуальны, ибо дают возможность учителям и методистам пополнить свой багаж научных и практических знаний, обменяться опытом работы, познакомиться с используемыми их коллегами программными и методическими средствами. Необходимо всячески поддерживать проведение таких школ, семинаров, творческих встреч, сделать их более частыми и массовыми, заблаговременно планируя для этого выделение необходимых средств и временное замещение командированных туда участников на их основной работе.

2. В дальнейшем необходимо изыскать возможности для организации достаточно продолжительной учебы учителей информатики с полным или частичным отрывом их от работы (разумеется, с сохранением средней заработной платы). Следует иметь в виду, что среди учителей информатики по сравнению с учителями других предметов очень мало лиц, получивших одновременно как педагогическое образование, так и профессиональную подготовку в области информатики.

3. Необходимо прекратить беспорядочное снабжение школ, УПК и межшкольных центров разнотипной техникой, так как это крайне затрудняет и техническое ее обслуживание, и создание программного обеспечения, и обмен этими программными средствами. Нужна единая техническая политика, ориентированная на доступные массовой школе типы вычислительной техники. В то же время для опережающей подготовки учительских кадров и некоторых видов

*Ленинградская Ассоциация преподавателей информатики была учреждена в декабре 1989 г.

внеклассной работы следует использовать современные профессиональные ЭВМ.

4. Планы разработки педагогических, программно-информационных средств следует создавать при самом активном участии педагогической общественности. Существующий сейчас порядок приводит к бессмысленной трате времени и средств на создание никому не нужных программных продуктов. В то же время теряется масса ценных разработок, создаваемых в учительской среде, хорошо

себя зарекомендовавших и нуждающихся лишь в определенной доработке и сопровождении.

5. Начало изучения информатики следует перенести в средние классы. Надо провести открытый и гласный конкурс программ этого курса, отобрать (с должным учетом мнения учителей) лучшие из них и затем на их основе разработать и учебники, и программно-информационные средства, и методические материалы для учителей.

КНИГИ

Играйте с компьютером!

«Играйте с компьютером, это захватывает...», «Нет ничего увлекательнее создания собственных компьютерных игр...» — этими лозунгами, взятыми из рекламы микрокомпьютеров, начинается новая книга* Жака Арсака, известного французского педагога и популяризатора, автора бестселлера «Основы программирования», выпущенного еще в начале 80-х гг.

Ж. Арсак пробует себя в новом амплуа: он обходит парижские магазины и везде представляется как отец, который хочет подарить своему сыну микрокомпьютер. Ему объясняют, что именно нужно купить и во сколько это обойдется. «Затем я, — вспоминает Арсак, — спрашивал об играх. И вот что приблизительно там происходило.

... — Кассеты с играми? А сколько стоит кассета?

Продавец называет мне цену, которую вы, должно быть, знаете.

— И сколько игр на кассете?

— Одна, месье.

— Но тогда двадцать игр стоят дороже самой машины! А я слышал от сведущих людей, что на компьютере интереснее всего создавать игры самому.

— Для этого, месье, нужно уметь программировать.

— Но по телевизору твердят, что это очень просто. Мой сын, должно быть, может научиться?

— Да, месье. У нас есть кассета с Бейсиком.

— Кассета с Бейсиком! Бейсик — это и есть программирование?

— Нет, месье. Чтобы уметь программировать, нужна еще вот эта кассета с языком ассемблера...

Я удалялся, исполненный отвращения, жалкий истец отцов семейства.. Реклама трубит: «Создавать — это увлекательно!» Но на практике все делается так, как будто подростки не способны что-либо создавать. И им не помогают начать делать что-нибудь творческое. Чтобы восполнить этот пробел, я и написал эту книгу...»

Читатель познакомится с описанием увлекательных игр («Плата за страх», «Игра роботов», «Твоя песенка слета», «Гениальный отгадчик», «Погоня за сокровищами», «Девушки из пансиона Киркмана», «Анаграмма» — зарубежный вариант «Балды»), шахматных и математических головоломок. К слову сказать, задача, которую нужно решить на компьютере, — это настоящая, дразнящая воображение головоломка.

Книга заинтересует не только юных «компьютерных фанатов». Она будет полезна преподавателям школ и лицеев, руководителям летних семинаров и клубов «Юный программист». Итак, в добрый путь, наш юный читатель! Ведь автор утверждает, что он «написал эту книгу для всех молодых людей от 14 до 77 лет, которые любят играть с компьютером или которые любили бы это делать, если бы средство для этого им было предоставлено». Переиначивая крестьянскую поговорку («Кто сам пилит дрова, согревается дважды»), Жак Арсак, надо думать, не без основания провозглашает: «Кто сам программирует свои компьютерные игры, наслаждается дважды».

*Арсак Ж. Программирование игр и головоломок: Пер. с фр. / Под ред. А. Г. Кушниренко. М.: Наука, 1990. 224 с.

ЧТО МОЖЕТ ЭВМ

На радость Дюймовочке

Звуковые письма — изобретение не новое. С ними были знакомы еще наши бабушки. На смену гибкой почтовой грампластинке пришла более долговечная, но пока еще и более дорогая «говорящая открытка» на интегральной микросхеме. И вот вновь сенсация! Японская фирма Sony приступила к выпуску микрокассет нового стандарта размером с ... почтовую марку (30×21,5×5 мм).

Новая микрокассета занимает в несколько раз меньший объем, чем миникассета, и в 25 раз меньше стандартной магнитофонной кассеты С-120, однако это не мешает ей вмещать два часа высококачественного звука.

Одновременно с кассетой фирма разработала и двухканальный 12-битный диктофон, умещающийся на ладони, который обеспечивает идеальную по качеству цифровую запись и воспроизведение 80 dB звука в диапазоне 10 Hz — 15 kHz. Это делает его серьезным конкурентом значительно более громоздких дисковых цифровых плееров и позволяет его владельцу сделать на профессиональном уровне произвольное количество абсолютно одинаковых копий, не уступающих по качеству оригиналу.

А система двойного сканирования (во время чтения головка вращается в два раза быстрее, чем во время записи) и встроенная цифровая память емкостью 1М бит, позволяющая откорректировать цифровой сигнал перед записью его на ленту, делает диктофон достойной оправой для уникальной по характеристикам кассеты. Имея такую высокую функциональную сложность и отличные характеристики, диктофон настолько мал, что пользоваться им смогла бы, наверное, даже Дюймовочка.

Использовать новую сверхминиатюрную кассету можно не только в диктофоне, но и в автоответчиках, радиотелефонах и автомагнитолах, выпускать которые планирует фирма. Тем более, что механизм загрузки и лентопро- тязки встроены в миниатюрную

кассету, а это упрощает использующие такую кассету устройства и повышает точность механического тракта.

Рассматривается применение микрокассеты и для использования в качестве компактного накопителя информации в персональных компьютерах.

Алло! Вы оцифрованы

Радиотелефоны — не новая продукция фирмы Sanyo Electric. Так же как и автоответчики. Но последние до недавнего времени несмотря на свое японское происхождение вызывали, как говорится, нарекания. А виной всему основной узел автоответчика — портативное лентопротяжное устройство. Этот механический узел требует нежного обращения и, несмотря на технологические успехи, сравнительно громоздок, что не позволяет встраивать его в компактные трубки радиотелефонов.

Выйти из положения позволила другая разработка фирмы: говорящие поздравительные открытки на интегральной микросхеме. Сделанный на основе ИС автоответчик занимает значительно меньший объем, не подвержен износу и потребляет меньше энергии. Вместе с радиотелефоном цифровой автоответчик может работать от встроенной никель-кадмиевой батареи четыре часа подряд. За это время он может принять и запомнить сообщения от 12 абонентов, выдав предварительно из своей памяти 15-секундный текст от имени владельца. Каждое приходящее сообщение должно быть не длиннее минуты, а общее время всех сообщений не должно превышать трех минут. Это немного, однако сообщения могут быть воспроизведены, а память очищена не только по возвращении хозяина, но и по команде с другого телефона.

Так что можно иногда позвонивать к себе домой и справляться о своих делах.

Компьютер учится быть... компьютером

Нейронные компьютеры все больше становятся похожими на человеческий мозг. И хотя они лишь условно моделируют его работу, многие термины, которые раньше можно было употреблять лишь по отношению к человеку, все чаще используются для оценки работы нейронных процессоров. Нейронный младенец, только что вышедший из-под «паяльника», ничего не умеет, но учится жизни завидными темпами.

Японская фирма Fujitsu объявила результаты испытаний нового нейронного суперкомпьютера с параллельной архитектурой, который обучается в 400 раз быстрее своего предшественника, осуществляя 500 миллионов связей между ячейками в секунду. Новая плата размером не более 30 сантиметров может научиться быть персональным компьютером всего за сутки, а освоить диалект сети NETtalk всего за 30 секунд.

Простенько, но со вкусом

Техническому прогрессу во все времена сопутствовало появление новой ортехники. Скоросцива-тели, шариковые ручки, электронные пишущие машинки, наконец. А теперь широкое распространение цифровых видео-, аудио- и компьютерных систем привело к появлению устройств для чистки цифровых компакт-дисков.

Внешне это устройство напоминает изящную раковину гигантского моллюска, под верхней створкой которого расположено ворсистое вращающееся чистящее кольцо, на которое наносится несколько капель специальной жидкости.

Компактный диск укладывается на фрикционную подушку рабочей стороной вверх и накрывается крышкой с чистящим кольцом. Вам остается вставить палец в отверстие диска на крышке устройства и несколько раз проверить его.



Всесоюзный отраслевой фонд алгоритмов и программ общего среднего образования

В соответствии с целевой комплексной программой информатизации Госкомитета СССР по народному образованию по согласованию с Министерством образования РСФСР и Московским городским комитетом по народному образованию 10 сентября 1990 г. образован Отраслевой фонд алгоритмов и программ общего среднего образования (ОФАП ОСО).

ОФАП ОСО образован с целью:

- максимального удовлетворения потребностей учреждений общего среднего образования СССР, России и г. Москвы в программных средствах (ПС);
- сбора, систематизации и апробации ПС для общего среднего образования, подготовки их к распространению и тиражированию;
- проведения оценки ПС посредством рабочих экспертных групп для дальнейшего их представления на утверждение всесоюзной экспертной комиссии;
- организации научно-информационных услуг.

В целях реализации программы информатизации общего среднего образования СССР, России и г. Москвы **ОФАП ОСО выполняет следующие функции:**

- фондирование ПС согласно Приложению I к Положению о ГосФАП ГКВТИ СССР от 07.07.89, № 29;
- формирование и ведение единого информационного фонда ПС для учреждений общего среднего образования СССР, России и г. Москвы;
- организацию информационных, посреднических и научно-технических услуг для учреждений общего среднего образования СССР, России и г. Москвы;
- учет и анализ потребностей в ПС учреждений общего среднего образования СССР, России и г. Москвы;
- ежегодное представление проекта приоритетных направлений разработки и тиражирования ПС для учреждений общего среднего образования СССР, России и г. Москвы;
- подготовку каталогов и информационных бюллетеней;
- организацию ежегодных выставок и на их базе общесоюзных, общероссийских и московских семинаров с целью демонстрации новых ПС;
- методическую и организационную помощь в создании и функционировании региональных центров ОФАП ОСО;
- организацию межфондового обмена на договорной основе.

**Государственный комитет по народному образованию СССР
Министерство образования РСФСР**

Московский комитет по народному образованию

Телефоны ОФАП ОСО: 290-10-49, 213-89-86, 272-11-25.

ул. Билибин, 10-11

Цена 1 р. 20 к.

Индекс 70423

OldPC.ru
7004
музей компьютеров



**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

